Méthodes d'autoréparation proactives pour les réseaux d'opérateurs

Bruno Vidalenc

THESE DE DOCTORAT CONJOINT TELECOM SUDPARIS et L'UNIVERSITE PIERRE ET MARIE CURIE

Spécialité :
Informatique et Réseaux

Ecole doctorale : Informatique, Télécommunications et Electronique de Paris

Présentée par

Bruno Vidalenc

Pour obtenir le grade de
DOCTEUR DE TELECOM SUDPARIS

Méthodes d'autoréparation proactives pour les réseaux d'opérateurs

Soutenue le 28 juin 2012

devant le jury composé de :

Djamal Zeghlache, Professeur à Télécom SudParis (Directeur de thèse)
Francine Krief, Professeur à l'ENSEIRB (Rapporteur)
Michelle Sibilla, Professeur à l'Université Toulouse III (Rapporteur)
Martin Vigoureux, Directeur de recherche à Alcatel-Lucent Bell Labs (Co-encadrant)
Eric Renault Maître de Conférences (HDR) Télécom SudParis(Co-encadrant)
Prosper Chemouil, Directeur de recherche à Orange Labs (Examinateur)
Guy Pujolle, Professeur à l'Université Paris VI (Examinateur)
Selma Boumerdassi, Maître de Conférences (HDR) au CNAM (Examinateur)

Thèse n° 2012TELE0025

Télécom SudParis
SAMOVAR UMR 5157
9, rue Charles Fourier
91 011 Evry

Alcatel-Lucent Bell Labs
Network and Networking
Route de Villejust
91 620 Nozay

Université Paris VI
Ecole doctorale EDITE
4, place Jussieu
75 252 Paris Cedex 05

Cette thèse est dédiée à ma maman.

Just Do It .

Remerciements

Je tiens en premier lieu à remercier mon directeur de thèse, M. Djamal Zeghlache, ainsi que mon encadrant M. Eric Renault pour m'avoir donné l'opportunité de réaliser cette thèse et surtout pour la patience dont ils ont su faire preuve jusqu'au terme de cette thèse.

Je remercie vivement Mme Francine Krief, Professeur à l'ENSEIRB et Mme Michelle Sibilla Professeur à l'université de Toulouse III d'avoir accepté d'être rapporteurs de ma thèse.

Je remercie également Mme Selma Boumerdassi Maître de conférences au CNAM de Paris, M. Guy Pujolle Professeur à l'université de Paris VI et M. Prosper Chemouil directeur de recherche à Orange Labs pour leur participation au jury de cette thèse.

Je remercie tous les membres de l'équipe *Network and Networking* chez Bell Labs France pour m'avoir accueilli à bras ouverts et de m'avoir offert un environnement de travail des plus stimulants.

Je remercie tout particulièrement M. Martin Vigoureux, directeur de recherche chez Bell Labs, pour son encadrement, ses questions et ses précieux conseils qui ont contribué à orienter mes travaux.

Je voudrais dire un grand merci à M. Ludovic Noirie pour son aide précieuse et sa rigueur technique qui ont été déterminants dans la progression de mes travaux, ainsi qu'à M. Laurent Ciavaglia pour les nombreuses discussions enrichissantes et pour toute l'aide qu'il m'a fournie durant cette thèse. Je remercie aussi vivement M. Samir Ghamri-Doudane, Mlle Leila Bennacer, M. Nabil Djarallah, M. Benaissa El Kattar, M. Richard Douville et Mlle Hélia Pouyllau pour leur soutien moral et technique qui m'a permis de mener cette expérience jusqu'à son terme.

Enfin je remercie toutes les autres personnes avec qui j'ai eu l'occasion de travailler aux Bell Labs pour leur aide, la qualité de leurs conseils, ou simplement les discutions enrichissantes que j'ai pu avoir avec elles.

Mes remerciements vont également aux partenaires du projet FP7 EFIPSANS et tout particulièrement à M. Arun Prakash, M. Nikolay Tcholtchev, M. Gabor Retvari, M. Michal Wodczak et Mme Monika Grajzer avec qui j'ai eu la chance de travailler, ainsi qu'à M. Ranganai Chaparadza, chef du projet EFIPSANS pour l'énergie sans limite qu'il a investie pour mener à bien ce projet.

Et merci à ma mère, Mme Colette Vidalenc à qui je dédie cette thèse et qui n'a jamais lésiné sur son investissement dans l'éducation de ses enfants.

Résumé

Les opérateurs de réseaux de télécommunications accordent une importance toute particulière à la gestion des pannes. En effet, la disponibilité et la qualité de service sont des paramètres prépondérants dans la compétition que se livrent les opérateurs. Malgré une meilleure fiabilité des équipements, les pannes et autres incidents n'ont pas disparu pour autant, obligeant les opérateurs à investir beaucoup d'efforts et d'argent dans la gestion des pannes. Cependant, l'implication de l'humain dans la prise de décision et l'analyse d'une quantité énorme d'alarmes et d'informations, ainsi que le caractère réactif des mécanismes de gestion des pannes, ne permettent pas la réactivité nécessaire à une gestion optimale des incidents.

Pour pallier ce problème, cette thèse s'intéresse à des mécanismes proactifs, qui, grâce aux fonctions de gestion autonome réalisées par les équipements, anticipent les pannes afin d'améliorer l'efficacité de leur gestion. En effet, les pannes sont bien souvent précédées d'alarmes ou de comportements symptomatiques laissant présager de leur apparition imminente. Malheureusement, la gestion actuelle ne permet pas d'exploiter ces informations. La mise en œuvre, dans les équipements, de composants autonomes capables d'analyser en permanence l'état de santé du réseau permettrait de fournir une information en temps réel sur le risque de panne, nécessaire au déploiement de nouveaux mécanismes d'autoréparation proactifs.

La première partie de cette thèse est donc consacrée à la définition des composants architecturaux indispensables à l'introduction de fonctions d'autoréparation proactives. Dans un deuxième temps, nous étudions et analysons en détail trois mécanismes d'autoréparation proactifs exploitant une information de risque de panne.

Le premier mécanisme a pour objectif d'accélérer la convergence des protocoles de routage à état de lien en adaptant la fréquence d'envoi des messages de détection de pannes en fonction du risque de panne.

Le deuxième mécanisme modifie dynamiquement les métriques de routage afin de détourner le trafic des équipements risqués et de minimiser l'impact d'une panne sur le trafic.

Enfin, le dernier mécanisme s'attache aux dispositifs de protection et de restauration du protocole GMPLS afin d'adapter dynamiquement la consommation des ressources, aux risques encourus.

Mots-clefs

Réseaux, réseaux autonomes, gestion des pannes, autoréparation, IP, (G)MPLS, OSPF.

Proactive Self-Healing Methods for Carrier Networks

Abstract

Network providers attach a significant focus to fault-management. Indeed, availability and quality of service are highly important parameters in the competition between networks operators. Despite still improved equipment reliability, breakdowns and other incidents have not gone so far, forcing operators to invest much effort and money in fault-management. However, the involvement of human in the decision making process and the analyzing a huge amount of alarms and information, as well as the reactive nature of fault management mechanisms, do not allow the required reactivity for optimal management of incidents.

To overcome this problem, this thesis focuses on proactive mechanisms, which, through self-management functions performed by network equipments, anticipate failures to improve the effectiveness of their management. Indeed, the failures are often preceded by alarms or symptomatic behaviors suggesting their imminent appearance. Unfortunately, the current management does not exploit this information. Implementation, in equipment, of autonomous components capable of continuously analyzing the network health would enable to provide a real-time risk of failure information, required to deploy new proactive self-healing mechanisms.

The first part of this thesis is devoted to the definition of architectural components necessary for the introduction of proactive self-healing functions. Then, in a second step, we study and analyze in detail three self-healing mechanisms exploiting a proactive risk-level of failure information.

The first mechanism is designed to accelerate the convergence of link-state routing protocols by adjusting the frequency of sending failure detection messages function of the risk-level.

The second mechanism dynamically tunes routing metrics in order to divert traffic flows from risky equipment and to minimize the failure incidence on traffic.

Finally, the last proposition is dedicated to the recovery mechanisms of GMPLS protocol by dynamically adapting the resources consumption of recovery to the involved risks.

Keywords

Networking, autonomic networks, fault-management, self-healing, IP, (G)MPLS, OSPF.

Table des matières

Introduction

Contexte

Les réseaux IP [1] constituent aujourd'hui une partie importante de l'infrastructure de l'Internet. Composé de l'interconnexion des réseaux de chaque opérateur, l'Internet est responsable de l'acheminement de nos données à travers le globe. Avec la démocratisation des télécommunications, l'infrastructure réseau n'as cessé de se développer pour former un système très complexe. Alors que la notion de *best effort* était à l'origine de l'Internet, les besoins des utilisateurs ont nécessité la mise en place de critères de fiabilité et de qualité de service (QoS [2]). Un SLA [3] définit de manière formelle l'accord entre un opérateur et son client sur les paramètres qualitatifs du service assuré. L'utilisation toujours plus importante des technologies de l'Internet tel que le multimédia, le télétravail ou l'informatique en nuage a aujourd'hui renforcé le niveau d'exigence en terme de QoS. Pour respecter les besoins en qualité de service et en disponibilité, les opérateurs ont dû mettre en place des mécanismes de gestion de la QoS et de gestion des pannes dans leurs systèmes de gestion de réseau. Ces besoins en outils de gestion de trafic et de réservation de ressources ont conduit les opérateurs à adopter le couple IP/MPLS [4] et plus généralement le protocole GMPLS [5] comme technologie pour leurs réseaux de cœurs.

Gestion des pannes

Les réseaux sont quotidiennement affectés par des incidents. Ces incidents sont des pannes matérielles, des *bugs* logiciels, des activités de maintenance, des erreurs humaines ou encore des catastrophes naturelles mais quelle que soit leur origine, il est essentiel pour l'opérateur de minimiser leur conséquence sur les communications. Malheureusement, malgré l'amélioration de la fiabilité des équipements, bon nombre d'incidents restent inévitables. Les opérateurs utilisent donc des systèmes de gestion des pannes intégrés au protocole de routage. Le routage IP utilise le dispositif de restauration IP intégré aux protocoles IGP [6] tels que OSPF [7] [Moy98] et IS-IS [8] [Ora90] qui impliquent un processus de reconvergence suite à la détection d'une panne. Le trafic est interrompu pendant l'exécution du processus de convergence, il est ensuite rerouté en utilisant les nouveaux chemins. Le protocole GMPLS fournit un ensemble plus complet de mécanismes de gestion des pannes [MP06] avec plusieurs configurations de protection et de restauration.

Les mécanismes de résilience interviennent en réaction à une panne lorsque le temps est compté, afin de rétablir les flux de trafic impactés par la panne. La performance du mécanisme dépend du délai pour rétablir le trafic. La stratégie la plus performante est d'effectuer le maximum d'étapes lors de la configuration du service afin de réduire le délai de rétablissement du service lorsqu'une panne apparaît. Malheureusement, cela implique la réservation de certaines ressources impliquant un coût d'autant plus important.

Le choix de l'utilisation de telle ou telle configuration de gestion des pannes nécessite une étape d'analyse des risques de panne de la part de l'opérateur qui, en fonction du niveau critique

1. *Internet Protocol*
2. Qualité de service (*Quality of Service*)
3. Contrat de niveau de service (*Service Level Agreement*)
4. *MultiProtocol Label Switching*
5. *Generalized MultiProtocol Label Switching*
6. *Interior Gateway Protocol*
7. *Open Shortest Path First*
8. *Intermediate system to intermediate system*

du trafic, choisit la mise en place du mécanisme de gestion des pannes le plus pertinent.

La gestion des pannes n'est pas sans conséquence pour l'opérateur car elle est très coûteuse aussi bien en CAPEX [1] qu'en OPEX [2] [MMJ08]. Dans le système très complexe qu'est le réseau, il est presque impossible d'avoir une vision globale de l'infrastructure, des services, des risques de panne, ce qui rend la gestion des pannes très coûteuse en OPEX. Les systèmes de gestion des pannes nécessitent des ressources réseaux redondantes pour transporter le trafic en situation de panne, et leur nature statique oblige les opérateurs à sur-dimensionner leur infrastructure afin d'être capable de faire face au maximum de situations, ce qui est très couteux.

La configuration des systèmes de gestion des pannes est un choix difficile entre coût et performance à la charge de l'opérateur lors de la mise en place d'un nouveau service.

Problématique

Dans un environnement réseau très complexe avec de multiples couches et des technologies différentes la gestion de réseau est devenue très difficile. La taille de ces réseaux a explosé et atteint la limite de traitement du cerveau humain. Dans cet environnement, la gestion des pannes est encore plus concernée car elle implique des temps de traitement extrêmement sensibles au temps. Les opérateurs s'appuient sur les mécanismes intégrés aux protocoles tels qu'OSPF ou GMPLS mais leur nature réactive est leur principale faiblesse. La probabilité de panne évolue dans le temps alors que la nature statique de ces mécanismes les confine à ne pas évoluer dans le temps. En effet, les délais de réaction de l'ordre de la seconde requis par la gestion des pannes ne permettent pas à des opérateurs humains de prendre des décisions pertinentes en si peu de temps. En conséquence la gestion des pannes de manière réactive n'est pas optimale car elle implique une indisponibilité trop importante pour le trafic *premium*, ou alors au détriment d'un coût beaucoup trop important.

Objectifs

L'objectif de cette thèse est de perfectionner l'efficacité de la gestion des pannes afin d'améliorer la disponibilité fournit par les opérateurs tout en se préservant d'augmenter les coûts, c'est-à-dire en utilisant les ressources de manière plus efficace qu'aujourd'hui.

Une solution est l'intégration d'une gestion autonome des réseaux au sein même des équipements, afin que ceux-ci puissent observer, analyser et prendre des actions de réparation de manière plus efficace qu'un opérateur humain. La capacité supérieur de traitement de l'information en quantité et en rapidité permet d'envisager des actions proactives, en anticipation des pannes, afin que celles-ci n'aient presque plus d'incidences sur les services fournis par l'opérateur à ses clients.

Contributions

Après un état de l'art sur la gestion autonome des pannes, la première contribution de cette thèse est la proposition d'une architecture autonome dédiée à la gestion des pannes UAFAReS [3] intégrée à l'architecture de gestion autonome GANA [4][WTVL13, TGV09, RNP+11, CPT+10, VNR+10, CPP+10, PKM+09, PKA+10, VCL+11, TDQ+09] élaborée en collaboration avec les partenaires du projet EFIPSANS [5]. En effet, la gestion autonome devient une nécessité pour des réseaux de plus en plus complexes, et la gestion des pannes, qui est un paramètre critique pour les opérateurs, requiert une attention toute particulière. L'architecture UAFAReS développée dans le Chap. 1 prend en compte cette problématique et exploite la rapidité d'intervention dont est

1. Dépenses d'investissement de capital (*CApital EXpenditure*)
2. Dépenses d'exploitation (*OPerational EXpenditure*)
3. *Unified Architecture for Autonomic Fault-Management, Resilience and Survivability*
4. *Generic Autonomic Network Architecture*
5. *Exposing the features in IP version six protocols that can be exploited/extended for the purposes of designing/building autonomic networks and services*

capable l'autogestion pour permettre l'implémentation de fonctions d'autoréparation proactives qui anticipent les pannes pour mieux les gérer.

Mes contributions ont plus particulièrement portées sur le nouveau cadre architectural de ce dernier aspect qui a permis trois propositions originales de mécanismes d'autoréparation proactifs qui sont étudiés dans les Chap. 2, 3 et 4 :

- un mécanisme d'adaptation dynamique de la fréquence du protocole *Hello* utilisé pour la détection des pannes dans les IGP [VGD11, VGDNR13]. Ce mécanisme permet, lorsque qu'un risque important de panne est détecté, de réduire le temps de détection d'une panne. L'indisponibilité s'en trouve réduite, tout en conservant une fréquence d'envoi des messages *Hello* acceptable pour la stabilité réseau ;
- un mécanisme de routage sensible au risque de pannes qui modifie dynamiquement les métriques des liens ayant une forte probabilité de panne afin de dévier le trafic vers des d'autres équipements [RNP+11, VC10, CPP+10, KAB+11, VNCR13]. La panne peut alors se produire sans aucune incidence sur le trafic ;
- un mécanisme de changement en temps réel du dispositif de résilience du protocole GMPLS entre la protection et la restauration en fonction du risque de pannes [VNC12, WTVL13, PKM+09, PKA+10, VCL+11]. Ce mécanisme permet un faible taux d'utilisation des ressources lorsqu'aucun risque de panne n'est détecté, et la mise en place du dispositif de protection assurant une haute disponibilité lorsque le risque de panne est élevé.

Cette thèse décrit ces trois mécanismes proactifs et en évalue l'intérêt pour les réseaux de cœur IP, tout en identifiant l'impact de l'incertitude du calcul du risque de pannes sur le comportement des mécanismes proposés. Pour cela, une première étude analytique est effectuée pour chaque mécanisme, puis celle-ci est confrontée à une implémentation dans un simulateur. Dans le cas du mécanisme de routage sensible aux pannes, un test de faisabilité est aussi réalisé sur un prototype expérimental.

Publications, brevets et rapports techniques

Les travaux développés dans cette thèse ont été l'objet de publications, d'un brevet et de rapports scientifiques pour le projet européen EFIPSANS dont les références sont disponibles ci-dessous.

Publications

–

– [WTVL13] Michal Wodczak, Nikolay Tcholtchev, Bruno Vidalenc et Yuhong Li, « **Design and Evaluation of Techniques for Resilience and Survivability of the Routing Node** », *International Journal of Adaptive, Resilient and Autonomic Systems, IJARAS*, Volume 4(3),IGI Global, 2013.
[VNC12] Bruno Vidalenc, Ludovic Noirie et Laurent Ciavaglia, « **GMPLS Adaptive Level of Recovery** », *Proceedings of the 47th IEEE International Conference on Communications, ICC*, Ottawa, Canada, Juin 2012, page 2768-2773.
– [RNP+11] Gabor Retvari, Felician Nemeth, Arun Prakash, Ranganai Chaparadza, Ibrahim Hokelek, Mariusz Fecko, Michal Wodczak et Bruno Vidalenc , « **A Guideline for Realizing the Vision of Autonomic Networking : Implementing Self-Adaptive Routing on top of OSPF** », *Book chapter in Formal and Practical Aspects of Autonomic Computing and Networking*, IGI Global, 2011.
– [VC10] Bruno Vidalenc et Laurent Ciavaglia, « **Proactive fault management based on risk-augmented routing** », *Proceedings of the IEEE Global Telecommunications Conference, GLOBECOM Workshops*, Miami, USA, Décembre 2010, page 481-485.
– [TGV09] Nikolay Tcholtchev, Monica Grajzer et Bruno Vidalenc, « **Towards a unified architecture for resilience, survivability and autonomic fault-management for self-managing networks** », *In Proceedings of the 2009 international conference on Service-oriented computing, Mona+*, LNCS, Stockholm, Sweden, 2009, page 335-344.

- [VGDNR13] Bruno Vidalenc, Samir Ghamri-Doudane Ludovic Noirie et Eric Renault, « **Adaptive Failure Detection Timers for IGP Networks** », *IFIP/IEEE International Symposium on Integrated Network Management, IM*, Ghent, Belgium, Mai 2013, *Soumis*.
- [VNCR13] Bruno Vidalenc, Ludovic Noirie, Laurent Ciavaglia et Eric Renault, « **Dynamic Risk-Aware Routing for OSPF Networks** », *IFIP/IEEE International Symposium on Integrated Network Management, IM* , Ghent, Belgium, Mai 2013, *Soumis*.

Brevet

- [VGD11] Bruno Vidalenc et Samir Ghamri-Doudane, « **Method for detecting failure of a router** », EP2363982, *Brevet européen*, Alcatel-Lucent, Publié le 07 septembre 2011.

Rapports techniques

Les rapports ci-dessous sont les rapports publics du projet européen ICT FP7 EFIPSANS dont je suis co-auteur.

- [CPA+10] Ranganai Chaparadza, Symeon Papavassiliou, Giorgos Aristomenopoulos, Timotheos Kastrinogiannis, Mary Grammatikou, Christos Argyropoulos, Zhaojun Li, Michal Wodczak, Mariusz Fecko, Yuhong Li, Wang Wendong, Nikolay Tcholtchev, Arun Prakash, Razvan Petre, Kevin Quinn, Alan Davy, John Ronan, Juan Manuel Gonzalez, Tasos Zafeiropoulos, Athanassios Liakopoulos, Vassilios Kaldanis, Thomas Scherer, Rolland Vida, Felician Nemeth, Gabor Retvari, Slawomir Kuklinski, Martin Vigoureux, Pierre Peloso et Bruno Vidalenc : « **Final version of autonomic behaviours specifications (ABs) for the diverse networking environments** », *Rapport scientifique du projet FP7 EFIPSANS*, INFSO-ICT-215549/EFIPSANS/WP1/D1.7/Part1, 2010.
- [CPT+10] Ranganai Chaparadza, Arun Prakash, Nikolay Tcholtchev, Razvan Petre, Symeon Papavassiliou, Giorgos Aristomenopoulos, Timotheos Kastrinogiannis, Mary Grammatikou, Christos Argyropoulos, Vassilis Merekoulias, Zhaojun Li, Michal Wodczak, Monika Grajzer, Tomasz Zernicki, Yuhong Li, Wang Wendong, Xiangyang Gong, Yan Shi, Xin Li, Kevin Quinn, Aland Davy, Lei Shi, Juan Manuel Gonzalez, Vassilios Kaldanis, Slawomir Kuklinski, Krzysztof Cabaj, Szymon Lis, Krzysztof Szczypiorski, Laurent Ciavaglia et Bruno Vidalenc : « **Final version of the specification and description tables for decision elements (DEs)** », *Rapport scientifique du projet FP7 EFIPSANS*, INFSO-ICT-215549/EFIPSANS/WP1/D1.7/Part3, 2010.
- [VNR+10] Bruno Vidalenc, Felicián Németh, Gábor Rétvári, Zoltan Theisz, Zhaojun Li, Ranganai Chaparadza, Nikolay Tcholtchev, Giorgos Aristomenopoulos, Mary Grammatikou, Timotheos Kastrinogiannis, Symeon Papavasileiou, Mariusz Fecko, Michal Wodczak, Monika Grajzer, Juan Manuel Gonzalez, Thomas Scherer, Vassilios Kaldanis, Alan Davy, Kevin Quinn, John Ronan et Slawomir Kuklinski : « **Final version of requirements specifications (RQs) regarding features, required in IPv6 protocols, network architectures and paradigms, in order to implement the specified autonomic behaviours** », *Rapport scientifique du projet FP7 EFIPSANS*, INFSO-ICT-215549/EFIPSANS/WP1/D1.8, 2010.
- [CPP+10] Ranganai Chaparadza, Arun Prakash, Razvan Petre, Alexej Starschenko, Felician Nemeth, Michal Wodczak, Monika Grajzer, Tomasz Zernicki, Slawomir Kuklinski, Jacek Wytrebowicz, Pawel Radziszewski, Bruno Vidalenc et Laurent Ciavaglia : « **Overall IPv6++ (EFIPSANS extended IPv6) feature combination scenarios for engineering autonomicity** », *Rapport scientifique du projet FP7 EFIPSANS*, INFSO-ICT-215549/EFIPSANS-/WP2/D2.5, 2010.
- [PKM+09] Symeon Papavassiliou, Timotheos Kastrinogiannis, Juan Manuel Gonzalez, Zhaojun Li, Ranganai Chaparadza, Petre Razvan, Nikolay Tcholtchev, Michal Wodczak, Grajzer Monika, Yuhong Li, Sheila Becker, Thorsten Ries, Xiangyang Gong, Yan Shi, Xin Li, Wang Wendong, John Ronan, Tasos Zafeiropoulos, Athanassios Liakopoulos, Vassilios Kaldanis, Rolland Vida, Gabor Vincze, Bruno Vidalenc, Giorgos Aristomenopoulos

et Eleni Tsiropoulou : « **Advanced network services in autonomic IPv6 networking : Performance analysis and evaluation** », *Rapport scientifique du projet FP7 EFIPSANS*, INFSO-ICT-215549/EFIPSANS/WP3/D3.2, 2009.

– [PKA+10] Symeon Papavassiliou, Timotheos Kastrinogiannis, Giorgos Aristomenopoulos, Eleni Tsiropoulou, Juan Manuel Gonzalez, Zhaojun Li, Nikolay Tcholtchev, Michal Wodczak,Grajzer Monika, Yuhong Li, Wang Wendong, John Ronan, Vassilios Kaldanis, Rolland Vida, Gabor Vincze, Peter Schaffer, Thorsten Ries et Bruno Vidalenc : « **Advanced network services in autonomic IPv6 networking : Architectures'performance analysis and evaluation** », *Rapport scientifique du projet FP7 EFIPSANS*, INFSO-ICT-215549/EFIPSANS-/WP3/D3.6, 2010.
– [VCL+11] Bruno Vidalenc, Laurent Ciavaglia, Yuhong Li, Xin Li, Lei Zhou, Nikolay Vassilev Tcholtchev, Ranganai Chaparadza et Michal Wodczak : « **Resilience, survivability and/for autonomicity in IPv6 networks** », *Rapport scientifique du projet FP7 EFIPSANS*, INFSO-ICT-215549/EFIPSANS/WP3/D3.4, 2011.
– [TDQ+09] Nikolay Tcholtchev, Alan Davy, Kevin Quinn, John Rohnan, Bruno Vidalenc, Bela Berde, Monika Grajzer, Agnieszka Cavalcante, Ranganai Chaparadza, Constantinos Marinos, Christos Argyropoulos et Cynthia Wagner : « **Components and mechanisms for autonomic Fault-Management** », *Rapport scientifique du projet FP7 EFIPSANS*, INFSO-ICT-215549/EFIPSANS/WP4/D4.5, 2009.
– [KAB+11] Vassilios Kaldanis, Domonkos Asztalos, Peter Benko, Zoltan Bacskay, Csaba Simon, Felicián Németh, Ferenc Ficsor, Wen-dong Wang, Xiangyang Gong, Xin Li, Shiduan Cheng, Yuhong Li, Alexej Starschenko, Arun Prakash, Ranganai Chaparadza, Nikolay Tcholtchev, Petre Razvan, Andras Zahemszky, Patrik Teppo, Monika Grajzer, Tomasz Zernicki, Mary Grammatikou, Timotheos Kastrinogiannis, Vassilis Merekoulias, Giorgos Aristomenopoulos, Constantinos Marinos, Zhaojun Li, Mick Wilson, Krzysztof Cabaj, Sheila Becker, Bruno Vidalenc, Anastasios Zafeiropoulos, Thanassis Liakopoulos, Slawomir Kuklinski, Lukasz Podkalicki, Pawel Radziszewski et Michal Ulaski : « **Final version of EFIPSANS demonstration results** », *Rapport scientifique du projet FP7 EFIPSANS*, INFSO-ICT-215549/EFIPSANS/WP5/D5.3, 2011.

Organisation de la thèse

Après cette brève introduction, le Chap. 1 est consacré à l'étude des concepts architecturaux nécessaires à l'introduction de fonctions d'autoréparation dans les réseaux. En effet, il est déraisonnable de penser que les fonctionnalités autonomes peuvent être déployées dans les réseaux d'aujourd'hui. Un socle architectural est nécessaire afin d'orchestrer et d'encadrer la multitude de fonctions autonomes vers des objectifs communs. Après un tour d'horizon des travaux relatifs aux architectures pour les réseaux autonomes, la nouvelle architecture UAFAReS dédiée à la gestion des pannes est définie au sein de l'architecture GANA. Cette architecture définit les blocs fonctionnels permettant le déploiement de dispositifs d'autoréparation proactifs tels que ceux développés dans les chapitres suivants. Le Chap. 2 propose un premier dispositif qui utilise le risque de panne en temps réel pour adapter la fréquence d'envoi des messages *Hello* de détection de pannes des protocoles IGP dans les réseaux IP. Une modélisation analytique ainsi qu'une implémentation du mécanisme dans un simulateur sont ensuite appliquées à trois topologies de réseau de cœur afin d'en étudier le comportement. Le deuxième mécanisme d'autoréparation s'applique aussi au routage IP, en modifiant de manière temporaire les métriques associées à des éléments de réseaux risqués. Ce mécanisme est étudié dans le Chap. 3 au travers d'un modèle analytique et d'une implémentation dans un simulateur mais aussi grâce à l'implémentation d'un prototype expérimental. Enfin, le Chap. 4 s'attache au routage orienté connexion en proposant d'adapter dynamiquement le dispositif de résilience du protocole GMPLS. Ce troisième mécanisme est d'abord étudié de manière théorique, puis de manière pratique dans un simulateur. La conclusion permet de faire le point sur les contributions de la thèse et d'envisager les futurs axes à explorer.

Chapitre 1

Considérations architecturales pour l'autoréparation

Sommaire

Bien que l'un des objectifs principal dans la réalisation de tout système informatique ai toujours été l'automatisation de celui-ci, force est de constater qu'aujourd'hui la situation est telle qu'il est urgent de passer à la vitesse supérieur. En effet, l'évolution fulgurante des systèmes informatiques et des systèmes de télécommunications lors de ces vingt dernières années a abouti à des systèmes tellement complexes que leur gestion est devenue un vrai cauchemar. Pour remédier à ce problème, le concept de système autonome, se gérant soi-même, est maintenant exploré par une partie de la communauté de recherche depuis une dizaine d'années [Hor01]. Les réseaux de télécommunications, en bien des points similaires aux systèmes informatiques, envisagent aussi une gestion autonome, afin notamment, de réduire les coûts de gestion en constante progression.

Dans cet objectif de gestion autonome des réseaux, la gestion des pannes, où chaque seconde est comptée, n'échappe pas à la règle, puisque l'efficacité et la rapidité des actions que pourrait entreprendre un système autonome permettraient des performances bien meilleures que ce que pourrait entreprendre un opérateur humain. Dans cette optique, la définition d'une architecture fonctionnelle, incluant la gestion des pannes est un prérequis indispensable, afin de fédérer, et donner une certaine cohérence aux multiples fonctions autonomes qui pourraient être introduites dans les réseaux.

1.1 Des réseaux de télécommunications autonomes

La vue d'ensemble des réseaux de télécommunications ressemble à un agglomérat complexe où des réseaux multicouches et multi-technologies sont interconnectés ensembles. La complexité des réseaux de télécommunications est une réalité mais certainement pas un avantage, obligeant les fournisseurs de réseaux à dépenser beaucoup d'énergie dans le seul but de maîtriser les coûts opérationnels dûs à la gestion de leur réseau. Le problème général de la complexité des réseaux ne cesse de croitre, à cause notamment, de leur hétérogénéité, de la nature incrémentale des technologies en jeu, avec l'entassement de couches multiples et de l'interconnexion de dizaines de protocoles différents. Bien que le modèle multicouche soit à l'origine du fondement de l'Internet, l'hétérogénéité des technologies impliquées aboutie sur de nombreux problèmes de compatibilité et d'interconnexion. Ceci est illustré par le nombre de protocoles qui régissent l'Internet avec notamment plus de 6000 IETF [1] RFCs. Avec les réseaux d'anciennes générations, les réseaux WDM [2], SDH [3], Ethernet, ATM [4], IP, MPLS et MPLS pour ne citer qu'eux, le nombre et les spécificités des technologies déployées sont directement responsables de la complexification de la gestion de réseau. Enfin la course à la réduction des coûts a entrainée des choix d'équipements hétérogènes venant de différents constructeurs, augmentant sensiblement la complexité du système dans son ensemble.

La taille est aussi un élément critique, puisqu'avec de plus en plus d'utilisateurs connectés, la taille des réseaux explose, complexifiant encore plus la donne, au point de dépasser la capacité de gestion humaine. Malheureusement, la gestion actuelle centralisée, telle que celle définie par l'architecture TMN [TMN96], qui centralise toutes les fonctionnalités de gestion au sein du *Network Management System* (NMS), accentue les difficultés, car dans un environnement, toujours plus partitionné composé de différents niveaux tels que les domaines, les aires, les systèmes autonomes, une gestion globale cohérente est devenue impossible. Le risque est donc de perpétuer un système de gestion, plus que complexe et sous-optimal.

Le trafic réseau est en constante augmentation, et de la même manière, le trafic sensible à la qualité de service augmente, au minimum, dans les mêmes proportions. Les trafics *best effort* (BE), VoIP [5], vidéo et P2P [6] nécessitent des traitements spécifiques de gestion de qualité de service, mais il est de plus en plus difficile, au sein des éléments de réseaux, de différencier ces trafics et de déterminer la manière de les gérer. Quand la complexité de la gestion du trafic, des différentes technologies, et les problèmes de taille des réseaux s'additionnent, la gestion de

1. *Internet Engineering Task Force*
2. *Wavelength-division multiplexing*
3. *Synchronous optical networking*
4. *Asynchronous Transfer Mode*
5. *Voice over IP*
6. *Peer to Peer*

réseau devient un véritable casse-tête. Dans ces conditions, l'optimisation des performances du réseau et de l'ingénierie de trafic n'en est que plus difficile, voire impossible.

La gestion de réseau est aujourd'hui opérée par des opérateurs humains, qui, bien qu'extrêmement qualifiés, sont bien plus lents, coûteux et sujets aux erreurs que ne pourrait l'être une machine. Une étude intéressante [Sur02] relevait en 2002 que 62% du temps d'arrêt des réseaux d'entreprise résultait d'erreurs humaines. De plus, le temps de réaction peut être un facteur critique dans la gestion de réseau, tout particulièrement lorsqu'un évènement inattendu survient, tel qu'une panne, car un délai de réaction trop lent peut annihiler le bénéfice des actions de correction, voire même les rendre contre-productives. Il ne fait aucun doute que la gestion de réseau devient trop complexe pour être effectuée efficacement par des opérateurs humains et nécessiterait une gestion automatique, qui serait largement distribuée, plus sûre, plus réactive, et donc plus efficace. Parallèlement, le déploiement de nouveaux services nécessite aujourd'hui une flexibilité et une rapidité bien plus importante, mettant en exergue le problème de la complexité comme un frein à l'évolution de l'infrastructure réseau et des services associés. De plus, la complexité augmentant, les besoins d'expertise augmentent aussi, entrainant un accroissement des coûts d'OPEX et une grande difficulté pour trouver du personnel possédant le niveau d'expertise demandé.

Dans un marché très compétitif, les fournisseurs d'accès sont à la recherche de différenciateurs tels que la qualité des services qu'ils proposent et non la gestion de réseau qui ne génère pas de bénéfice. Sachant que les coûts d'OPEX représentent 60% des bénéfices des opérateurs [Inf08], la volonté des opérateurs, de vouloir réduire ceux-ci grâce à une gestion automatisée des leurs opérations répétitives et ennuyeuses, est d'autant plus compréhensible. Une transition vers l'autogestion des réseaux libèrerait l'opérateur de toutes ces barrières et lui permettrait de se concentrer sur le déploiement de nouveaux services à plus forte valeur ajoutée.

En réponse, l'objectif affiché des réseaux autonomes est de fournir un environnement opérationnel digne de confiance et passant à l'échelle, qui se chargerait de gérer et cacher cette complexité pour l'opérateur. La gestion décentralisée, le partage de connaissances ainsi que l'auto-organisation sont les caractéristiques essentielles dans la construction d'un processus collaboratif qui permettait de maintenir le réseau en phase avec la stratégie globale désirée par l'opérateur de réseau.

1.1.1 La gestion de réseau

L'essor de l'Internet et des réseaux ayant rapidement mis l'accent sur le besoin d'utiliser des mécanismes solides de gestion, l'accroissement continu des réseaux a sans cesse remis en question les systèmes de gestion en vigueur. Mais qu'est exactement la gestion de réseau ? Cette activité regroupe un grand nombre d'interventions de la part de l'opérateur qui ont pour but de maintenir le bon fonctionnement du réseau et des services qu'il fournit. Elle regroupe donc les mécanismes de dimensionnement, d'allocation de ressources, de configuration de l'infrastructure et des services, d'observation, de mesure de l'activité de celui-ci, de maintenance, de diagnostic et de réparation des anomalies, ainsi que les processus d'optimisation des réglages permettant de tirer le meilleur parti des ressources. D'un autre point de vue, on peut observer que durant ces vingt dernières années, l'évolution des systèmes de gestion ont eu pour objectif de répondre aux principales préoccupations des opérateurs à savoir [BX02] :

- la configuration et la gestion des ressources et des services ;
- L fiabilité et la sécurité de fonctionnement ;
- des informations de gestion les plus simples et générales possibles ;
- des coûts d'OPEX limités.

Bien que les systèmes aient évolués, ils ne satisfont toujours pas les opérateurs [WL09]. C'est pourquoi ceux-ci placent de grands espoirs dans les réseaux autonomes, afin de régler, tous les défauts des systèmes actuels. Voyons donc brièvement quels sont ces standards de management qui sont en vigueur actuellement dans les réseaux de télécommunications.

1.1.1.1 Les standards de la gestion de réseau

Plusieurs organismes de standardisation tels que l'IETF, l'ISO [1],l'ITU-T [2], le DMTF [3] ont proposé des solutions à la gestion de réseaux, chacun avec ses prôpres spécificités.

ISO

Le standard le plus ancien est le standard ISO [ISO89, ISO98] dont une des parties définit la gestion de réseau *via* cinq modèles :

- *le modèle fonctionnel* définit les domaines fonctionnels de la gestion de réseau. Ces domaines sont connus sous l'acronyme FCAPS (voir la Fig. 1.1) pour *Fault*, *Configuration*, *Accounting*, *Performance* et *Security* . Le domaine de la configuration regroupe les actions de configuration et de contrôle des équipements pour assurer leur bon fonctionnement. Le domaine des pannes comprend les étapes de détection, de localisation, de réparation, ainsi que l'analyse des pannes, nécessaires à l'opérateur pour assurer une certaine disponibilité à ses clients. Le domaine lié à la comptabilité s'occupe de tout ce qui est coût en CAPEX, et facturation. Le domaine s'attachant à la performance, a pour but l'optimisation des ressources et des services réseau, conformément aux objectifs définis par l'opérateur. Enfin, la sécurité regroupe tout ce qui est lié à la politique de sécurité de l'opérateur ainsi que la mise en place de celle-ci ;

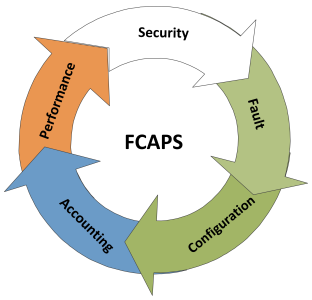


Figure 1.1: Les fonctions de gestion de réseaux FCAPS.

- *le modèle architectural* définit les entités impliquées dans les activités de gestion ainsi que leurs relations respectives ;
- *le modèle organisationnel* définit le concept de base, impliquant un gestionnaire et un agent. Le gestionnaire est l'entité logicielle envoyant les ordres de gestion à l'agent qui se trouve au sein de la ressource gérée et qui se charge d'exécuter les commandes reçues. Dans la pratique le manager est souvent une entité centrale contrôlant de multiples agents se trouvant au sein des équipements du réseau ;
- *le modèle informationnel* définit le modèle d'information représentant et caractérisant les ressources réseau sous forme d'objets. À ces objets sont associés des attributs de description de la ressource, les actions de gestion autorisées, les notifications que peut envoyer l'objet lors de modifications de celui-ci ainsi que la définition complète du comportement de l'objet. Ces objets sont stockés au sein des équipements dans une base de données appelée MIB [4], que l'on retrouve dans la plupart des équipements de réseau aujourd'hui ;
- *le modèle de communication*, quant à lui, définit le protocole de communication CMIP [5] entre le gestionnaire et l'agent.

Bien que les concepts du standard OSI soient encore utilisés aujourd'hui, le standard laisse de nombreux points sans réponse et a surtout été définit pour le monde informatique. C'est pourquoi d'autres organismes de standardisation ont tenu à ajouter leurs contributions.

1. *International Organization for Standardization*
2. *International Telecommunication Union - Telecommunication Standardization Sector*
3. *Distributed Management Task Force*
4. *Management Information Base*
5. *Common Management Information Protocol*

TMN

L'organisme de standard ITU-T a lui aussi tenu à apporter sa contribution, en proposant le standard TMN [1] [TMN96] spécifique aux contraintes des réseaux de télécommunications. Contrairement au standard ISO, ce standard est entièrement dédié à la gestion des équipements et des services réseau. L'architecture définie est décomposée en trois parties, une *architecture fonctionnelle* que nous allons détailler par la suite, une *architecture physique* définissant l'implémentation de l'architecture fonctionnelle dans les équipements de télécommunication et une *architecture informationnelle* similaire au modèle informationnel du standard ISO. L'architecture fonctionnelle est composée de quatre éléments de base, l'OSF [2], le NEF [3], le WSF [4] et le TF [5]. Toutes ces entités sont reliées entre elles par des interfaces de type différents (q, f, x, g et m) suivant le type de commande qui transite sur ces interfaces.

- les fonctions du système d'exploitation (OSF) sont les fonctions centrales qui contrôlent, observent et traitent les opérations de gestion en agissant sur les éléments de réseau (NEF). On retrouve donc ici le modèle gestionnaire/agent du standard ISO ;
- les fonctions des éléments de réseau (NEF) sont les fonctions exposées par l'agent de gestion de l'élément réseau à l'OSF pour que celui-ci puisse le contrôler ;
- les fonctions du poste de travail (WSF) permettent à l'opérateur d'observer l'état de son réseau et d'accéder au service de gestion que propose l'OSF ;
- enfin, les fonctions de transformation (TF) permettent de connecter deux entités du système de gestion ayant des protocoles de communication ou d'information différents.

De plus, une décomposition hiérarchique (voir la Fig. 1.2) des responsabilités de gestion a été définie. La couche la plus basse est la couche de gestion des éléments du réseau (*Element Management Layer*) et concerne la gestion d'un, voire plusieurs, équipements. Viens ensuite la couche où l'entité de base est le réseau tout entier (*Network and sustems Management Layer*). On passe ensuite à des visions plus axées sur l'activité économique de l'opérateur avec la gestion des services que fournit le réseau à ses clients (*Service Management Layer*) et enfin la gestion de l'activité de l'entreprise (*Business Management Layer*) qui concerne les aspects de stratégie économique et commerciale.

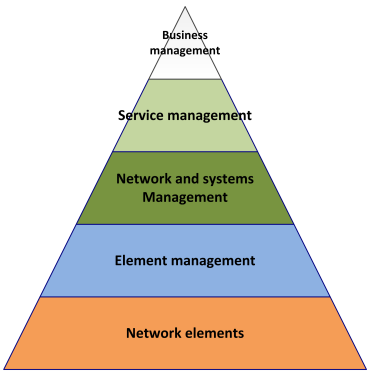


FIGURE 1.2: Gestion hiérarchique avec TMN.

Bien que le standard TMN soit spécifique au monde des télécommunications, et qu'il aille plus loin dans les spécifications d'implémentation, les grands principes architecturaux restent similaires au modèle ISO en bien des points et notamment sur sa nature très centralisée s'appuyant sur un unique gestionnaire (OSF) qui, du fait de la complexité des réseaux d'aujourd'hui, ne permet plus d'effectuer les tâches de gestion de façon optimale.

1. *Telecommunications Management Network*
2. *Operations System Function*
3. *Network Element Function*
4. *Workstation Function*
5. *Transformation Function*

SNMP

Le standard SNMP [1] [CFSD90], certainement le plus connu et le plus utilisé dans la gestion de réseaux, est bien moins complet que les deux standards décrits précédemment, car il se concentre sur la définition du protocole de communication entre les équipements de gestion et les équipements du réseau, ainsi que sur le modèle informationnel de représentation des éléments de réseau et de leur gestion. En effet, contrairement aux standards ci-dessus, le standard SNMP fournit une implémentation précise pour le modèle gestionnaire/agent, avec pour commencer, la définition d'un protocole de communication permettant d'envoyer, de récupérer ou de modifier les valeurs stockées dans les bases de donnée des équipements, utilisés pour leur gestion (MIB) [RM91]. Ces communications peuvent être à l'initiative du gestionnaire, sous forme de requêtes (messages *GetRequest*, *GetNextRequest*, *SetRequest*), ou bien à l'initiative de l'agent lors de changements d'état spécifiques (message *Trap*). D'autre part, le modèle d'information [RM90] (SMI [2]) utilisé dans les MIB est décrit précisément et repose sur des objets possédant un identifiant unique (OID) et utilisant la syntaxe ASN.1 pour le typage des données ainsi que l'encodage BER pour relier les informations à un objet.

Une fois encore, la gestion se fait de manière complètement centralisée, avec un gestionnaire SNMP qui contrôle les équipements réseau en utilisant les agents SNMP présents sur chaque équipement comme un moyen d'accès aux fonctions de contrôle de ces derniers. Pour limiter ce problème, l'IETF a proposé plusieurs évolutions et notamment RMON [3] [WCKR03] qui délègue certaines fonctionnalités de collecte et d'analyse d'information à des agents RMON placés dans les équipements. Ces agents RMON sont des sondes configurables par des filtres, qui agissent tels des *proxys* entre l'équipement et le gestionnaire SNMP, afin de ne remonter que les informations pertinentes et donc de décharger le travail du gestionnaire dans les équipements. Néanmoins, cette amélioration n'a fait que repousser le problème, car la complexité actuelle nécessiterait l'intégration de bien plus de fonctionnalités de gestion au sein des équipements. Par la suite, les spécifications de SNMPv2 [CMRW93] ont continué à s'attacher au problème du passage à l'échelle en introduisant des gestionnaires intermédiaires. Avec cette évolution, lorsqu'une même commande SNMP concerne plusieurs équipements, une seule requête est envoyée au gestionnaire intermédiaire, celui-ci se chargeant ensuite d'appliquer cette requête à tous les équipements spécifiés par le gestionnaire principal. Similairement, le gestionnaire intermédiaire réassemble toutes les réponses (*Bulk*) pour n'envoyer qu'un seul message au gestionnaire principal. Cette décomposition hiérarchique permet donc, un peu à la manière du routage *multicast*, de réduire sensiblement le nombre d'échanges de messages, mais conserve tout de même la grande majorité des fonctions de gestion au sein du gestionnaire principal. Enfin la dernière version en date, SNMPv3 [CMPS02] cible les aspects de sécurisation qui avaient été omis jusque-là, afin de fournir des dispositifs d'authentification et de cryptage indispensables pour une fonctionnalité aussi critique que la gestion de réseaux.

WBEM

Bien que SNMP soit très répandu, il existe bien d'autres protocoles de management tels que Netconf [Enn06], Q3/CMIP, ainsi qu'une multitude de protocoles propriétaires, qui font des réseaux multi-vendeurs d'aujourd'hui, de véritables casses-têtes pour les opérateurs souhaitant une vue unifiée de leur réseau. L'approche WBEM [4] a pour objectif d'unifier et de mettre à jour les protocoles de gestion, vers une version utilisant des technologies orientées WEB. Le standard propose donc un nouveau modèle d'information générique, le CIM [5] qui est un modèle objet permettant, en plus de la description des éléments de réseau, de spécifier les relations entre ces éléments. Enfin, conformément à la philosophie du standard, les technologies WEB sont utilisées pour le transport des messages avec le protocole HTTP [6] et pour l'encodage des informations

1. *Simple Network Management Protocol*
2. *Structure of Managed Information*
3. *Remote MONitoring*
4. *Web-Based Enterprise Management*
5. *Common Information Model*
6. *Hypertext Transfer Protocol*

avec le langage XML [1]. Les efforts pour se détacher des spécificités de gestion de chaque élément de réseau sont ici une avancée intéressante, d'autant plus que le modèle d'information sera à la base de futurs modèles d'information utilisés dans des propositions d'architecture de réseaux autonomes [SAL06]. Néanmoins l'utilisation d'un gestionnaire central reste un défaut important face à la situation des réseaux de télécommunications actuels.

1.1.2 Origine de l'autonomie

1.1.2.1 Systèmes informatiques autonomes

L'évolution des systèmes informatiques, aussi bien sur le plan matériel que sur le plan logiciel, a, depuis le début, toujours été soutenue. Malheureusement les outils et méthodes de management ne permettent plus de gérer correctement des systèmes aussi complexes. Ce constat est à l'origine d'une nouvelle approche s'inspirant du fonctionnement des systèmes biologiques qui ont pour tâche de gérer des systèmes bien plus complexes que les systèmes informatiques, par exemple le système nerveux humain. Cette analogie a débouché sur l'émergence d'une nouvelle thématique de recherche inspirée de l'intelligence artificielle et qui a pour but la construction de systèmes autonomes, décidant eux-mêmes des actions à mener, en observant, optimisant constamment leur états et en s'adaptant aux conditions changeantes qui les entourent. Cette autogestion nécessite donc de multiples fonctions autonomes telles que l'auto-configuration, l'auto-optimisation, l'auto-protection, l'autoréparation, l'auto-gouvernance, l'auto-adaptation, l'auto-organisation, et l'autodiagnostic pour ne citer que les plus importantes, qui sont les fondements du concept de système informatique autonome, à l'origine développé au laboratoire de recherche d'IBM [Hor01], il y a de cela une dizaine d'années. Parmi ces fonctionnalités autonomes, quatre aspects sont plus régulièrement mis en valeur par IBM : l'auto-configuration qui comprend les processus de configuration et de reconfiguration du système afin que celui-ci s'adapte à l'environnement changeant ; l'auto-optimisation qui s'applique à observer l'état du système afin de réajuster l'allocation des ressources et de remplir au mieux les objectifs haut niveau ; l'autoréparation en charge de détecter, diagnostiquer et réparer de manière autonome les pannes ou fautes du système ; et l'autoprotection qui se charge de défendre le système contre des attaques malveillantes, en identifiant, voire anticipant les attaques et en déterminant les mesures de sécurité adéquates permettant de minimiser les dommages de l'attaque. L'autre composante essentielle d'un système autonome est sa nature fortement distribuée [KC03], où chaque élément autonome est décomposé comme présente sur la Fig. 1.3, avec un gestionnaire autonome (*Autonomic Manager*) et une ressource contrôlée (*Managed Resource*).

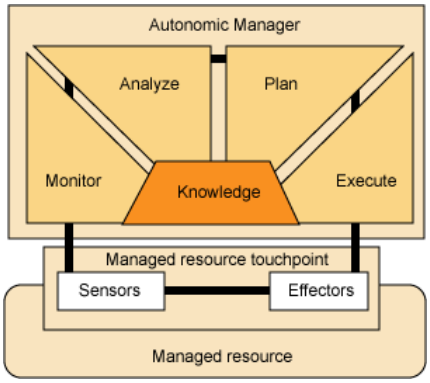


FIGURE 1.3: Modèle MAPE-K d'IBM.

Les autres composantes de ce schéma fondateur sont le capteur qui permet au gestionnaire d'observer l'état de la ressource et l'actionneur qui permet au gestionnaire d'influer sur le comportement de la ressource. Ces quatre éléments forment donc une boucle de contrôle au sein de laquelle les fonctions d'observation, d'analyse, de planification et d'exécution réalisées par le gestionnaire permettent de gérer la ressource contrôlée dès que son état le nécessite. Les

1. *Extensible Markup Language*

paragraphes suivants définissent plus en détail, les quatre fonctions centrales de l'architecture MAPE-K [1], au sein desquelles la connaissance joue un rôle clé.

Observation

L'observation est la première étape consistant à collecter *via* l'interface du capteur, des informations précises sur l'état interne de l'élément contrôlé telles que des statistiques sur la charge du processeur, la consommation mémoire, le nombre d'accès disque, etc, pour ensuite agréger, filtrer et corréler toutes ces informations et obtenir une information plus riche, nécessaire dans l'analyse de la situation.

Analyse

L'étape d'analyse se base sur les observations collectées pour examiner en détail la situation, et déterminer si des changements sont nécessaires. C'est dans cette étape que des mécanismes de raisonnement venant de l'intelligence artificielle ou de l'apprentissage basé sur des modèles statistiques tels que les chaînes de Markov, les réseaux bayésiens, ou les séries temporelles permettent une analyse bien plus rapide et précise que ne pourrait le faire un être humain. L'utilisation de ces dispositifs d'apprentissage permet la construction de modèles comportementaux permettant de mieux comprendre la situation et d'anticiper son évolution afin de décider si le système nécessite une intervention. Si c'est le cas, on passe alors à l'étape de planification.

Planification

La planification est l'étape de prise de décision et de construction d'une stratégie plus ou moins complexe, permettant à la ressource contrôlée de remplir les objectifs haut niveau définis par l'administrateur du système. Cette planification peut se faire sous forme de règles de politique, ou de commandes bien spécifiques, qui seront ensuite transmises à l'étape d'exécution.

Exécution

L'exécution comprend l'ordonnancement et la mise en place des commandes à envoyer à la ressource contrôlée afin d'exécuter au mieux la stratégie définie dans l'étape de planification. Pour cela, le gestionnaire utilise l'interface de l'actionneur. Le résultat de ce plan d'exécution permet ensuite de venir enrichir la base de connaissances partagée par toutes les étapes de cette boucle de contrôle afin de servir lorsqu'une situation similaire se représentera.

Connaissance

La connaissance est une notion centrale dans toutes les étapes précédemment décrites, que ce soit lors de la construction d'une information plus riche résultant de la corrélation des observations, de l'utilisation d'apprentissage basé sur l'historique, de la prise en compte de la stratégie de l'administrateur. Elle permet, grâce à l'échange de cette connaissance entre les différents gestionnaires de ressources, d'améliorer le fonctionnement global de ces systèmes autonomes, de manière collaborative.

1.1.2.2 L'autonomie appliquée aux réseaux de télécommunications

De manière similaire aux systèmes informatiques, les réseaux de télécommunications ont connu une progression constante, avec, au fil des ans la démocratisation de l'Internet, de l'utilisation des technologies sans-fil, des *smartphones*, de l'accès aux contenus multimédia, ce qui a aboutie aux mêmes conséquences, à savoir, des réseaux de télécommunications trop complexes, difficilement gérables, et avec une évolution très limitée. L'exemple de la gestion des pannes est symptomatique des limites du mode de gestion actuel. Malgré la collecte de nombreux indicateurs sur l'état du réseau, l'interdépendance et la quantité de ces indicateurs rendent la détection, mais surtout la recherche de l'origine d'une panne, un véritable calvaire pour un opérateur humain. De plus, la stratégie à adopter pour rapidement isoler et réparer une panne est une tâche complexe à effectuer par un opérateur humain, qui nécessite donc du personnel

1. *Monotoring, Analyse, Planification, Execution - Knowledge*

ultra-qualifié, et où l'urgence dans laquelle s'effectuent de telles interventions peut aboutir à des erreurs qui compliqueront encore plus une tâche qui l'est déjà bien assez. C'est la raison pour laquelle, les besoins de réactivité, de fiabilité et de maitrise des coûts, nécessitent l'exploration des techniques de gestion autonome appliquées aux réseaux. Cela permettra d'adapter automatiquement la configuration du réseau à l'état de l'infrastructure et du trafic afin de remplir les objectifs haut niveau définis par l'opérateur. Ce nouvel axe de recherche dédié aux réseaux autonomes, dont le rapport de S. Dobson et al. [DDF+06] détaille les particularités, reprend les mêmes principes que pour les systèmes informatiques, avec les fonctions d'auto-configuration, d'auto-optimisation, d'autoréparation et d'autoprotection, mais adaptées au monde des réseaux, où le système est beaucoup plus distribué et où les problématiques de passage à l'échelle sont renforcées. Les notions de collaboration et de coordination ont donc une importance plus critique qu'avec les systèmes informatiques. Le but ultime étant de décharger l'opérateur des tâches répétitives de gestion pour qu'il se focalise sur des taches à plus forte valeur ajoutée, où seuls des objectifs de haut niveau sont définis, pour ensuite être dérivés automatiquement, en fonction du contexte, en politiques de configuration pour les différents équipements du réseau.

1.1.3 Les principes fondateurs de l'autonomie dans les réseaux

Bien que les fondements des réseaux autonomes soient proches des systèmes informatiques autonomes, il existe des spécificités qui caractérisent les réseaux autonomes.

1.1.3.1 Boucle d'adaptation

La boucle d'adaptation, qui est au cœur même de la notion d'autonomie, reprend dans le cadre de la gestion de réseau, les mêmes éléments que ceux définis par IBM. Cependant, la nature distribuée des réseaux, introduit une notion hiérarchique plus importante, où l'élément contrôlé peut être un équipement réseau, mais aussi un ensemble d'équipements, une aire de routage, voire même un domaine regroupant plusieurs centaines d'équipements. Cette boucle d'adaptation, a pour objectif d'optimiser le fonctionnement de la ressource contrôlée, afin que celle-ci participe à l'effort commun de réponse aux objectifs haut niveau de l'opérateur. Ces objectifs fixés par l'opérateur sous forme de politiques (voir la Fig. 1.4), peuvent aussi provenir de décisions prisent par des boucles d'adaptation de niveau supérieur.

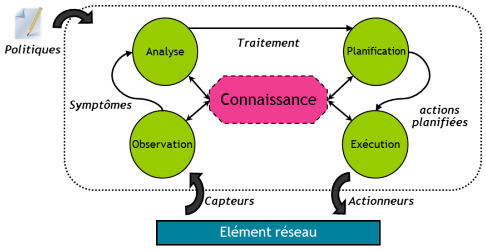


FIGURE 1.4: Boucle de rétroaction autonome pour les réseaux.

Mais l'optimisation d'un réseau soulève de nombreux problèmes, le nombre de ressources, de paramètres, d'interdépendances entre les ressources et ses paramètres, rend l'optimisation globale très complexe, notamment lors de la compétition entre des objectifs contradictoires venant de domaines différents. Compte tenu de la nature des réseaux, composés de centaines d'éléments distribués hétérogènes dont l'interdépendance est très complexe, des boucles de contrôle doivent être présentes un peu partout au sein des équipements du réseau, formant de ce fait, une macro-boucle invisible, distribuée dans tout le réseau.

1.1.3.2 Une gestion distribuée

La gestion des réseaux a toujours été effectuée de manière très centralisée au sein d'un unique gestionnaire de réseau (NMS [1]), mais la taille et la complexité des réseaux actuels rend

1. *Network Management System*

ce mode de gestion inadapté. Le passage à l'échelle de la gestion de réseaux nécessite donc un mode de gestion autonome intégrant des algorithmes distribués, le défi étant de garantir une gestion optimale et cohérente basée sur des informations locales et incomplètes. La gestion de la cohérence entre les différents agents distribués est un enjeu majeur afin de garantir une certaine stabilité, indispensable au bon fonctionnement du réseau. Enfin, la tolérance aux pannes d'un tel système est aussi une donnée importante pour la viabilité d'un système de gestion autonome.

Les systèmes multi-agents

Pour la mise en place et le fonctionnement d'un système distribué de gestion, une bonne partie de la communauté de recherche s'appuie sur les systèmes multi-agents tels l'approche proposée dans « *Autonomous Agent for Autonomic Networks* »[BZG+06] ou dans « *Multi-agent based autonomic architecture for network management* »[Tia03]. Le principe est d'associer un agent autonome à un élément du réseau, par exemple un domaine, un routeur, une interface, etc, et où, grâce aux informations partielles auxquelles cet élément a accès, il sera en mesure de résoudre certains problèmes ou sous-problèmes pour lesquels il aura été défini. Chaque agent est donc autonome, ne nécessite aucune orchestration centralisée forte, et contribue, à son échelle, à résoudre un problème qu'aucun agent n'aurait pu résoudre individuellement. Les caractéristiques fondamentales des agents telles que l'autonomie, leur capacité de communication entre eux, leur réactivité par rapport aux événements et aux stimuli extérieurs ainsi que leur capacité à pouvoir agir proactivement en prenant des initiatives, en font des outils indispensables pour un réseau autonome.

1.1.3.3 Les politiques

Les politiques, au travers du concept de gestion par politiques (PBM [1]), ont pour objectif de simplifier la reconfiguration des éléments de réseau. La définition d'un modèle et d'un langage global de gestion des ressources permet d'automatiser l'exécution des décisions de haut niveau venant de l'opérateur *via* un système de gestion par politiques, en configuration propre à chaque équipement et à chaque technologie. En effet, la maitrise de la configuration de chaque élément du réseau est impossible par un opérateur humain, ce qui est un souci majeur. L'utilisation de politiques permet de simplifier la tâche, en définissant des objectifs simples, qui se traduisent automatiquement, sans intervention humaine, en configurations complexes au sein des équipements. C'est un des axes forts développés par Strassner et al. [SAL06] dans leur architecture de gestion autonome.

1.1.3.4 La connaissance

La connaissance est au centre de l'architecture autonome d'IBM, et l'est aussi dans sa transposition aux réseaux de télécommunications comme le prône D. Clark dans son article « *A Knowledge Plane for the Internet* »[CPRW03]. En effet, avec la distribution des mécanismes de gestion dans les équipements réseau, la problématique d'accès à l'information devient réelle car, contrairement à un système centralisé regroupant toutes les informations en un seul point, un système distribué nécessite des dispositifs d'accès et de partage d'information afin de rendre le maximum d'information accessible au plus grand nombre d'équipements. Des contraintes de cohérence des données, d'actualisation, de temps d'accès, de confiance en l'information doivent être levées afin de fournir un plan de connaissance comme outil de partage des connaissances au sein du système de gestion de réseau autonome. Ce plan de connaissance, viens s'ajouter au traditionnel plan de données qui comprend tous les mécanismes responsables de l'acheminement des paquets au travers du réseau, et au plan de contrôle qui comprend les mécanismes de contrôle du plan de données, à savoir, la réservation des ressources, le routage, la détection des pannes, l'activation de nouveaux services, etc. Même si les approches diffèrent dans la façon d'instancier ce plan de connaissance, il existe un consensus sur la nécessité de proposer des mécanismes de partage et d'accès à la connaissance standardisés [SFL+08], afin que chaque agent autonome

1. *Policy Based Network Management*

puisse prendre ses décisions avec un maximum d'information en sa possession. Malheureusement, la taille des réseaux d'aujourd'hui rend la quantité d'information à traiter gigantesque. Il est donc nécessaire d'ajouter des dispositifs de filtrage, d'agrégation et d'apprentissage afin de construire, à partir d'informations brutes, des connaissances utiles et adaptées à la prise de décision des agents autonomes.

L'apprentissage

La connaissance construite automatiquement par les systèmes autonomes, se fait grâce à un processus que l'on appelle l'apprentissage automatique, car l'explosion combinatoire de tous les états du système rend leur description trop complexe avec les langages de programmation classiques. Il est donc nécessaire de se doter de mécanismes capable apprendre des événements passés pour construire dynamiquement un modèle de comportement du système évolutif dans le temps, permettant d'accélérer, d'améliorer et d'adapter la prise de décision à l'évolution du système dans le temps. L'apprentissage s'appuie sur des techniques combinant l'intelligence artificielle, la théorie des probabilités et des statistiques ainsi que l'optimisation. Il existe plusieurs types d'apprentissage plus ou moins supervisés, et de nombreuses approches algorithmiques telles que les arbres de décision, les règles d'association, les réseaux de neurones, la programmation génétique, la programmation logique inductive, le partitionnement des données, les machines à vecteurs de support, les réseaux bayésiens et l'apprentissage par renforcement pour les plus couramment utilisés. Cet apprentissage permet ensuite un traitement basé sur l'expérience, afin de réagir au mieux, aux changements d'état du réseau, voire même de détecter, en avance de phase, des modèles de comportement permettant de prendre des actions proactives, comme proposées dans les chapitres 2, 3 et 4 de cette thèse pour la gestion des pannes.

1.1.3.5 Les fonctions de base

Les quatre fonctions définies par IBM dans [Hor01] sont facilement transposables aux problématiques des réseaux et constituent, comme pour les systèmes informatiques autonomes, les fonctions de base que se doit de proposer un système de gestion de réseaux autonomes.

Auto-configuration

La fonction d'auto-configuration, pourrait être vue comme la fonction la plus attendue par les opérateurs, car elle permet d'économiser des coûts substantiels pour des tâches bien souvent répétitives, et sujettes à de nombreuses erreurs d'inattention. La composition hétérogène des équipements et des différentes technologies composant un réseau, rend la configuration de celui-ci complexe, longue, couteuse et source de nombreuses erreurs. Cette tâche répétitive, nécessite néanmoins du personnel hautement qualifié et très coûteux, pour une tâche à faible valeur ajoutée. Un système d'auto-configuration permet à l'opérateur de ne spécifier que des règles générales de configuration *via* un système de politiques, qui sont ensuite dérivées de façon transparente, en configurations, au sein des équipements et services réseau. Ces politiques expriment des objectifs simples de la part de l'opérateur, à charge ensuite au système d'auto-configuration d'en déduire la stratégie pour les remplir et les configurations adéquates, permettant le déroulement des étapes planifiées par le système, pour remplir les objectifs globaux. Ce procédé permet d'éviter le grand nombre de mauvaises configurations responsables de nombreuses indisponibilités des services réseau actuels, notamment lors des mises à jour, de réduire grandement les délais de déploiement d'un nouveau service, ou de l'ajout de nouveaux équipements au sein d'une infrastructure existante, et enfin, permet une réduction substantielle des coûts d'OPEX et une utilisation plus intelligente du potentiel du personnel qualifié, pour d'autres fonctions à plus forte valeur ajoutée.

Auto-optimisation

Le réseau est un système complexe en perpétuelle mutation, que ce soit les variations de longue durée comme l'ajout de matériel, l'activation et la désactivation de nouveaux services, l'interconnexion avec d'autres réseaux, la mise à jour du système d'exploitation des équipements, ou les variations plus courtes comme celles générées par une panne, une activité de maintenance,

une intrusion informatique ou un pic de trafic. Mais les réseaux sont aujourd'hui conditionnés pour un fonctionnement « en moyenne » optimal, avec une configuration statique, basée sur les estimations des opérateurs, et où toute ré-optimisation est la plupart du temps impossible car limitée par la capacité de traitement des opérateurs humains. En effet, aussi qualifiés soient-ils, leurs délais de réaction ne sont pas en phase avec les délais que requièrent les changements d'état du réseau. Compte tenu de la complexité des réseaux, seuls des mécanismes automatiques, basés sur des fonctions d'apprentissage statistiques et tirant parti de la puissance de calcul et de la rapidité d'exécution des équipements, peuvent permettre une optimisation continue des paramètres et ressources du réseau en fonction de son état au temps t. En effet, la quantité de données énorme à traiter pour évaluer l'état du réseau, la complexité de résolution des problèmes d'optimisation, et la nécessité d'agir vite, ne sont possibles que grâce à un système d'auto-optimisation effectué par des agents intelligents et autonomes disséminés dans les équipements réseau.

Autoréparation

La disponibilité et la qualité de service assurées par un opérateur est aujourd'hui un des points critiques au cœur de la concurrence que se livrent les différents fournisseurs de services réseau. Néanmoins, assurer une disponibilité et une QoS importantes implique des investissements importants dans le système de gestion des pannes, de la redondance, et beaucoup d'experts hautement qualifiés pour diagnostiquer, dans cet environnement complexe, l'origine des anomalies. Dans cette tâche où la réactivité est essentielle, l'utilisation de la capacité d'analyse de milliers d'alarmes et de paramètres, de la rapidité d'exécution, et de la rapidité d'investigation et de raisonnement des machines, fait de l'autoréparation un très bon candidat pour l'introduction de fonctions autonomes dans les réseaux, car une entité capable d'analyser continuellement le système pour détecter et localiser les pannes, et ensuite proposer des actions correctives dans un temps réduit, serait sans comparaison avec la gestion actuelle. De plus, grâce à l'apprentissage, la prédiction des pannes est envisageable, ce qui laisse la porte ouverte à de nouveaux mécanismes proactifs capables d'anticiper les pannes pour mieux les gérer, tels que ceux proposés dans cette thèse.

Autoprotection

Similairement aux anomalies, les capacités d'analyse et la rapidité d'exécution font de l'autoprotection la fonctionnalité rêvée par les opérateurs pour se défendre contre les attaques de pirates informatiques ou contre les programmes malveillants. L'intervention humaine apportant de longs délais néfastes à cette tâche, l'utilisation d'un mécanisme autonome capables de se défendre lui-même contre les attaques, voire même de reconnaitre des modèles de comportement pour les anticiper, permettrait de mieux gérer ce type de situation, tout en réduisant les coûts générés par les interventions d'experts en sécurité, qui sont coûteux et difficiles à trouver.

1.2 Architectures fonctionnelles pour les réseaux autonomes

Les succès des télécommunications et de l'Internet ont pour origine des modèles de référence bien établis, comme les modèles ISO ou TCP/IP, dont les concepts sous-jacents sont relativement bien maitrisés par l'ensemble des personnes concernées. Cela a permis l'implémentation de systèmes interopérables, permettant de ce fait, le développement d'infrastructures multi-vendeurs à moindre coût. Malheureusement, étant donné que le concept de réseau autonome est relativement nouveau, aucun des modèles architecturaux actuellement en place n'a été conçu pour supporter l'autogestion. Il est donc nécessaire de définir un nouveau modèle architectural prenant en compte les problématiques d'autogestion.

L'introduction de l'autonomie au cœur des réseaux n'est pas une chose aisée et nécessite une approche cohérente, c'est pourquoi les diverses initiatives concernant les réseaux autonomes ont proposé des architectures fonctionnelles avec chacune leurs spécificités. En effet la définition d'une architecture est le moyen indispensable de définir d'une manière unifiée l'organisation des

processus autonomes, les fonctionnalités et responsabilités de chaque composant, qui permettent de donner un cadre favorisant le déploiement de l'autonomie dans nos réseaux.

Les sections suivantes décrivent quelques une des dernières architectures en date ainsi que leurs propositions pour introduire de l'autonomie dans les réseaux.

1.2.1 4D

L'architecture 4D est la plus ancienne des architectures présentées dans cet état de l'art, et est l'une des premières architectures proposées pour palier aux défauts du système de gestion actuel. Bien que A. Greenberg [GHM+05] ne mentionne jamais le concept de réseau autonome, l'architecture proposée peut être considérée comme l'une des premières architectures autonomes de par les idées qu'elle développe. Les principes simples de l'architecture qui en font sa force, sont quatre plans (décision, dissémination, découverte et données comme indiqué par la Fig 1.5) qui tranchent avec la complexité des systèmes actuels. Cette architecture fait table rase du

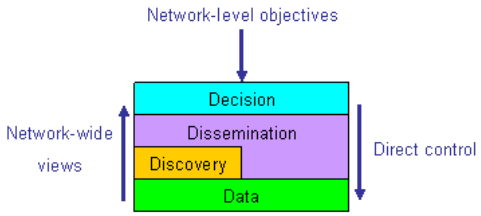


FIGURE 1.5: Architecture 4D.

système actuel en proposant une architecture où les routeurs, commutateurs et autres éléments de réseau, ne conservent que leur fonction de transfert de paquets, et forme le plan de données (*Data plane*).

Toute l'intelligence de gestion et de contrôle du réseau se trouvent dans des agents de décision (*Decision Element*) qui forment le plan de décision (*Decision Plane*). On notera qu'un plan de décision constitué d'agents distribués et de plusieurs niveaux hiérarchiques peut aussi être défini pour répondre à des problèmes de passage à l'échelle. Ces agents de décision fournissent donc toutes les fonctions, de gestion du routage, de contrôle d'accès, de sécurité, etc, afin de configurer les équipements du plan de données. Ces éléments du plan de données, dénués de toute intelligence, se cantonnent donc à transférer les données en suivant les configurations envoyées par les agents de décision.

Le plan de dissémination (*Dissemination Plan*) fait le lien entre les agents de décision et les éléments de transfert de paquets, en fournissant des mécanismes robustes de communication entre ces deux plans. Le plan de données ne pouvant pas fonctionner sans le contrôle du plan de décision, ce plan de dissémination est donc dissocié du plan de données, afin d'être opérationnel dès l'étape de configuration du plan de données.

Enfin, le plan de découverte (*Discovery Plane*) est responsable de la découverte des capacités et caractéristiques des équipements, ainsi que de leurs relations respectives, dans le but de les fournir au plan de décision. Cela inclut la découverte des interfaces, des capacités de ces interfaces, des relations de voisinage, etc, et l'attribution d'identifiants permettant aux agents de décision de les manipuler.

Ces quatre plans sont aussi au service de principes de gestion promouvant une gestion par l'opérateur, non plus de chaque élément de réseau séparément, mais du réseau comme élément unitaire. Cela se traduit par la configuration du réseau en utilisant des objectifs et des politiques appliqués à celui-ci, et à la prise en compte de l'importance d'une vue complète et efficace de l'état du réseau. Enfin, l'élimination des mécanismes de gestion indirects induis par les protocoles de contrôle distribués tels qu'OSPF, en faveur d'un contrôle directe par le plan de décision est un des objectifs avoué vers une simplification et un meilleur contrôle du réseau.

Bien que cette proposition puisse paraître utopique vis-à-vis de l'état actuel des réseaux, et souffrir de nombreux manques par rapport aux architectures ci-dessous, sa simplicité est un atout important et l'une des raisons pour laquelle l'architecture GANA, décrite dans la Sec. 1.4, s'en est inspirée.

1.2.2 FOCALE

FOCALE pour *Foundation, Observation, Compare, Act, Learn*, et *rEason* est l'architecture de gestion autonome proposée par les laboratoires de recherche de Motorola et du *Waterford Institute of Technology* [SAL06]. L'architecture s'articule autour de trois principes fondateurs : le modèle d'information DEN-ng [1] l'utilisation d'ontologies et la gestion par politiques.

Le modèle d'information DEN-ng est un modèle d'information qui décrit les aspects *business* et système d'une entité gérée, ainsi que les relations entre elles. Basé sur le modelé CIM défini par l'organisme DMTF, DEN-ng est un modèle objet utilisant des machines à états finis pour modéliser l'état actuel d'une ressource réseau, ainsi que son comportement.

Les éléments gérés étant différents les uns des autres, l'architecture propose une couche de traduction (voir la *Model based translation layer* sur la Fig. 1.6) afin de convertir les modèles de données et les commandes spécifiques à chaque équipement en un même modèle générique commun. C'est là qu'intervient l'utilisation d'ontologies, afin de découvrir les similarités, les concepts communs et de les traduire vers un même modèle d'information, en l'occurrence DEN-ng. La gestion des ressources de types différents est donc faite de manière similaire par le gestionnaire, pour ensuite être traduite vers le langage propre à chaque équipement.

L'importance de la gestion par politiques est affichée par la présence d'un serveur de politiques à tous les niveaux de l'architecture (voir Fig. 1.6), afin de fournir à l'opérateur le moyen de contrôler le réseau et aux gestionnaires autonomes un moyen de contrôler des ressources ou d'autres agents autonomes.

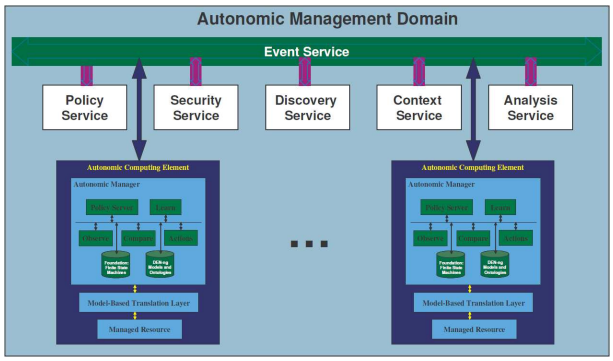


Figure 1.6: Architecture FOCALE.

La Fig. 1.6 représente les différents blocs fonctionnels présents dans chaque agent autonome. Ces agents autonomes distribués au plus près des équipements suivent le modèle gestionnaire/ressource contrôlée, et implémentent la boucle de contrôle autonome afin de prendre en compte l'état du réseau dans le processus de gestion.

L'architecture autonome FOCALE, bien qu'assez complète, met fortement l'accent sur le rôle du modèle d'information dans la construction de la connaissance, délaissant malheureusement d'autres points et notamment la définition des interfaces de communication ainsi que des recommandations technologiques pour une implémentation.

1.2.3 ANA

L'architecture ANA [2] [BJT+10] propose l'introduction d'abstractions génériques ainsi que de primitives de communication permettant la construction d'éléments réseau et de réseaux qui soient à la fois flexibles, dynamiques et entièrement autonomes. Les blocs architecturaux illustrés sur la Fig. 1.7 sont au nombre de cinq :

- le *network Compartment* regroupe toutes les parties d'un réseau appartenant au même contexte ;
- l'*information channel* (IC) fournit une abstraction vers le service de communication à l'intérieur d'un *network compartment* ;

1. *Directory Enabled network - Next generation*
2. *Autonomic Network Architecture*

– le *functional block* (FB) est une abstraction du composant logiciel interne à l'élément du réseau qui implémente une fonctionnalité telle que la génération, le traitement ou le transfert de données ;
– l'*information dispatch point* (IDP) est un point d'entrée vers le FB auquel est relié un IC. Un FB peut posséder plusieurs IDP qui peuvent être ajoutés ou supprimés dynamiquement ;
– enfin, le *node compartment* est l'élément de niveau nœud qui ressemble au *network compartment* mais sans la possibilité de fournir d'IC.

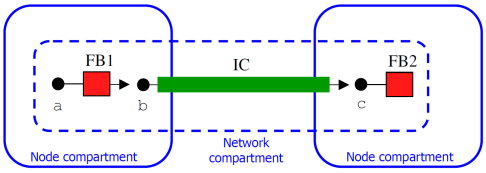


FIGURE 1.7: Architecture ANA.

L'architecture propose donc des concepts novateurs mais ressemble plus à une méta-architecture, laissant vide des aspects tels que le système d'information, les mécanismes de construction de la connaissance, la définition des interfaces et des protocoles de communication ainsi que le système de politiques.

1.2.4 INM

L'architecture INM [DBN+09] pour *In Network Management* a été proposée dans le cadre du projet européen 4WARD. Elle a comme point fort, sa capacité d'évolution graduelle, avec la prise en compte des différents niveaux d'évolution des systèmes actuels vers le système INM optimal. En effet, l'architecture orientée service (SOA [1]) est composée d'entités INM mais aussi d'entités non-INM. Une entité INM est un élément de réseau supportant les fonctionnalités INM alors qu'une entité non-INM est un élément de réseau comme il en existe actuellement, c'est-à-dire sans les fonctionnalités INM. Alors qu'une entité non-INM nécessite une intervention extérieure pour être gérée, le principe d'une entité INM est la gestion de l'élément réseau par lui-même, possédant en son sein, les algorithmes de gestion et les fonctions de monitoring, d'adaptation et d'autogestion. Le fait que chaque élément de réseau possède une couche de gestion INM, permettant aux équipements, de manière collaborative, d'effectuer les actions de gestion nécessaires, sans intervention extérieure, a pour résultat un système de gestion décentralisé, auto-organisé et autonome.

La Fig. 1.8 décrit les principaux blocs fonctionnels de l'architecture INM. Elle est composée principalement d'un *Global Management Point* (GMP) et de *Self-managing Entities* (SEs), possédant chacune, une ou plusieurs *Management Capabilities* (MCs).

Le *global management point* est la porte d'accès de l'opérateur par laquelle il peut fixer des objectifs haut niveau au réseau sous forme de politiques, et observer en temps réel, à quel point ses objectifs sont remplis. Les politiques sont ensuite envoyées aux différents domaines de gestion, puis aux SEs qui se chargent de les exécuter.

Les *self-managing entities* sont des éléments orientés service, qui offrent au sein même des équipements, les fonctionnalités d'autogestion et d'auto-configuration ainsi que des interfaces de communication avec le GMP ou d'autres équipements.

Enfin, chaque fonction ou chaque algorithme de gestion est embarqué au sein d'une ME. Chaque SME contient un ou plusieurs MCs pouvant collaborer entre eux pour fournir les comportements autonomes attendus.

Cette architecture qui prône l'incorporation de fonctions de gestion au plus proche des équipements, a accordé un intérêt tout particulier aux fonctions d'auto-adaptation et de monitoring. Afin de connaitre l'état du réseau en temps réel, des métriques de niveau réseau, des méthodes statistiques sur l'estimation des groupes (*group size estimation*), des algorithmes de découverte

1. *Service Oriented Architecture*

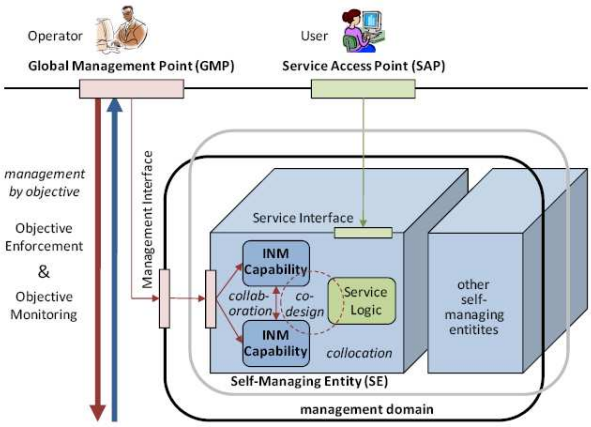


Figure 1.8: Architecture INM.

de voisins et des détections d'anomalies de façon distribuée (*algorithme en arbre* [WDS08], *Gossip* [WDSC07], *Nato* [CL09], *Hide and Seek* [GSSM10]) sont utilisés afin de respecter les contraintes temporelles que nécessite la fonction de *monitoring* temps réel. Grâce à cette fonction de monitoring temps réel, l'auto-adaptation est naturellement plus efficace mais de nouvelles techniques inspirées des sciences chimiques sont aussi proposées afin d'incorporer une étape de prédiction dans la boucle autonome d'auto-adaptation.

1.2.5 AutoI

L'architecture AutoI [GDB+09] pour *Autonomic Internet* se démarque des autres architectures par la prise en compte des problématiques de gestion inter-domaine et de réseau virtuel. L'architecture est basée sur cinq plans d'abstraction :

- le *plan de virtualisation* permet de gérer toutes les problématiques liées à la virtualisation des ressources physiques afin de permettre la migration et la reconfiguration à la volée des ressources réseau ;
- le *plan de management* est responsable de la création et de la gestion des multiples boucles autonomes ainsi que des fonctions qui la composent. Des règles de politique et le contexte sont ici pris en compte dans le processus de gestion autonome ;
- le *plan activateur* de service s'attache au déploiement des services réseau en utilisant les ressources virtuelles fournies par le plan de virtualisation ;
- le *plan de connaissance* sert de base de données distribuée afin de fournir à chacun des autres plans les informations qu'il désire en utilisant un modèle d'information commun et des ontologies ;
- le *plan d'orchestration* est en quelque sorte un super-plan de management qui orchestre la gestion des différents domaines, leurs interactions, et qui joue le rôle de médiateur lors d'éventuels conflits. Son rôle est de déterminer la stratégie globale qui satisferait tous les domaines et d'agir sur les plans de management afin de réaliser cet objectif.

Cette architecture utilise bien des concepts communs avec les autres architectures autonomes mais se démarque avec l'utilisation de la virtualisation et du plan d'orchestration pour s'affranchir de l'hétérogénéité des réseaux.

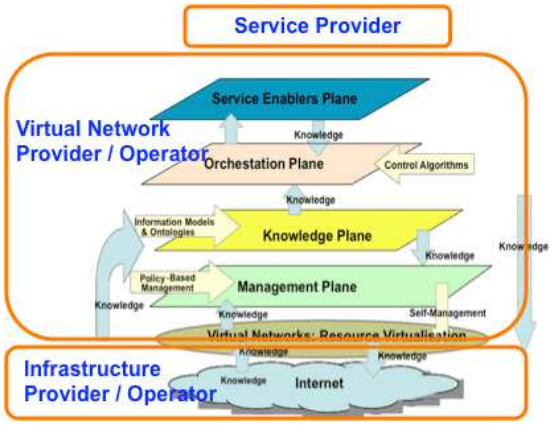


FIGURE 1.9: Architecture AUTOI.

1.2.6 CASCADAS

L'architecture du projet européen CASCADAS [1] [MZ06, BDFM+09] s'articule autour de la notion de service, afin de gérer, de la même manière, les domaines informatique, télécom et Internet. Cette architecture est composée de composants légers, distribués dans le réseau, qui ont pour tâche de fournir un vue abstraite des ressources en exposant celles-ci sous forme de service. Le composant principal de cette architecture est appelé élément de communication autonome (ACE [2]) et permet de construire tout service de base pour l'autonomie. Mais la collaboration et la composition dynamique de plusieurs ACE peut aussi être utilisée pour des services plus complexes, grâce à un *framework* proposant des fonctionnalités d'auto-découverte et d'auto-organisation. Un ACE est lui-même composé d'organes qui définissent son comportement. Il existe des organes facilitateurs, exécuteurs, répertoires de fonctionnalités, gestionnaires, passerelles et de supervision comme illustré par la Fig. 1.10.

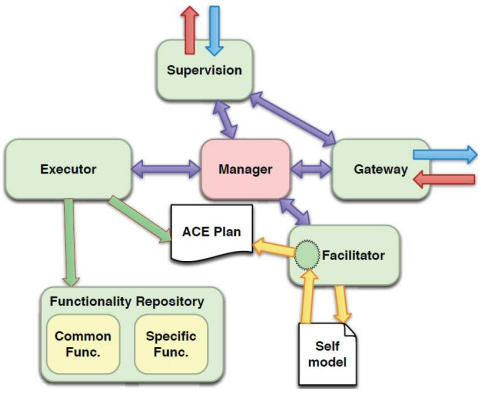


FIGURE 1.10: Architecture interne d'un ACE.

L'approche de CASCADAS se démarque en définissant un modèle de composant simple, unique et dont la composition permet de fournir l'ensemble des fonctionnalités de la gestion autonome de réseau.

1.2.7 Self-NET

L'architecture Self-NET [3] [MNBA09b, MNBA09a] prône l'introduction de fonctionnalités cognitives pour l'autogestion des réseaux. Pour cela une approche hiérarchique est proposée, avec un gestionnaire cognitif d'éléments de réseau (NECM [4]) et un gestionnaire cognitif de domaines

1. *Componentware for Autonomic, Situation-aware Communications and Dynamically Adaptable Services*
2. *Autonomic Communication Elements*
3. *Self-Management of Cognitive Future InterNET Elements*
4. *Network Element Cognitive Manager*

réseau (NDCM [1]) de niveau supérieur, comme le représente la Fig. 1.11. Ces éléments sont des agents logiciels fournissant des fonctionnalités de monitoring et d'exécution afin d'implémenter la boucle de rétroaction autonome pour les éléments du réseau.

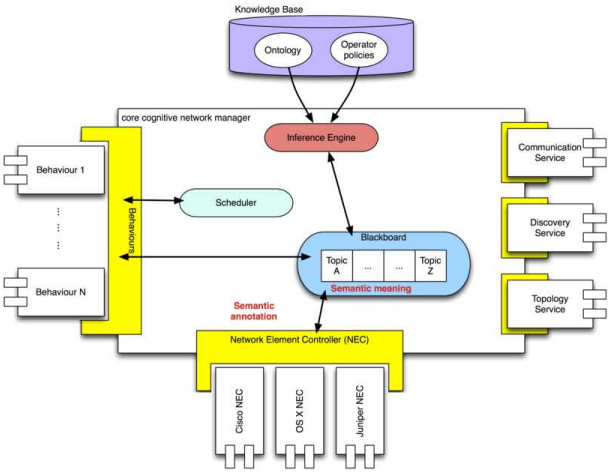


Figure 1.11: Architecture Self-NET.

Le NECM utilise les connaissances locales ou celles disponibles, grâce aux autres NECM, dans sa vue située pour la résolution de problème. Dans le cas où cela ne serait pas suffisant, il fait appel au NDCM. Le NDCM est aussi un composant cognitif, mais de niveau supérieur ayant une vue plus globale du réseau et des services, au travers de ses NECM. Il est donc en charge de résoudre des problèmes plus globaux, que les NECM ne sauraient traiter.

La représentation des connaissances de ces éléments cognitifs utilise le langage d'ontologie OWL [2] ainsi que le langage de règle SWRL [3] pour les connaissances *a priori*, alors qu'un moteur d'inférence de règles floues est utilisé pour l'identification d'évènements. Mais, alors que la coopération et la coordination entre les différents gestionnaires cognitifs est un aspect central de l'architecture, le protocole de communication est un point qui n'a malheureusement pas été défini.

1.2.8 SerWorks

L'architecture SerWorks [DP07] provient du projet BIONETS [4] dont la thématique diffère des autres projets sur les réseaux autonomes, en s'inspirant très fortement des systèmes biologiques. L'architecture orientée service, supporte des opérations très dynamiques rarement envisagées, comme la génération à la volée de protocole réseau, ou encore la désactivation d'une partie du réseau en fonction des besoins. L'architecture est divisée en trois *frameworks*, comme illustré sur la Fig. 1.12.

La couche supérieur, le *framework service*, regroupe les mécanismes applicatifs et de service, ainsi que les fonctions nécessaires à leur gestion et leur exécution autonome distribuée. Dans ce plan, l'entité *mediator* est responsable de la gestion autonome des services.

Le *framework* intermédiaire d'*interaction* est responsable des communications entre les services et leur *mediator*, ainsi qu'entre les *frameworks* service et réseau, en fournissant des modèles d'interactions concurrentes pour les communications entre les services distribués, et en fournissant un espace de données partagé. Une version simplifiée du *framework* interaction est proposée pour les petits équipements et les capteurs ayant des limites de puissance, sous la forme d'une entité *IF footprint*.

Le *framework réseaux*, assure les fonctions de communication de base, qui sont, dans le cadre du projet, des connexions sécurisées sur des réseaux opportunistes ou des routes orientées connexion sur des réseaux IP.

1. *Network Domain Cognitive Manager*
2. *Web Ontology Language*
3. *Semantic Web Rule Language*
4. *BIOlogically inspired NETwork and Services*

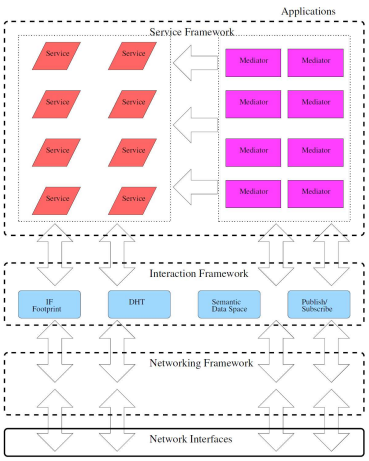


FIGURE 1.12: Architecture SerWorks.

Cette architecture initialement construite pour les réseaux de capteurs, est aussi valable pour d'autres types de réseau et nous donne un bon aperçu de ce qu'une approche inspirée de la biologie peut apporter.

1.2.9 Ginkgo

L'architecture Ginkgo [GPSZ06, BKH+08] est un architecture qui laisse une place centrale au concept multi-agents et au plan de connaissance. Bien que l'architecture soit moins complète et complexe que les architectures susmentionnées, c'est une des rares implémentations commerciales d'architecture pour la gestion autonome des réseaux. Même s'il existe bien d'autres outils et systèmes multi-agents comme Jade [Jad11] par exemple, la plateforme Ginkgo est spécifiquement conçue pour les réseaux et les fonctions d'autonomie. C'est la raison pour laquelle elle a été choisie pour le développement du prototype détaillé au Chap. 3 (Cf Sec. 3.10).

Cette architecture est composée d'un système multi-agents où chaque équipement de réseau est associé à un agent. Le contrôle, la collaboration et la coordination se font ensuite uniquement grâce au plan de connaissance qui joue ici un rôle essentiel et indispensable. Le plan de connaissance fonctionne sur le modèle de la vue située, c'est-à-dire qu'un agent possède seulement les informations venant d'un sous-ensemble de voisins plus ou moins proches, essentiellement pour des raisons de passage à l'échelle et de récence de l'information.

L'architecture d'un agent est illustrée en Fig. 1.13 [Ger10]. Les actions de l'agent sont définies par des *behaviors* qui sont des blocs fonctionnels indépendant au sein de l'agent, exécutant des fonctionnalités différentes, pouvant collaborer par l'intermédiaire d'une base de données locale et étant orchestrées par le planificateur dynamique de l'agent. Tous les *behaviors* ont donc accès à l'élément de réseau contrôlé au travers des interfaces de monitoring et de contrôle, ainsi qu'aux informations du plan de connaissance à vue située.

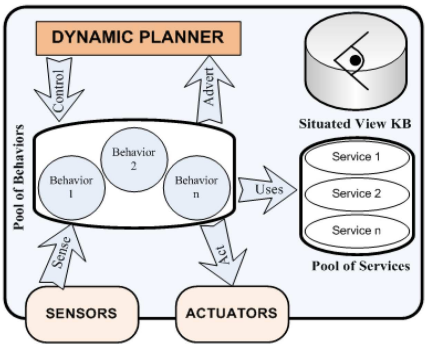


FIGURE 1.13: Architecture Ginkgo.

Cette architecture peut paraître moins complète que certaines des architectures présentées auparavant, mais la simplicité de ses concepts n'en fait pas moins une architecture efficace, dont

l'accessibilité pourrait être un atout pour convaincre les constructeurs. C'est pour ces raisons que nous avons choisi de nous baser sur cette plateforme pour l'implémentation des agents de décision GANA, nécessaire au prototype décrit à la Sec. 3.10.

1.3 Les architectures autonomes pour la gestion des pannes

1.3.1 L'autoréparation

Les pannes existent et elles existeront toujours, car elles trouvent leur origine dans des imperfections de conception ou dans des concours de circonstance qu'il sera toujours impossible de prévoir à l'avance. Malheureusement, le coût des pannes dans les réseaux de télécommunications, aussi bien en CAPEX qu'en OPEX [1] [MMJ08] et surtout des mécanismes de gestion de ces pannes, est très important. L'identification de la gestion des pannes par le modèle FCAPS [2] (voir Sec. 1.1.1.1) comme un élément essentiel de la gestion de réseau n'est pas anodin et reflète une vrai préoccupation des opérateurs.

Malheureusement, bien que les dispositifs de résilience aient été largement étudiés par le passé, les réseaux et les services fournis par ces réseaux sont toujours vulnérables aux pannes matérielles, aux *bugs* logiciels, aux attaques malveillantes, aux erreurs de configuration ou encore aux catastrophes naturelles.

De plus, les contraintes de temps que nécessite une gestion efficace des pannes ne sont pas compatible avec la gestion actuelle des incidents qui implique la présence d'experts humains pour analyser les centaines d'alarmes, prendre en compte l'étendue complexe de la situation et réagir rapidement après l'apparition d'un incident.

Ce problème est majeur pour les opérateurs qui font de la corrélation d'alarmes et de l'autoprotection leurs préoccupations de recherche principales [WL09]. L'application de l'autonomie aux processus de gestion des pannes pourrait permettre l'exploitation de la puissance et de la vitesse d'analyse et d'exécution des machines pour décupler l'efficacité de la gestion des pannes. Cette fonctionnalité est d'autant plus cruciale dans des conditions extrêmes, telles une catastrophe naturelle où les dommages sur l'infrastructure du réseau sont si importants qu'il est presque impossible pour des opérateurs humains de réagir rapidement et de manière efficace.

Un système autonome d'autoréparation se doit de réaliser la détection des pannes, leur localisation, le diagnostic complet ainsi que les actions nécessaires pour remédier aux problèmes, l'objectif avoué étant de maintenir l'ensemble des services assurés par le réseau, malgré un environnement dégradé, en maintenant la qualité de service promise aux clients.

Pour effectuer ces fonctionnalités dans l'environnement complexe des réseaux d'aujourd'hui, la conception d'une architecture distribuée, spécifiant les blocs fonctionnels de base, leurs relations et leurs modes de fonctionnement, est indispensable. De précédants travaux ont déjà formulé plusieurs propositions dont les spécificités sont détaillées dans la section suivante.

1.3.2 Des premières propositions spécifiques à l'autoréparation

1.3.2.1 Autonomic Network Architecture for Self-Healing

Lu et al. [LDRK10] proposent une architecture autonome pour l'autoréparation. La proposition s'attache aux problématiques de gestion des pannes dans un environnement de VoIP avec IMS [3]. Cette proposition diffère peu des architectures autonomes suscitées en suivant le modèle gestionnaire/ressource gérée avec deux niveaux hiérarchiques de gestionnaire :

- un gestionnaire autonome de ressources (*Autonomic Ressource Manager*) pour gérer les éléments de réseau ;
- un gestionnaire autonome de réseau (*Autonomic Network Manager*) avec une vue d'ensemble qui gère les différents gestionnaires autonomes de ressources.

1. Dépenses d'exploitation (*OPerational EXpenditure*)
2. *Fault, Configuration, Accounting, Performance, Security*
3. *IP Multimedia Subsystem*

Mais cette proposition met plutôt l'accent sur le modèle d'information utilisé pour l'autodiagnostic. Une représentation sous forme de graphe de causalité permet de trouver la source d'un problème en fonction des observations collectées. Ce graphe est composé de cinq types de nœud et de deux types d'arc différents comme illustré sur la Fig. 1.14

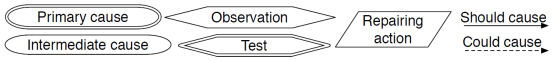


FIGURE 1.14: Éléments du graphe de causalité.

On retiendra de cette proposition l'utilisation d'un modèle pour un autodiagnostic efficace mais où l'architecture fonctionnelle nous semble peu adaptée aux nombreuses problématiques spécifiques à l'autoréparation.

1.3.2.2 CONMan

L'architecture CONMan [1] [BF07] est une architecture qui a pour but de masquer la complexité de gestion de chaque protocole en fournissant une interface de gestion générique. Pour cela, chaque protocole est décrit par un module d'abstraction tel que celui présenté par la Fig. 1.15 pour IP et qui détaille de façon générique ses capacités, ses fonctionnalités et ses dépendances.

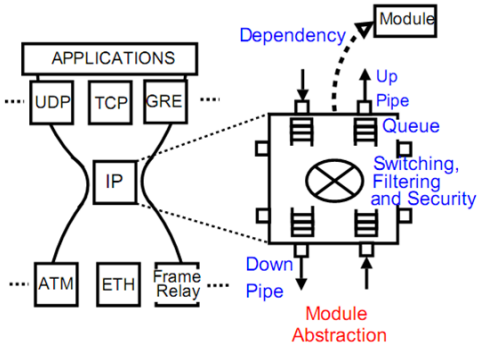


FIGURE 1.15: Module d'abstraction de l'architecture CONMan.

Chaque équipement possède ces modules d'abstractions qui sont utilisés par les gestionnaires de réseau (*Network Manager*) pour configurer et gérer le réseau en fonction des objectifs de haut niveau et en s'abstrayant des spécificités de chaque protocole.

Malgré aucune spécificité de l'architecture dédiée à la gestion des pannes, l'article « fault Management Using the CONMan Abstraction » [BF09] met en valeur les avantages de l'architecture CONMan, notamment grâce à son module d'abstraction et à son modèle de dépendance, dans la détection et la localisation des pannes.

1.3.2.3 UniFAFF

Le *framework* UniFAFF [2] [Cha09, CTS09] est le *framework* intégré à l'architecture ANA (voir Sec. 1.2.3) dédié à la gestion des pannes. Cette architecture définit d'abord plusieurs méta-modèles qui sont utilisés pour résoudre de manière autonome les problématiques de gestion des pannes, à savoir un méta-modèle de fiabilité, de causalité entre les erreurs, les fautes et les pannes, de causalité pour les pannes, fautes et erreur détectées, pour les alarmes et pour la connaissance fournie par les composants de monitoring. Les composants d'autoréparation de cette architecture sont situés dans chaque nœud de réseau, sous la forme de trois composants majeurs : le FDE [3], l'ANM [4] et le DMRepo (répertoire contenant les modèles de relation entre les entités) comme le

1. *Complexity Oblivious Network Management*
2. *Unified framework for implementing autonomic fault management and failure detection for self-managing networks*
3. *Failure Detection Engine*
4. *Autonomic Node Manager*

montre la Fig. 1.16 Le FDE est composé d'un contrôleur central et de divers autres composants

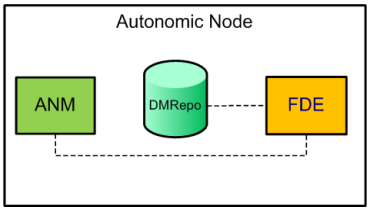


Figure 1.16: Architecture UniFAFF.

internes dont de nombreux registres et répertoires. Le contrôleur central a la charge d'initialiser et de contrôler les composants internes du FDE, d'invoquer certains mécanismes internes de détection, de diagnostic, de localisation, d'isolation et parfois même de suppression de panne. Il peut aussi, le cas échéant, interroger des répertoires pour obtenir certaines informations. Les autres blocs fonctionnels composant ce FDE sont :

- le *local incident registry* répertorie les erreurs, fautes et pannes du nœud en utilisant un méta-modèle pour les pannes/fautes/erreurs détectées. Il décide ensuite lesquelles doivent être disséminées vers les autres nœuds par l'intermédiaire de l'IDSP (voir ci-dessous) ;
- le *local and external alarm registry* stocke les alarmes du nœud ou du réseau en utilisant le méta-modèle d'information pour les alarmes ;
- l'*external incident registry* est alimenté par l'IDSP ou par des fonctions de monitoring avec les pannes/fautes/erreurs externes au nœud, c'est-à-dire, qui concerne une route, un lien, d'autres nœuds ou des services distants. Pour cela le registre utilise le méta-modèle pour les pannes/fautes/erreurs détectées ;
- le *specialized monitoring info-repository* contient la connaissance observée par les composants de monitoring, telle que des modèles de panne, des comportements anormaux, des situations de surcharge, etc ;
- le *fault diagnosis/localisation/isolation function including network troubleshooting and debugging function* est le composant exécutant les fonctions de diagnostic, d'isolation et de localisation de pannes grâce aux modèles de fiabilité et de causalité à sa disposition.
- l'*asserted incident registry* contient les pannes concernant des entités externes (par exemple, d'autres nœuds), afin, de manière collaborative, d'isoler ce nœud si celui-ci n'est pas en mesure de se réparer lui-même. Ce registre utilise donc le méta-modèle pour les pannes/fautes/erreurs détectées ;
- le *CMRepo-repository for fault-error-failure causality models/graphs* enregistre les modèles de causalité des blocs fonctionnels locaux. Ceux-ci peuvent être soit renseignés lors de sa conception, soit édités par la suite par un expert humain ;
- l'*incident information/knowledge dissemination service part* (IDSP) implémente les mécanismes de dissémination pour les informations locales concernant les alarmes et les pannes/erreurs/fautes détectées. Il sert aussi à avertir les autres nœuds de son rétablissement ou de son retour à la vie après une panne, ainsi qu'à la réception d'alarmes, de pannes, d'erreurs et de fautes externes au nœud ;

Le *dependability models repository* (DMRepo) stocke les modèles/graphes de fiabilité qui utilisent le modèle d'information de fiabilité. Il contient des informations de contexte ainsi que les informations de causalité du modèle de causalité. Ce répertoire est utilisé par les fonctions de diagnostic, de localisation et d'isolation mais peut aussi être utilisé par des composants externes au FDE. C'est la raison pour laquelle il se trouve à l'extérieur de celui-ci.

L'*autonomic node manager* (ANM) est une entité autonome en relation avec les autres blocs fonctionnels du nœud. Il est responsable de la gestion du nœud et de la collaboration avec les autres ANM lors d'opérations de gestion de plus grande ampleur. Il possède des fonctions autonomes impliquées dans les processus de diagnostic, de gestion des alarmes et de suppression des pannes, en permettant la supervision de l'état du nœud, la détection d'entités corrompues (mémoire, processus, etc.), le remplacement, la mise à jour ou le rechargement de certaines de ces entités ainsi que le redémarrage du nœud si nécessaire.

Ce *framework*, intégré à l'architecture ANA est la première architecture réellement spécifique à la gestion des pannes, définissant à la fois les blocs fonctionnels architecturaux et les modèles d'informations. Cependant, cette architecture semble se restreindre à une gestion réactive des pannes, alors que les méthodes proactives nous semblent être un pan important des possibilités offertes par l'autoréparation.

1.3.2.4 D^2R^2+DR

Le *framework* D^2R^2+DR [SHc+10, SHS+11] qui signifie *Defend, Detect, Remediate, Recover and, Diagnose and Refine* est un *framework* développé dans le cadre des projets ResiliNets et ResumeNet. La stratégie de ce *framework* s'articule autour d'une boucle d'opérations en temps réel présentée dans la Fig. 1.17. Cette boucle définie les composants nécessaires pour atteindre

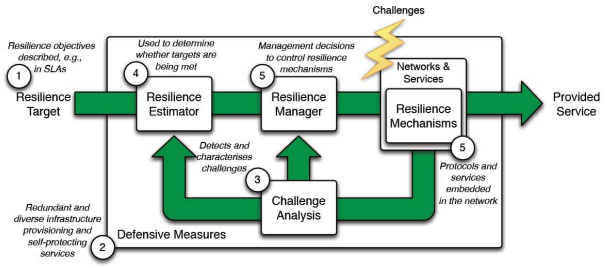


FIGURE 1.17: Boucle de contrôle D^2R^2+DR.

une résilience optimale dans les réseaux.

1. le premier élément de cette boucle est l'objectif de résilience (sous forme de métriques) que doit assurer le réseau en termes de service pour l'utilisateur final.
2. les mesures défensives sont des mesures de protection, configurées *a priori* et indépendantes de la boucle de contrôle, permettant d'aider à respecter l'objectif de résilience en cas d'incident (par exemple, la réservation de ressources redondantes).
3. l'analyse des challenges permet de détecter et de caractériser des incidents non pris en compte par les mesures préventives telles que des mauvaises configurations ou des attaques non prédites.
4. l'estimateur de résilience permet de déterminer, en fonction du contexte et de l'analyse de challenge, si les objectifs de résilience seront respectés.
5. le gestionnaire de résilience est responsable de l'activation des mécanismes de résilience appropriés, en fonction du rapport de l'estimateur de résilience ou de l'analyse de challenge.

Le bon fonctionnement de la boucle de contrôle de résilience requière l'utilisation et le partage de connaissances sur les incidents entre toutes les entités de cette boucle. Pour cela un système de *publish/subscribe* appelé DISco [1] permet l'échange d'informations nécessaires à la réalisation de cette boucle, comme illustré par la Fig. 1.18. Il est aussi à noter la présence d'une base de données séparée, permettant de stocker les politiques et les modèles de challenge, ainsi que la possibilité pour le gestionnaire de résilience de niveau réseau, d'interroger des composants externes tel que des *Path computational element* (PCE) pour l'aider dans sa tâche.

Ce *framework*, au même titre qu'UniFAFF semble être l'un des *frameworks* les plus aboutis concernant la gestion de pannes. Les diverses fonctions de la gestion des pannes sont prises en compte dans la conception de ces deux dernières architectures à l'exception des méthodes d'autoréparation proactives. En effet, dans un environnement où chaque seconde est comptée, la gestion des pannes pourrait être largement améliorée par l'utilisation de la rapidité d'analyse et d'exécution des machines pour anticiper les pannes et prendre des mesures proactives quelques secondes avant l'apparition de certains incidents.

1. Distributed Store for Challenges and their Outcome

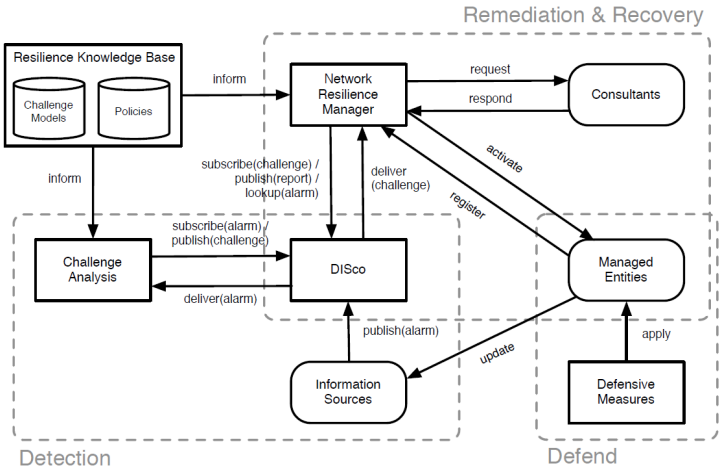


FIGURE 1.18: Architecture D^2R^2+DR.

1.3.3 Autoréparation proactive

L'autoréparation est bien souvent considérée comme la capacité d'une entité (un protocole, une application, un routeur, un réseau) à réagir d'une manière autonome aux incidents tels que les fautes, les erreurs ou les pannes, afin d'assurer la fourniture de services aux utilisateurs finaux, coûte que coûte. Cette fonctionnalité englobe les étapes de détection et de localisation de pannes, ainsi que les actions d'isolation, de masquage et de suppression des pannes.

Alors que la gestion traditionnelle des pannes s'appuie sur une analyse préliminaire des risques afin de dimensionner et de concevoir proactivement une infrastructure réseau tolérante aux pannes, la gestion autonome des pannes permet d'envisager un mode de fonctionnement proactif dynamique, qui anticiperait les pannes pour mieux les traiter. En effet la première étape d'analyse en temps réel des risques de panne et de prédiction des pannes en exploitant les alarmes et les comportements symptomatiques, devient envisageable grâce aux capacités de traitement des systèmes informatiques sans comparaison avec les opérateurs humains. Similairement, le système autonome a la capacité d'intervenir de manière extrêmement plus rapide qu'un opérateur humain, avant que la panne ne survienne, afin de minimiser les dégâts engendrés par la panne lors de son occurrence, voire même de l'éviter, en s'attaquant directement à la source du problème.

Dans cette thèse, nous nous concentrons sur l'introduction de la proactivité dans la gestion autonome des pannes, notamment par l'intermédiaire des trois dispositifs détaillés dans les Chap. 2 à 4. Cela nécessite la prise en compte, dès la conception de l'architecture de gestion autonome, d'entités permettant l'exécution de mécanismes proactifs, ce qui n'a pour l'instant jamais été considérée. Nous proposons donc, lors de la conception de la nouvelle architecture de gestion autonome GANA, la présence des blocs de base nécessaires à une gestion autonome des pannes efficace, et permettant l'usage de mécanismes d'autoréparation proactifs.

1.4 *Generic Autonomic Network Architecture* (GANA)

Cette section, présente la nouvelle architecture pour la gestion autonome des réseaux développée dans le cadre du projet EFIPSANS et dans laquelle nous avons conçu la gestion des pannes. Le modèle de référence pour la conception de réseaux autonomes, GANA [CPK+09], introduit une boucle de contrôle avec deux éléments de base : l'élément de décision ou*Decision Element* (DE) et l'entité gérée ou *Managed Entity* (ME). De plus, afin de gérer le réseau dans son ensemble, la prise en compte des différents niveaux de granularité constituant un réseau se fait par l'introduction de quatre boucles de contrôle hiérarchiques. Le protocole, la fonction, le nœud et le réseau constituent les différents niveaux hiérarchiques définis par l'architecture GANA.

1.4.1 Un modèle générique pour les réseaux autonomes

L'architecture GANA [CPT+10, CPK+09], comme la plupart des architectures autonomes, est structurée autour du concept d'une boucle de contrôle agissant dans le réseau et dans les équipements afin de rendre possible des fonctions avancées d'autogestion. Cette boucle est contrôlée par un élément de décision (DE) qui implémente la logique de décision nécessaire à la gestion d'un élément de réseau. L'envoi des commandes de gestion du DE par l'intermédiaire des interfaces de gestion de l'élément contrôlé (ME), ainsi que l'observation de l'état du ME, créent la susmentionnée boucle de contrôle par laquelle toutes les fonctions d'autogestion sont implémentées. Par conséquence, la conception d'une fonctionnalité d'autogestion dans GANA passe par l'assignation d'un DE à un élément de réseau, l'implémentation des interfaces de gestion et la conception de l'intelligence de gestion au sein du DE.

La Fig. 1.19 montre en détail le modèle générique d'un système autonome de réseau construit sur la base d'une boucle de contrôle. Ce modèle est une adaptation du modèle MAPE [Hor01] d'IBM, aux spécificités des réseaux de télécommunications. Le modèle gestionnaire/ressource gérée est ici reproduit par la paire DE/ME. Un ME peut être une ressource, c'est-à-dire un élément de réseau, mais aussi d'une manière plus générale, une tâche automatisée telle qu'un module protocolaire ou une fonctionnalité autonome complète.

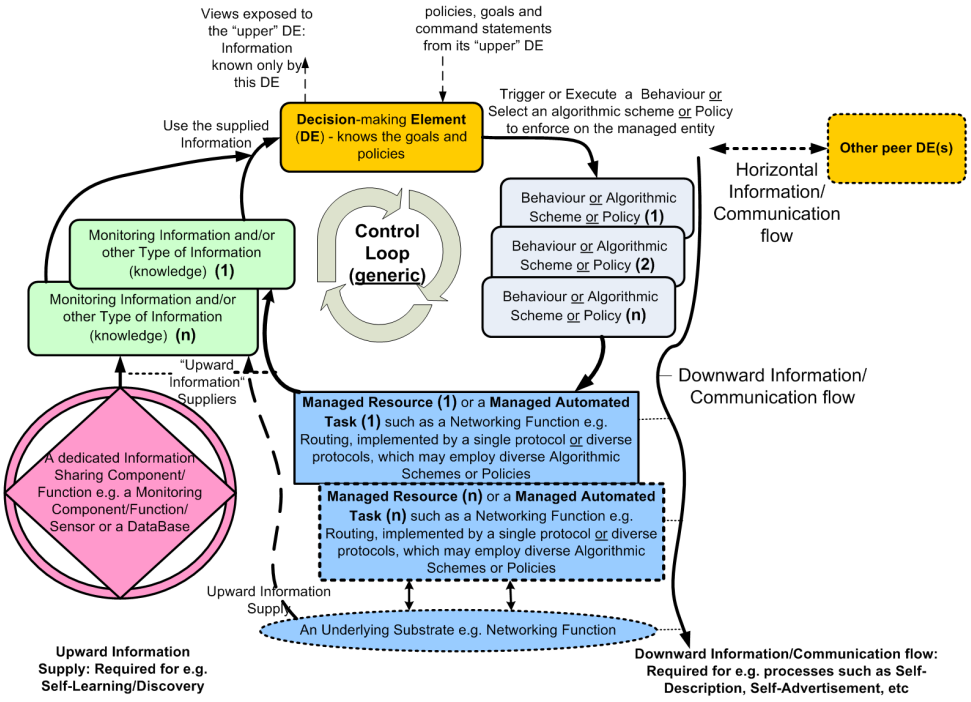


FIGURE 1.19: Modèle de référence de GANA.

La Fig. 1.19 illustre le mode de fonctionnement de la boucle de contrôle. Les informations sur l'état de l'entité gérée sont collectées par le *Decision Element* au travers de composants de *monitoring* ou de bases de connaissances, puis elles sont ensuite agrégées et analysées par le DE afin de construire une vue abstraite. Il convient ensuite au DE de déclencher et d'exécuter les actions appropriées au sein du ME afin de respecter les objectifs de gestion du réseau et les politiques venant des DE supérieures. Contrairement au modèle MAPE, les informations alimentant un DE peuvent provenir de multiples sources distribuées dans le réseau, augmentant de ce fait, la flexibilité dans la gestion et le partage de la connaissance nécessaire à l'exécution de mécanismes de contrôle autonomes. En contrepartie, cette flexibilité nécessite obligatoirement la standardisation des interfaces d'échange d'informations. C'est pourquoi les partenaires du projet EFIPSANS ont entrepris la création de l'AFI[1] [CCW+09] : un nouveau groupe de standardisation de l'ETSI[2] dédié aux réseaux autonomes.

1. *Autonomic network engineering for the self-managing Future Internet*
2. *European Telecommunications Standards Institute*

La Fig. 1.19 illustre aussi le fait que les actions prisent par un DE ne se restreignent pas au contrôle des ME mais peuvent aussi concerner des actions de communication avec d'autres entités, notamment d'autres DE. Les flux d'informations descendant et horizontal, indiqués sur la figure, ainsi que l'exposition des informations du DE aux DE supérieurs et la réception d'objectifs, de politiques et de commandes de la part de ces mêmes DE supérieurs, constituent ces nouveaux types de communications que peuvent effectuer un DE, en plus de ceux effectués avec son ME. L'intérêt de telles communications se justifie notamment pour les mécanismes de découverte autonomes nécessitant l'auto-description et l'auto-déclaration de chaque DE. Mais cela sous-entend qu'une hiérarchie est obligatoirement présente au sein du plan de DE et des boucles de contrôle, où des DE de niveau supérieur peuvent exécuter des actions de gestion sur des DE de niveau inférieur, impliquant de ce fait qu'une entité gérée (ME) d'une boucle de contrôle, peut aussi être un élément de décision d'une autre boucle de contrôle inférieure.

Dans le modèle GANA, une fonction autonome telle que l'autodescription, l'autoréparation, l'autoconfiguration pour ne citer qu'eux, est modélisée comme une action du DE essayant de réguler le comportement de son ME associé. Cette fonction autonome peut être spontanément démarrée par le DE ou déclenchée par la réception d'une nouvelle information d'une entité externe.

En plus de la définition des composants architecturaux et des interfaces de communication, GANA a défini des spécifications de comportement autonomes [Cha08], afin de régir les modes d'interaction et de collaboration des composants constituant une architecture autonome. Malgré un modèle volontairement simple, composé de seulement deux entités, il est nécessaire de permettre de multiples moyens de communication et de collaboration afin de prendre en compte l'étendu des situations complexes de gestion de réseau.

1.4.2 Modèle de relation entre blocs de base

L'architecture GANA propose l'utilisation de plusieurs niveaux hiérarchiques de DE, correspondant à différents niveaux d'abstraction. Ces niveaux hiérarchiques permettent des relations de type pair-à-pair, frère/sœur ou hiérarchique, entre les différents éléments de décision. Il existe donc des DE capables à la fois, de contrôler de manière autonome des ME, mais aussi, d'interagir avec d'autres DEs pour une gestion d'ensemble plus efficace.

La création de différents niveaux d'abstraction, permet, similairement au modèle en couche des réseaux, de s'abstraire de la complexité de conception d'une architecture autonome, en concevant des modules indépendants et en spécifiant leurs interactions. GANA définit quatre niveaux d'abstraction auxquels sont assignés les DE, ME et boucles de contrôle du modèle GANA.

- le niveau « protocole » est le niveau le plus bas qui concerne la logique d'un protocole réseau. Il s'agit là de décisions simples et pragmatiques nécessaires au fonctionnement du protocole. Les décisions de plus grande ampleur étant réservées à des niveaux supérieurs. On pourra citer à titre d'exemple : RSVP [1], ICMP [2] et Ethernet ;
- la « fonctionnalité » est le deuxième niveau, regroupant parfois plusieurs protocoles et mécanismes tels le routage, le transfert de paquets, la QoS, la gestion de la mobilité, etc ;
- le niveau « nœud » est le niveau regroupant les fonctionnalités et comportements d'un nœud du réseau tel qu'un routeur, un commutateur ou tout autre équipement du réseau ;
- enfin, le niveau le plus haut est le niveau « réseau » qui régit le comportement d'un domaine plus ou moins large du réseau dans son ensemble.

La Fig. 1.20 illustre l'instanciation de ces différents niveaux d'abstraction pour la gestion des pannes, où le DE du routage est considéré comme un ME par un DE de niveau supérieur. Le DE de niveau nœud a donc accès aux informations du DE inférieur ainsi qu'à d'autres informations auxquels ce DE n'a pas accès (autres DE, bases de connaissances partagées), afin d'influencer le comportement du DE de niveau fonction. Le DE de niveau nœud influence donc indirectement le comportement des ME de l'élément de décision du routage.

1. *Resource ReSerVation Protocol*
2. *Internet Control Message Protocol*

La collaboration *via* une relation de pair-à-pair est aussi représentée entre deux DE de même niveau (FM_DE [1] et R&S_DE [2]), ce qui permet des échanges d'information indispensables pour éviter les fluctuations résultant de prises de décision conflictuelles et non cohérentes.

Enfin, le dernier type de relation n'est pas représenté sur cette figure mais intervient lorsque deux DE sont contrôlés par un seul et même DE de niveau supérieur. Cette relation similaire à une relation entre frères est en quelque sorte un mélange entre les relations hiérarchiques et les relations pairs à pairs.

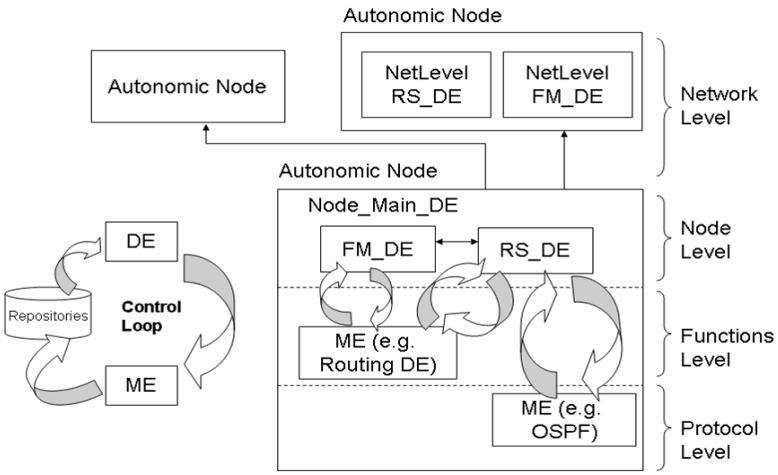


FIGURE 1.20: Niveaux d'abstraction dans GANA.

Pour mieux comprendre l'intérêt du concept de boucle de contrôle hiérarchique, son fonctionnement peut être illustré avec le cas du routage.

Au niveau le plus bas, le niveau protocole considère le protocole de routage. Celui-ci peut être un nouveau protocole de routage autonome, ou bien un protocole parmi ceux largement utilisés aujourd'hui tels qu'OSPF, IS-IS ou BGP [3]. Les fonctions d'autogestion déjà présentes au sein des protocoles (par exemple, le calcul des nouveaux plus courts chemins après une panne) sont alors modélisées au travers de DEs, de boucles de contrôle et de Mes, de niveau protocole. Il se peut que ces composants abstraits ne soient pas clairement visibles dans le fonctionnement de certains protocoles actuels, mais il est nécessaire d'avoir au moins une représentation virtuelle de ceux-ci, afin de pouvoir exposer les fonctions d'autogestion aux DE de niveau supérieur. Grâce à cette abstraction, il est alors possible d'incorporer les protocoles actuels dans le modèle GANA, afin qu'ils puissent bénéficier des apports en autonomie de cette architecture.

Viens ensuite le second niveau, la fonction abstraite, qui contient les éléments d'un équipement impliqués dans la tâche de routage. Par exemple, un équipement peut exécuter plusieurs protocoles de routage en parallèle, un protocole comme OSPF pour le routage intra-domaine et un protocole de type BGP pour le routage inter-domaine. C'est à la charge de l'élément de décision de niveau fonction de contrôler chaque protocole individuellement et d'orchestrer leurs interactions (par exemple, l'exportation de routes).

Au-dessus du niveau fonction, se trouve le niveau nœud, dont les DEs ont la tâche de coordonner les diverses opérations nécessaires au fonctionnement d'un équipement aussi complexe qu'un routeur.

Enfin, le niveau réseau qui représente le niveau hiérarchique le plus haut, possède de nombreux procédés pour contrôler le comportement d'un réseau dans son ensemble. Il est possible d'utiliser un DE centralisant les fonctions d'analyse de l'état du réseau et du contrôle de celui-ci, sur un équipement taillé en conséquence. Ce DE contrôle alors le réseau par l'intermédiaire des DE présents dans les trois autres niveaux. Mais il est bien plus opportun d'envisager un modèle distribué semblable à celui proposé dans l'architecture 4D [GHM+05] où le plan de gestion est

1. *Fault-Management Decision Element*
2. *Resilience and Survivability Decision Element*
3. *Border Gateway Protocol*

géographiquement séparé du plan de données. L'intelligence peut donc être implémentée au travers plusieurs DE constituant le niveau d'abstraction réseau, ou bien sans aucun DE de niveau réseau. Dans ce cas, les DE de chaque nœud forme, en coopérant entre eux, une sorte de niveau réseau virtuel en charge de gérer le réseau.

Les boucles de contrôle hiérarchiques sont une façon de représenter les relations et l'organisation des composants de l'architecture GANA, mais une autre perspective qui divise les fonctionnalités en quatre plans fonctionnels est disponible. Inspiré de l'architecture 4D [GHM+05], GANA est composée de quatre plans fonctionnels [Cha08] : le plan de découverte, le plan de dissémination le plan de données et le plan constitué de l'ensemble des DE : le plan de décision.

Après avoir passé en revue les caractéristiques du modèle générique GANA, il est intéressant de voir quels sont les DEs qui constituent cette architecture ainsi que leurs responsabilités respectives.

1.4.3 Instanciation des éléments de décision

Le projet EFIPSANS définit un certain nombre de DE [CPT+10]. Cette liste non exhaustive est susceptible d'être augmentée. Néanmoins, les DEs décrits ci-dessous constituent une base solide permettant le fonctionnement de manière autonome, de la majorité des fonctions fournies par les réseaux d'aujourd'hui.

1.4.3.1 *Network-Level Decision-Elements*

Tout d'abord des éléments de décisions ont un périmètre d'action étendu au management de tout le réseau. Les éléments de décision de niveau réseau sont des éléments de décision que l'on retrouve aussi au niveau nœud et fonction, et qui permettent de gérer les multiples DE du réseau de façon cohérente :

- le *Network-Level Routing-Management Decision-Element* a pour fonction la configuration et le partitionnement automatique du routage dans le réseau ;
- le *Network-Level Quality-Of-Service-Management Decision-Element* gère la QoS pour tout le réseau en collectant des métriques sur l'état du réseau, en gérant le contrôle d'accès et en optimisant la gestion et la réservation des ressources ;
- le *Network-Level Mobility-Management Decision-Element* adapte les politiques de gestion de la mobilité du réseau à l'activité et la capacité des terminaux et de leurs ressources de connexion ;
- le *Network-Level Resilience-&-Survivability Decision-Element* réalise les fonctions de masquage et de prédiction des pannes concernant plusieurs nœuds ;
- le *Network-Level Fault-Management Decision-Element* a pour objectif principal d'analyser et de réagir aux incidents portant sur un sous-ensemble du réseau, en réalisant des actions de diagnostic, de localisation, d'isolation et de suppression automatique des pannes ;
- le *Network-Level Security Decision-Element* teste la stabilité du réseau par des techniques de *fuzzing*, gère la confiance et le contrôle d'accès pour les DE ;
- enfin, le *Network-Level Service Management Decision-Element* a une vue globale de tous les services du réseau afin de contrôler leur bon fonctionnement ;

1.4.3.2 *Node-Level Decision-Elements*

La Fig. 1.21 donne une vision d'ensemble des DE présents au niveau du nœud et au niveau de la fonction :

- le *Node Main Decision-Element* est un DE d'importance un peu particulière puisqu'en plus de réaliser l'auto-découverte, il contient tous les autres DE de niveau nœud et leur fournit les interfaces et mécanismes de communication avec l'extérieur (*i.e.* les autres DEs) ;
- le *Node-Level Auto-Configuration Decision-Element* est en charge de configurer de manière autonome un nœud du réseau ;
- le *Node-Level Resilience-&-Survivability Decision-Element* réalise une première fonction de masquage des pannes afin d'assurer un fonctionnement optimal en présence de pannes, et

une deuxième fonction de prédiction de risque de pannes nécessaire pour l'autoréparation proactive ;
- le *Node-Level Fault-Management Decision-Element* a pour objectif principal d'analyser et de réagir aux incidents, en réalisant des actions de diagnostic, de localisation, d'isolation et de suppression automatique des pannes au sein du nœud ;
- le *Node-Level Security Decision-Element* réalise les fonctions d'autoprotection au sein d'un nœud ;

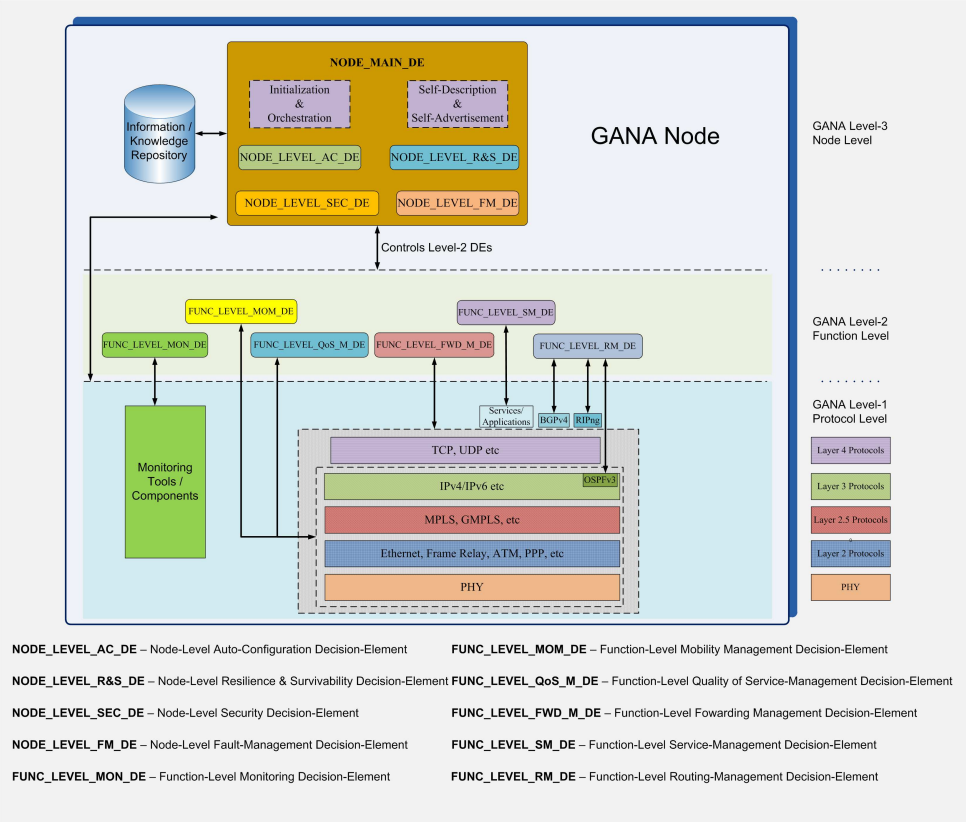


FIGURE 1.21: Structure d'un nœud GANA.

1.4.3.3 *Function-Level Decision-Elements*

Les DE présents dans le plan de fonction sont aussi représentés sur la Fig. 1.21. Ils sont affiliés à une et une seule fonctionnalité et à son fonctionnement au sein d'un nœud :
- le *Function-Level Routing-Management Decision-Element* règle en temps réel les paramètres du protocole de routage, permettant un fonctionnement optimal et une meilleure résilience aux pannes ;
- le *Function-Level Forwarding-Management Decision-Element* est en charge de régler les paramètres des protocoles du plan de données ;
- le *Function-Level Quality-Of-Service Management Decision-Element* fournit des fonctions autonomes de sélection et d'optimisation des paramètres de connectivité afin de proposer la meilleure QoE [1] possible ;
- le *Function-Level Mobility-Management Decision-Element* optimise la gestion de la mobilité d'un nœud ;
- le *Function-Level Service Management Decision-Element* gère l'exécution et le bon fonctionnement des composants d'un service réseau sur un nœud, tel que la VoIP par exemple ;
- le *Function-Level Monitoring Decision-Element* est un DE particulièrement critique dans cette architecture autonome puisqu'il fournit les fonctions d'observation de l'état d'un nœud et les dispositifs de partage de cette information à tous les DEs susceptibles d'être intéressés ;

1. Quality of Experience

1.4.3.4 *Protocols-Level Decision-Elements*

Les DE de niveau protocole sont nombreux car il y en a théoriquement un pour chaque protocole intégrant des mécanismes d'adaptation et d'autogestion. En voici quelques exemples illustratifs :

- le *Protocol-Level Neighbour-Discovery++ Decision-Element* gère le tout nouveau protocole de découverte de voisin proposé par le projet EFIPSANS ;
- le *Protocol-Level Base Station Resource and QoS Management in CDMA / WLAN* fournit des mécanismes de gestion du signal permettant une meilleure utilisation des ressources et de la QoS pour des stations de base utilisant les protocoles CDMA ou WLAN ;
- le *Protocol-Level Packet-Marking Decision-Element* permet la gestion d'un mécanisme qui marque les paquets pour observer la QoS ;
- le *Protocol-Level Queue-Management Decision-Element* permet aux DE en charge de la QoS de gérer le fonctionnement des files d'attente ;
- le *Protocol-Level Service-Aware Decision-Element* est utilisé pour détecter et identifier automatiquement les services utilisant le réseau ;
- le *Protocol-Level-Network-Based-Mobility-Management-Decision-Element* fournit les mécanismes nécessaires à la gestion des *proxys IPv6 mobile* permettant la mobilité dans IPv6 ;

L'instanciation de l'architecture GANA pour la gestion des réseaux actuels se doit de répondre aux principales fonctions de la gestion de réseau, et notamment aux fonctions définies par le modèle FCAPS (voir Sec. 1.1.1.1). Parmi ces fonctions, cette thèse s'intéresse plus particulièrement à la gestion des pannes. Dans la section suivante, le *framework* de gestion des pannes de GANA auquel nous avons contribué propose des blocs de base, permettant notamment aux fonctions d'autoréparation d'agir de manière proactive, en anticipation des pannes.

1.5 Architecture d'autoréparation dans GANA (UAFAReS)

L'architecture UAFAReS [WTVL13, TGV09, RNP+11, CPA+10, CPT+10, VNR+10, CPP+10, PKM+09, PKA+10, VCL+11, TDQ+09] est l'architecture en charge de la gestion des pannes au sein du modèle générique GANA. Partant du constat de l'évolution de la gestion de réseau vers l'autogestion, l'architecture UAFAReS propose une réponse ne nécessitant pas d'intervention humaine dans le processus de gestion des pannes, pour le déploiement de fonctionnalités d'autoréparation dans les réseaux. Quel que soit l'environnement réseau, l'architecture UAFAReS a pour objectif de fournir un nouveau *framework* prenant en compte les nombreux dispositifs de résilience dans leurs diversités afin d'implémenter une gestion des pannes autonome dans les réseaux. Le challenge dans la conception d'une architecture unifiée de gestion des pannes est de prendre en considération les nombreux mécanismes de résilience multicouches présents dans les équipements d'aujourd'hui. Afin de superviser et gérer au mieux les incidents, sans interférer dans les mécanismes d'autoréparation propres aux protocoles actuellement utilisés.

Nous avons donc cherché à identifier les principes de conception, ainsi que les composants principaux qu'un nœud de réseau autonome se devait de posséder, afin de faire face aux alarmes, erreurs, et autres pannes. Notre approche est basée sur l'analyse des facteurs qui influencent et nécessitent d'être adressés par une architecture visant à améliorer la fiabilité, la robustesse et la résilience de l'infrastructure réseau pour fournir de meilleures conditions aux services et aux applications. Il en résulte l'émergence de certaines fonctions indispensables que sont la détection et la dissémination des incidents, leur isolation, leur suppression ainsi que les fonctions de résilience aussi bien réactives que proactives. L'architecture doit donc proposer des composants et des interfaces adressant l'ensemble de ces fonctions de manière unifiée, collaborative et cohérente.

La fonction d'isolation des pannes, pour sa part, nécessite la prise en compte des relations de causalité (*i.e.* du modèle de propagation des fautes) entre les symptômes visibles (les pannes et alarmes) et la propagation des évènements découlant d'une panne.

Dans la même idée, la fonction de suppression des pannes requière la considération des dépendances entres les nœuds, les protocoles et les services afin d'avoir connaissance des entités

affectées par une action de suppression de panne (par exemple, le redémarrage d'un nœud).

Concernant les mécanismes de résilience proactifs, le besoin de composants architecturaux permettant d'estimer le risque de pannes des entités du réseau dans un futur proche nous apparait comme une évidence. Une évaluation des risques de panne permettrait de prendre des actions préventives afin de limiter, voire de supprimer les impacts d'une panne sur les services applicatifs (par exemple, le re-routage proactif).

Il convient aussi de s'assurer de la stabilité de la boucle de contrôle de gestion des pannes, afin de garantir que les actions entreprises par les composants de cette architecture ne sont pas contradictoires et contribuent à améliorer l'état du réseau. Pour cela, un composant permettant de synchroniser les actions des diverses entités autonomes est nécessaire. De plus, une fois les actions effectuées, le système de gestion des pannes se doit de s'assurer de la réussite de sa stratégie en vérifiant si tous les composants affectés par la panne sont de nouveau opérationnels.

Le *framework* architectural a besoin d'implémenter un mécanisme permettant aux applications d'exprimer leurs besoins (par exemple, les contraintes de temps) pour obtenir les informations vitales à propos des incidents et des pannes, pour permettre à la couche applicative de réagir efficacement lors d'un incident.

Enfin, il est totalement indispensable pour un système d'autoréparation de fournir des mécanismes et des interfaces de partage d'informations concernant les alarmes et les incidents, à toutes les entités qui pourraient en avoir besoin. En effet, les incidents sont une information précieuse qui doit se trouver dans tous plans de connaissances d'un réseau autonome.

De part ces première constatations, nous en avons déduit une liste d'exigences de conception que nous nous somme attachée de respecter dans la création de notre architecture d'autoréparation :

1. l'architecture doit définir les composants et interfaces pour les fonctionnalités de détection de pannes, de dissémination d'incidents, d'isolation de pannes, de réparation de pannes, et de résilience réactive et proactive ;
2. les entités autonomes du *framework* doivent avoir un accès complet aux blocs fonctionnels internes au nœud afin de pouvoir gérer leurs pannes ;
3. les mécanismes de résilience internes aux éléments de réseau et aux protocoles doivent être pris en compte ;
4. le système de gestion des pannes doit être en relation avec les services et les applications afin de leur fournir les informations relatives aux incidents selon leurs besoins ;
5. l'architecture doit fournir les outils pour partager les informations sur les incidents à tous les éléments d'un nœud et d'un réseau ;
6. les dépendances entre les protocoles, les nœuds et les services doivent être disponibles et considérées par les mécanismes de gestion des pannes ;
7. les relations de causalité entre la source d'un incident et la propagation des erreurs, des alarmes et d'autres symptômes observables doivent être considérées ;
8. les mécanismes proactifs de gestion des pannes nécessitent de savoir à l'avance si des éléments de réseau seront susceptibles de tomber en panne dans un avenir proche ;
9. les différents mécanismes agissant sur un incident ont besoin de s'assurer de la cohérence et la non contradiction de leurs actions ;
10. le résultat des actions du *framework* doit être vérifiable.

Basé sur le modèle GANA et la prise en compte des principes susmentionnés, nous avons défini les composants et dispositifs de collaborations pour unifier et automatiser la gestion des pannes en combinant la détection, l'isolation, la réparation, et le masquage des incidents. Pour cela, l'architecture fait la distinction entre les fonctions de *fault-management* et de résilience. Elle est donc principalement composée de deux éléments de décision le *Fault-Management Decision Element* (FM_DE) et le *Resilience and Survivability Decision Element* (R&S_DE), mais peut aussi s'appuyer sur d'autres DEs et notamment sur le *Routing Management Decision Element* (RM_DE) pour toutes les problématiques de routage tolérant aux pannes. Cette architecture est présente à la fois au niveau nœud comme le montre la Fig. 1.23 et au niveau réseau, afin de

gérer des situations complexes impliquant plusieurs nœuds et de fournir une stratégie unifiée et cohérente dans tout le réseau.

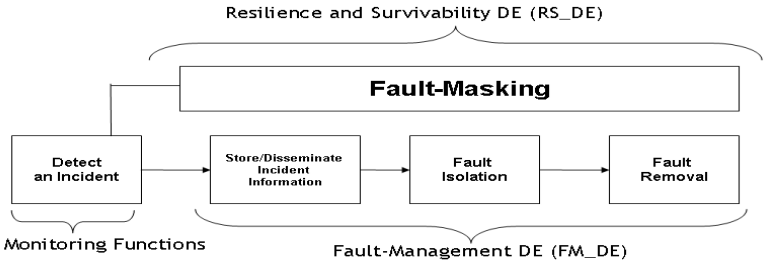


Figure 1.22: Responsabilités de gestion des pannes dans GANA.

Les responsabilités des deux DE sont bien différentes puisque le R&S_DE intervient dans un premier temps pour rapidement rétablir les services malgré la panne en employant des mécanismes de tolérance aux pannes. Alors que le FM_DE a la plus lourde tâche de réparer la panne et ses conséquences, en identifiant et supprimant la source du problème. Cela se traduit par le découpage illustré par la Fig. 1.22. L'implémentation des mécanismes d'isolation et de suppression des pannes est réservée au FM_DE tandis que les mécanismes proactifs d'estimation de l'état de santé du réseau et de masquage des pannes sont implémentés au sein du R&S_DE.

Les deux DE sont introduits au niveau nœud au sein du Node_Main_DE, comme l'illustre la Fig. 1.23, afin d'avoir accès à toutes les entités fonctionnelles (*i.e.* les DE et ME). Les deux DEs présents dans chaque nœud ont pour vocation de collaborer entre eux afin de construire une connaissance globale relative aux incidents, en utilisant des dispositifs de dissémination d'informations d'incidents détaillés par la suite. En utilisant leur vue locale, ils essayent individuellement de résoudre les problèmes relatifs aux pannes. Si cela n'est pas possible, interviennent les FM_DE ou R&S_DE de niveau réseau, uniques pour chaque réseau ou domaine, qui, à leur tour, avec des capacités et une vue étendue, essayent de résoudre le problème.

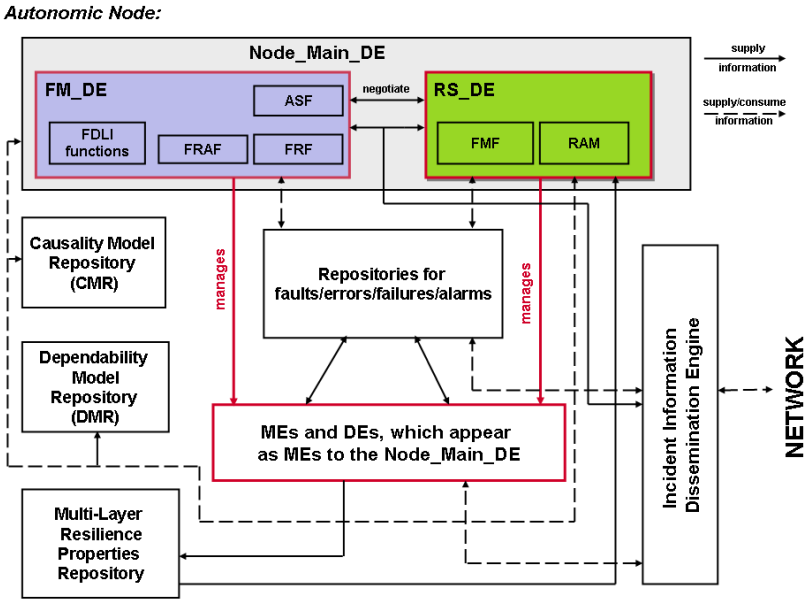


Figure 1.23: Instanciation de l'architecture UAFAReS dans un nœud.

La Fig. 1.23 montre le rôle central du FM_DE et du R&S_DE, ainsi que la forte collaboration entre ces deux DE.

Le FM_DE est composé de quatre modules le module FDLI [1] responsable des fonctions de

1. *Fault-Diagnosis/Localization/Isolation functions*

diagnostic, de localisation et d'isolation d'incident, le module FRF [1] en charge de la suppression/réparation des pannes, le module ASF [2] pour la synchronisation des différents mécanismes du *framework* et le module FRAF [3] qui permet de vérifier si la réparation d'une panne est bien effective.

Le R&S_DE est quand à lui composé de deux modules. Le premier module FMF [4] se charge des fonctions de masquage des pannes. Le second module est le module RAM [5] en charge d'évaluer le risque de pannes.

Les collaborations et communications entre ces DE sont nombreuses que ce soit au travers de dépôts externes tels que les *fault/error/failure/alarm repositories* ou de composants internes tels que le RAM qui permet à la fois, d'augmenter les connaissances de la base de données contenant les relations de causalité (CMR [6]) et d'aider dans la fonction vérifiant le succès ou non d'une action de réparation (*Fault-Removal*).

L'échange et le partage d'information est la raison majeure de l'introduction de cinq composants externe aux DEs, qui assistent les deux DEs dans leurs rôles :

- l'IDE [7] joue le rôle de *proxy* en permettant le partage et la dissémination des incidents, des alarmes et des erreurs aux DE et ME du nœud et du réseau qui en auraient besoin. Il permet notamment de renseigner les applications et les services sur les incidents du moment, afin que ceux-ci s'adaptent en conséquence. Lorsqu'il reçoit un incident, il a la charge de l'envoyer aux dépôts concernés et de le transmettre vers l'extérieur, aux autres DE du réseau ;
- le dépôt CMR est le dépôt stockant le modèle de causalité. Ce modèle représentant les différentes relations entre les symptômes (pannes et alarmes), les évènements intermédiaires non visibles résultant de la propagation de la panne, et la faute source correspondante. Ces relations sont conservées d'une certaine façon pour être analysées de manière quasi-immédiate, puisqu'elles sont parcourues à chaque action d'isolation du module FDLI ;
- le DMR [8] fournit les relations de dépendance entre les nœuds, les protocoles et les services nécessaires au processus de suppression des pannes. Elles permettent de connaitre l'étendue des entités pouvant être affectées par la tentative de suppression de la panne ;
- le dépôt pour les alarmes et incidents (*Fault/Errors/Alarm/Failure Repositories*) est le dépôt recensant tous les incidents venant du nœud ou du réseau. Cette information est structurée dans différents dépôts suivant le type d'évènement (alarmes, incidents, etc.) et leur localisation (interne ou externe au nœud). Il possède des dispositifs d'abonnement afin de permettre aux entités intéressées de recevoir les derniers incidents et alarmes au travers de l'IDE ;
- enfin, le dernier dépôt contient les propriétés des mécanismes de tolérance aux pannes et de résilience des protocoles actuels afin de permettre aux deux DE de composer avec ces mécanismes dans leurs actions de gestion des pannes.

Les fonctions des deux DE du *framework* UAFAReS sont développées dans les section qui suivent, ainsi que sur le RM_DE qui, en lien avec le R&S_DE, permet l'introduction de la résilience proactive au sein des fonctions de routage. Alors que les mécanismes implémentés par le FM_DE ne seront pas développés dans cette thèse, les R&S_DE et RM_DE constituent les blocs de base permettant la réalisation des mécanismes proactifs développés dans les chapitres suivants.

1. *Fault-Removal Functions*
2. *Action Synchronization Functions*
3. *Fault-Removal Assessment Functions*
4. *Fault Masking Function*
5. *Risk Assessment Module*
6. *Causality Model Repository*
7. *Incident Information Dissemination Engine*
8. *Dependability Model Repository*

1.5.1 Élément de décision de gestion des pannes

Le processus de gestion des pannes autonome commence lorsqu'un symptôme ou une anomalie est détectée par un composant de *monitoring* au sein d'un nœud (Mon_DE[1]). À partir de la détection d'un symptôme particulier, les composants de *monitoring* rapportent leurs observations à un ensemble de dépôts stockant les alarmes, les erreurs, les pannes, et les fautes, qui eux-mêmes en notifient le *Fault-Management Decision Element* (FM_DE).

C'est alors que le FM_DE entre en jeu par l'intermédiaire de ses quatre modules : le module *Fault-Diagnosis/Localization/Isolation functions* (FDLI), le module *Fault-Removal Functions* (FRF), le module *Action Synchronization Functions* (ASF) et le module *Fault-Removal Assessment Functions* (FRAF).

Une fois un incident rapporté au FM_DE, le FDLI commence par diagnostiquer la cause initiale du problème. En effet, le FDLI est le composant responsable du diagnostic et de l'isolation d'une panne. Cela correspond à la phase d'analyse de la boucle de contrôle. La fonction FDLI collecte toutes les alarmes et les incidents afin de déclencher, le cas échéant, le processus d'isolation. Ce processus lourd, n'est pas déclenché pour chaque alarme, mais seulement lors d'incidents biens particuliers nécessitant l'implication du FM_DE. Le nombre d'alarmes et d'erreurs, ainsi que leur sévérité sont les critères prépondérants dans le processus de décision du déclenchement des mécanismes d'isolation d'un incident. Ce processus d'isolation essaye de déterminer la source à l'origine de l'incident en utilisant un modèle de causalité basé sur les chaînes de Markov [TCC10]. Ce modèle est entreposé dans le dépôt CMR, et permet, *via* la définition des relations entre différentes alarmes et pannes, de connaitre la façon dont se propage un incident.

Après avoir identifié le problème initial, le résultat du processus d'isolation envoie son résultat au module de réparation (FRF) qui va tenter de rétablir l'élément à la source de l'incident. Les techniques utilisées peuvent se restreindre au rechargement d'un processus, à la reconfiguration d'un protocole ou d'une interface mais aussi aller jusqu'au redémarrage complet d'un équipement. Le module FRF a donc une forte influence sur les ME qu'il contrôle, ce qui le rend particulièrement critique vis-à-vis de la stabilité globale du système. C'est ici qu'intervient le module ASF de synchronisation qui permet d'éviter des actions contradictoires avec les autres composants de la gestion des incidents. Ce composant permettant de respecter les objectifs haut niveau est systématiquement interrogé par toutes tentatives d'action de la part du FM_DE et du R&S_DE, afin de détecter les éventuelles actions contradictoires pouvant aboutir sur des oscillations qui seraient néfastes pour la stabilité du réseau.

De plus, une fois l'action de suppression de la panne autorisée, le FM_DE doit s'assurer de connaitre toutes les entités impactées par le processus de suppression. Le FRF peut alors utiliser le dépôt DMR contenant les relations de dépendance entre les nœuds, les protocoles et les services afin de découvrir précisément les entités qu'il va perturber et de s'assurer qu'il ne va pas empirer la situation.

Enfin, la dernière étape de la boucle de contrôle du FM_DE vérifie que les actions entreprises ont correctement résolu l'incident. Le module FRAF va sonder et tester les différentes fonctions et services afin de vérifier qu'ils ont été restaurés avec succès et qu'ils sont de nouveau pleinement opérationnels. Il peut aussi interroger le module de risque, afin de s'assurer que la situation est totalement réparée et dépourvue de vices cachés qui pourraient relancer une série de nouvelles pannes. Si le résultat est positif, le module FRAF va alors marquer l'incident comme résolu ou, dans le cas contraire, déclencher l'intervention du FM_DE de la couche réseau ou encore demander l'intervention d'un opérateur humain.

1.5.2 Élément de décision de tolérance aux pannes

L'autre élément de décision au cœur de l'architecture UAFAReS est le *Resilience and Survivability Decision Element* (R&S_DE). Cet élément de décision a pour principal objectif le masquage des pannes, c'est-à-dire éviter au maximum qu'une panne ne perturbe les services des utilisateurs finaux. Ce rôle est implémenté par le module FMF qui intervient immédiatement

1. *Monitoring Decision Element*

après la détection d'un incident. Sa connexion directe avec le FM_DE lui permet de bénéficier des informations à jour sur les incidents détectés et résolus et son accès aux diverses bases de donnée partagées, à l'ensemble des informations relatives aux incidents. À ses côtés, le module RAM est au cœur de cette thèse puisqu'il a pour objectif d'analyser l'état du réseau, de ses équipements, et de prévenir les autres entités (notamment le module FMF), lorsque le risque de panne devient important sur un élément de réseau. Ce module couplé au module FMF, permet la réalisation de mécanismes de résilience proactifs, intervenant quelques secondes avant l'occurrence d'une panne en plus des mécanismes réactifs traditionnellement utilisés. C'est en effet le module FMF qui implémente la fonction d'adaptation du niveau de protection du protocole GMPLS (*Adaptive Level of Recovery* (ALR)) développée au Chap. 4.

Il est aussi intéressant de noter que les fonctions du FMF ne résolvent pas un incident à la manière du FM_DE mais propose des solutions temporaires pour permettre aux services de continuer de fonctionner correctement avec une qualité de service acceptable et ce malgré un environnement dégradé. Néanmoins et, similairement au FM_DE, le module FMF doit, préalablement à toute action, consulter le module de synchronisation (ASF) afin de s'assurer que son action ne compromettra pas l'intégrité d'un élément de réseau. Un autre aspect du FMF est la considération des dispositifs de résilience autonomes déjà présents au sein des protocoles et équipements actuels. Des exemples de tels dispositifs sont les mécanismes de reconvergence d'OSPF lorsqu'une panne apparaît [Moy98], ou les mécanismes de protection et de restauration des protocoles de transport tels que GMPLS [MP06]. Pour une meilleure efficacité, le module FMF se doit de composer avec de tels mécanismes au travers d'un dépôt spécifiant les propriétés et comportements de ces mécanismes de résilience.

1.5.3 Le module de détection du risque en bref

Au sein du R&S_DE se trouve le module RAM, qui a pour objectif de permettre au système de gestion des pannes d'évoluer d'une gestion purement réactive à un mode de fonctionnement dynamiquement proactif, et qui essaye de prédire les pannes afin de les anticiper et de mieux les gérer. Il existe quantité de mécanismes préconfigurés avant les pannes, mais ceux-ci sont conçus et dimensionnés lors du déploiement du service en se basant sur des statistiques de disponibilité des équipements sur le long terme. Le *Risk Assessment Module* (RAM) permet un raisonnement similaire mais de manière dynamique où, pendant l'exécution d'un service, il est possible d'anticiper une panne quelques secondes à l'avance, et de mettre en place dynamiquement les dispositifs de résilience optimaux pour cette panne.

En effet, ce composant compose avec la nature fluctuante de la probabilité de panne, car le risque de panne d'un équipement réseau ou d'un sous-ensemble du réseau, évolue avec le temps en fonction de multiples facteurs tels que l'état du trafic, l'âge des équipements ou même des conditions externes tels que l'état du système d'alimentation électrique. Les approches actuelles considèrent le risque de pannes d'une manière statique, en ignorant l'aspect dynamique de l'exposition aux risques de l'infrastructure réseau. La conception des systèmes de résilience actuels se base sur des statistiques de disponibilité moyenne (tels que le MTBF [1]) pour le dimensionnement initial, qui est ensuite conservé durant toute la vie du service. De plus, pour des raisons de simplicité et afin résister à la plupart des situations extrêmes, les systèmes de résilience sont surdimensionnés, ce qui est très coûteux pour l'opérateur. Un exemple parlant est l'utilisation de liens redondants pour protéger le trafic contre une panne éventuelle, où le deuxième lien est inutilisé dans plus de 99% des cas et double les coûts en CAPEX de l'opérateur. Le RAM est là pour résoudre ce déficit en donnant aux systèmes de résilience un accès à une information de risque de panne en temps réel, permettant de mettre en place, avant l'occurrence de la panne, des mécanismes minimisant sa sévérité. Pour cela le module RAM observe de multiples sondes et métriques (par exemple, le nombre de paquets rejetés par une interface) permettant la détection de comportements symptomatiques annonciateurs de panne. Quelques exemples tels que la température du processeur ou de la carte mère, l'état et la vitesse de rotation des ventilateurs, la stabilité de la tension d'alimentation, l'état du disque dur vis-à-vis du nombre de secteurs

1. *Mean Time Between Failure*

défectueux et le différentiel entre le temps d'utilisation de l'équipement et le temps moyen entre deux pannes de l'équipement illustrent ces paramètres. Des paramètres logiciels peuvent aussi être observés tels la charge du processeur ou de la mémoire, les fuites de mémoire, les fichiers corrompus, les erreurs systèmes ou la présence de virus. Enfin, des activités externes peuvent être surveillées telles les attaque de pirates informatiques ou tout simplement la présence d'activités de maintenance informatique dans le voisinage. Ces indicateurs annonciateurs de panne sont bien souvent déjà présents au milieu de centaines d'informations rapportées au superviseur de réseau. Mais la configuration et l'activation d'un nouveau mécanisme de résilience avant l'occurrence de la panne n'étant pas possible pour un opérateur humain, ces informations ne sont utilisées qu'à des fins de réparation. Il y a donc un potentiel important à intégrer ces informations dans un processus exécuté dans les équipements de réseau, et qui permettrait de déclencher rapidement des mécanismes de résilience, en anticipation des pannes. Plus en détail sur les spécificités et la faisabilité de ce module d'évaluation des risques sont donnés dans la Sec. 1.6.

La fonction du RAM est donc d'envoyer le résultat de son évaluation à toutes les entités intéressées dont fait partie le module FMF afin de déclencher les mécanismes de résilience proactifs, mais aussi le RM_DE [1] qui implémente deux mécanismes d'autoréparation proactifs.

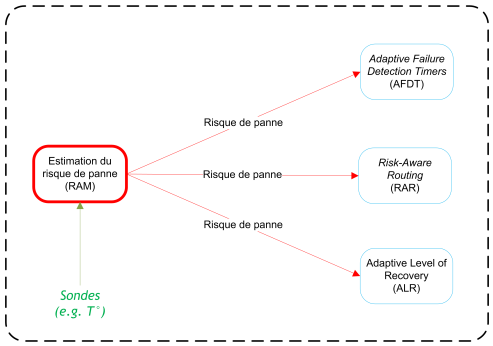


Figure 1.24: Utilisation du RAM.

En effet, comme l'illustre la Fig. 1.24, le module d'évaluation des risques est le déclencheur des trois dispositifs développés dans les trois prochains chapitres. Le mécanisme relatif à l'adaptation du niveau de protection du protocole GMPLS (*Adaptive Level of Recovery* (ALR)) est implémenté au sein du module FMF du R&S_DE (Cf Chap. 4), mais les deux autres mécanismes relatifs au routage sont implémentés au sein du RM_DE. Ces deux mécanismes concernent la modification proactive du schéma de routage pour éviter les routeurs risqués (*Risk-Aware Routing* (RAR)), ainsi que l'adaptation de la fréquence d'envoi des messages de détection de pannes au risque de pannes courant (*Adaptive Failure Detection Timers* (AFDT)).

1.5.4 Élément de décision de routage

Le *Routing Management Decision Element* (RM_DE) est l'élément de décision responsable de la fonction de routage. Ce DE de niveau fonction est en charge du bon fonctionnement des protocoles tels qu'OSPF, IS-IS et BGP. Son but est tout d'abord, de correctement configurer le protocole de routage en décidant de manière autonome du choix de découpage des aires de routage et des différents paramètres de configuration. En plus de cette fonction d'autoconfiguration, le RM_DE possède deux modules d'autoréparation proactive dédiés au routage. Le premier module est le RAR [2] qui est en charge de modifier dynamiquement les métriques de routage pour obliger le trafic à éviter les routeurs dont le RAM a annoncé qu'ils vont probablement tomber en panne prochainement (Cf Chap. 3). Le deuxième module est l'AFDT [3] qui a pour fonction d'adapter la fréquence d'envoi des messages de détection de pannes, au risque de pannes (Cf Chap. 2). Pour cela, ces deux modules sont fortement liés au RAM, qui permet comme illustré par la Fig. 1.25, de déclencher ou non l'activation du mécanisme implémenté en leur sein.

1. *Routing Management Decision Element*
2. *Risk-Aware Routing*
3. *Adaptive Failure Detection Timers*

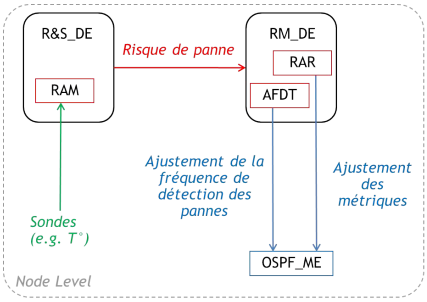


FIGURE 1.25: Utilisation du RAM par le RM_DE.

La section suivant s'attache à éclaircir de manière plus précise, les caractéristiques du RAM, qui joue ici un rôle indispensable majeur sur lequel s'appuient les trois dispositifs d'autoréparation qui seront développés dans cette thèse.

1.6 Le module de détection du risque en détail

La logique des mécanismes développés dans cette thèse s'appuie sur la capacité d'un système informatique à donner une information de risque de panne en temps réel, c'est-à-dire à prédire les pannes quelques secondes à l'avance avec une certaine probabilité. Cette idée, qui peux paraître farfelue dans un premier temps, n'est pas nouvelle, et commence à être développée par plusieurs industriels en commençant par la société Sentient Business Systems [Cas00], mais aussi par des acteurs beaucoup plus importants tels que Nokia-Siemens au travers de l'initiative *Proactive Care* [Net10] et Oracle (anciennement Sun Microsystems) par l'implémentation de fonctionnalité de *Predictive Self-Healing* [Ora08] au sein du système d'exploitation Solaris. Dans cette même idée, nous pensons que l'introduction d'un module d'évaluation du risque de pannes (RAM), au sein de chaque élément de réseau, responsable de surveiller les indicateurs d'état de santé du système, d'en déduire une information de risque en temps réel et de la diffuser aux modules de résilience proactive, n'est pas utopique. En effet, malgré le caractère soudain d'une panne, de nombreux éléments nous laissent penser qu'il serait possible de prédire certaines pannes à l'avance.

Il est important de commencer par l'étude des caractéristiques et propriétés des incidents qui perturbent les réseaux d'aujourd'hui, et notamment les réseaux de cœur IP auxquels nous nous intéressons dans cette thèse.

1.6.1 Caractéristiques des pannes

Bien que les équipements soient toujours plus fiables, les pannes restent une préoccupation majeure des opérateurs. Malheureusement, ceux-ci ne communiquent que très peu sur leur expérience dans ce domaine. En effet, la quantité et les caractéristiques des pannes sont des données critiques que les opérateurs se gardent bien de diffuser. Néanmoins, la communauté de recherche peut s'appuyer sur quelques travaux de référence caractérisant les pannes dans les réseaux de cœur IP tels que l'article de C. Labovitz et al. [LAJ99], les études du réseau de cœur IP de l'opérateur Sprint [ICM+02, MIB+08] et la récente analyse d'un réseau de l'opérateur Orange et d'Internet2 [MTFM10, MTM07, MTFM09, Med10]. De plus, les similarités d'un réseau de télécommunications avec un réseau informatique d'entreprise permet de tirer des enseignements des rapports tels ceux du *Yankee group* [Sur02], qui sont beaucoup plus nombreux dans ce domaine. Bien que l'étude de ces différents travaux montre une évolution des caractéristiques des pannes au cours du temps et des spécificités propres à chaque réseau, il est néanmoins possible d'en tirer de nombreux enseignements laissant penser que bon nombre des incidents pourraient être détectés quelques secondes à l'avance, si l'on s'en donne les moyens.

Les termes de faute, panne, erreur étant bien souvent utilisés à tort et à travers, il est nécessaire de définir brièvement la signification de ces termes :

– une *panne* est un évènement qui impacte le service fourni à l'utilisateur final. En d'autres

termes, lorsqu'un évènement implique un dysfonctionnement dans le service délivré, et que celui-ci est visible par l'utilisateur final, il s'agit d'une panne. L'utilisateur final pouvant aussi bien être un humain, qu'un système informatique (notamment dans les systèmes *Business to Business*) ;
- une *erreur* est une situation où le système dévie de son comportement normal. C'est une situation anormale, mais qui peut rester invisible et ne pas obligatoirement impacter le service. Bien que les erreurs soient traquées par les systèmes de gestion de pannes, elles ne sont pas toujours détectées, c'est pourquoi, on différencie généralement les erreurs détectées et les erreurs non détectées ;
- la *faute* est l'origine de l'erreur : sa cause. Inversement, l'erreur est la conséquence visible d'une faute. Une faute est un évènement qui peut être silencieux pendant une certaine durée de temps, puis ensuite déclencher un comportement anormal, c'est-à-dire une erreur ;
- le terme *incident* permet de nommer un évènement de type panne, erreur, ou faute, en regroupant ces trois notions sous un même terme ;
- enfin, un *symptôme* est une manifestation visible des effets de bord que génère une erreur.

Les pannes survenant dans les réseaux de télécommunications peuvent être classées en trois grandes catégories : les fautes liées aux problèmes matériels (panne d'un composant), aux problèmes logiciels (*bug*, virus, etc.) et aux problèmes environnementaux (erreur humaine, problème électrique, panne du système de climatisation, catastrophe naturelle).

Les pannes les plus difficiles à prédire sont de nature environnementale et selon l'étude récente de Medem et al.[MTM07], ne représentent plus que 7,3% des incidents. Les catastrophes naturelles dont les orages représentent 34%, les inondations 17% et les tremblements de terre 9% [Cas00] sont, bien entendu, les plus difficiles à prédire. Mais elles constituent une part infime des incidents que l'on peut sans conteste ignorer. Néanmoins, avec la disponibilité des alertes météorologiques sur l'Internet, il serait tout à fait possible de relier ces prédictions à un système de prédiction de pannes, afin d'incorporer ces données, dans le modèle de risque. Les coupures de fibre optiques ont souvent été mises sur le devant de la scène par la presse, puisqu'elles peuvent avoir des répercussions importantes, allant jusqu'à la coupure de l'Internet pour plusieurs pays. Néanmoins, ces cas correspondent bien souvent à des cas bien particuliers, car ne disposant pas de liens redondants. Ils ne sont pas si courant qu'on veut bien le penser, car ils ne représentent plus que 4% des incidents du réseau VPN d'orange [MTFM10] alors qu'ils dépassaient les 15% dans l'étude de C. Labovitz et al. [LAJ99]. De plus, bien que les ravages des pelleteuses, les dégradations des rongeurs, ou les dommages sous-marins soient des événements très soudains, des études envisagent sérieusement la prédiction de ces coupures *via* l'analyse des états de polarisation de la fibre [PLR10]. Les systèmes d'alimentation électrique, ainsi que les systèmes de climatisation sont critiques pour le bon fonctionnement de l'infrastructure de réseau. Leurs dysfonctionnements peuvent entrainer l'arrêt, voire la dégradation des équipements de réseau. Heureusement, ces systèmes intègrent de plus en plus des procédés redondants permettant de limiter les répercussions sur les équipements réseau. Les incidents de ce type représentent un faible pourcentage (3% selon [MTFM10]) et peuvent être anticipés grâce à des sondes de température, l'observation des tensions d'alimentation et de l'état des batteries de secours. Pour terminer avec les incidents environnementaux, les erreurs humaines telles les mauvaises configurations représentent 4,5% des incidents selon [MTM07]. La gestion humaine est source de nombreuses erreurs d'où cet engouement pour les techniques d'autogestion. Ces erreurs intervenant presqu'intégralement lors d'opération de maintenance, la prise en compte des lieux et horaires des opérations de maintenance dans le modèle de risque est une option à envisager sérieusement.

Ensuite les problèmes matériels comprennent entre 15 et 20% des incidents selon les études [MTFM10, LAJ99]. Les pannes matérielles résultant de la défaillance d'un ou plusieurs composants déclenchent des signaux avant-coureurs qui sont visibles grâce aux fonctions de détection d'anomalie déjà présentes dans les équipements. Par exemple, les travaux de Y. Li [LGLS07] utilisent ces signaux pour la prédiction de pannes dans les systèmes distribués. Les pannes matérielles des composants ne sont pas concernées dans les mêmes proportions puisque selon une étude réalisée sur les systèmes informatiques [Cas00], à elles seules, les pannes touchant les disques durs et les alimentations comptent pour 83%. Plus précisément, les composants les plus touchés sont les processeurs et les contrôleurs (4%), la mémoire (5%), les systèmes de refroidis-

sement internes à l'équipement tels les ventilateurs, les systèmes hydraulique ou à effet Peltier (8%), l'alimentation (28%) et les disques durs (55%).

Pour finir, les dysfonctionnements logiciels, au regard des études citées précédemment, ont fortement progressé (de 9% dans l'étude datant de 1999 à environ 30% en 2010). Ce phénomène peut s'expliquer par l'augmentation du nombre de fonctionnalités implémentées dans les équipements et la complexité grandissante pour intégrer le tout. Les dysfonctionnements d'ordre logiciel incluent les *bugs* mais aussi les virus et attaques de pirates informatiques. Néanmoins ces deux derniers ne réunissaient que 1,5% des incidents dans l'étude de C. Labovitz et al. [LAJ99]. C'est dans ce domaine que la prédiction des pannes a le plus fort potentiel, car de nombreux symptômes peuvent indiquer l'apparition d'un *bug* critique, tels qu'une utilisation anormale du processeur, une augmentation continue de l'utilisation de la mémoire et bien d'autres paramètres qui ont été identifiés dans de précédentes études [GMT08]. De même, les virus et le piratage informatique suivent des modèles de comportement bien identifiés par les antivirus et les systèmes de détection d'intrusion (IDS [1]).

L'analyse de la littérature et les observations des équipes de support opérationnel soulignent qu'une grande partie des pannes sont répétitives et précédées par des signes avant-coureurs bien connus. Il est donc important de concevoir des systèmes de prédiction de pannes analysant les symptômes précédant une panne, évaluant le risque de panne et informant les systèmes d'autoréparation du réseau.

1.6.2 La prédiction de pannes

Les méthodes classiques de la théorie de la fiabilité utilisent des taux moyens de panne et des prédictions sur le long terme qui ne considèrent pas l'état courant du réseau et qui sont uniquement exploités dans la phase de dimensionnement d'un réseau ou d'un service. Malheureusement, ces techniques ne reflètent pas la dynamique du réseau et de ses pannes. Il convient donc de mettre en place des composants de prédiction à court terme, permettant dans un délai très court, de mieux préparer l'arrivé d'une panne. La prédiction à court terme, s'appuie sur l'observation de certains paramètres du système, en temps réel, afin d'évaluer la probabilité d'occurrence d'une panne dans un futur très proche (de l'ordre de quelques secondes à quelques minutes). Elle est la première étape d'un processus permettant de dépasser les limites de la gestion traditionnelle des pannes pour aller vers une gestion proactive.

1.6.2.1 La prédiction de pannes temps réel au sein de la gestion proactive

L'objectif de la prédiction de pannes en temps réel est de prédire à l'avance, l'apparition des pannes dans un futur proche, et ce, pendant le fonctionnement du système, en analysant son état de santé courant. Elle est le déclencheur indispensable au processus de gestion proactive des pannes, représentées sur la Fig. 1.26.

FIGURE 1.26: Les étapes de la gestion proactive des pannes.

Ce processus débute avec une surveillance continuelle de plusieurs paramètres clés utilisés pour la prédiction. Quelques secondes avant une panne, lorsque suffisamment d'indices laissent penser qu'une panne va apparaître, une prédiction de panne est effectuée. La panne éventuelle allant apparaître dans un court délai, il est important d'envoyer cette information de risque de panne à tous les mécanismes de gestion proactive. Les délais étant très court, il est bien souvent impossible de supprimer la faute initiale, mais il est possible de préparer le réseau, afin de limiter l'impact de la panne sur les services. Une fois la prédiction reçue, les mécanismes de résilience proactive, analysent le problème, décident s'il est opportun d'intervenir et choisissent

1. *Intrusion Detection System*

la stratégie à adapter en fonction de la panne. Pour les cas favorables, le mécanisme proactif exécute, le plus rapidement possible, une action préventive, minimisant les dégâts, dans le cas où la panne prédite surviendrait. Lors de l'apparition de la panne, les dispositifs de résilience sur-mesure permettent de limiter la perturbation des services, mais une réparation plus en profondeur est nécessaire pour revenir à l'état stable initial. La Fig. 1.27 illustre les différents

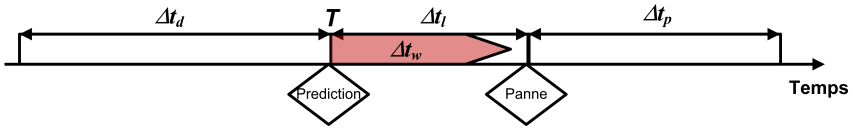


Figure 1.27: Délais considérés lors d'une prédiction.

délais caractérisant la tâche de prédiction. T étant le temps présent, la prédiction se base sur un historique de paramètres observés sur une fenêtre de temps Δt_d. Lors d'une prédiction, afin de rendre possible l'intervention de mécanismes proactifs, un délai minimum Δt_l avant l'occurrence de la panne est définit. Celui-ci dépend du délai de réaction du mécanisme proactif Δt_w puisqu'il doit lui être supérieur, sans quoi la prédiction serait inutile. Enfin, une prédiction ne reste pas valide *ad vitam aeternam*, mais est bornée dans le temps avec une durée de validité Δt_p au bout de laquelle, si la panne n'est pas apparue, la prédiction est déclarée comme fausse.

1.6.2.2 Les métriques de performance de la prédiction de pannes

La notion de performance est importante dans la prédiction de pannes, car elle aura un impact direct sur l'utilité des mécanismes proactifs engagés par la suite. La qualité de la prédiction de pannes s'évalue au travers de deux axes distincts : la capacité à détecter le plus grand nombre de pannes et la capacité à minimiser le nombre de mauvaises prédictions. Un *vrai positif* (TP) est une panne prédite à l'avance, un *vrai négatif* (TN) signifie que le risque de panne est nul et qu'aucune panne n'apparaît, un *faux positif* (FP) se réfère à une panne prédite qui ne survient pas et un *faux négatif* (FN) est le cas d'une panne non précédée par une prédiction.

À partir de ces quatre cas de figure, deux métriques évaluent la qualité de la prédiction de pannes. La première évalue la capacité à détecter les pannes. Le *Recall* est le ratio de pannes prédites par rapport au nombre total de pannes comme le définit l'Eq. (1.1).

$$Recall = \frac{TP}{TP + FN} \tag{1.1}$$

La deuxième métrique apprécie l'aptitude à minimiser le nombre de fausses prédictions. La *Precision* (Cf Eq. Équation 1.2) est le rapport entre les pannes correctement prédites et le nombre total de prédictions (vrai ou fausse).

$$Precision = \frac{TP}{TP + FP} \tag{1.2}$$

Ces deux métriques représentant deux aspects distincts dans la qualité d'un mécanisme de prédiction, il est très difficile de comparer deux mécanismes ayant chacun l'avantage sur une des métriques. Pour palier en partie ce problème, le *F-measure* (Cf Eq. Équation 1.3) combine ces deux métriques en une seule.

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{1.3}$$

Néanmoins, le but de cette thèse n'étant pas d'évaluer tel ou tel mécanisme de prédiction, mais de prendre en compte la performance de la prédiction de pannes dans l'évaluation des méthodes proactives proposées dans les chapitres suivants, seuls le *Recall* et la *Precision* sont utilisés dans la suite de cette thèse.

1.6.2.3 Les techniques de prédiction de pannes

Afin de posséder des mécanismes proactifs les plus efficaces possibles, il est important de s'appuyer sur des techniques de prédiction de pannes performantes. Il n'y a malheureusement pas de recette miracle, car même si chaque technique permet d'attendre des performances tout à fait honorables, une approche combinant plusieurs techniques, (une sorte de méta-prédiction [GLL+07]) parait la plus à même de permettre une prédiction efficace de tous les types de panne qu'un réseau peut subir. La section fait l'inventaire des différentes techniques de prédiction de pannes qui pourraient être implémentées dans les réseaux. Elle sont classées en quatre grandes catégories suivant les données qu'elles analysent, à savoir : les pannes, les symptômes, les rapports d'erreurs ou les erreurs non détectées.

Suivi de panne

La première catégorie de techniques utilise les occurrences précédentes de panne pour en déduire les futures pannes.

- **Estimation de la distribution de probabilité**
  Cette méthode cherche à estimer la distribution de probabilité du délai jusqu'à la prochaine panne, en fonction de l'historique des pannes au temps t. Elle utilise les bases de la théorie de la fiabilité, mais appliquées en temps réel, pendant le fonctionnement du réseau, avec l'utilisation de prédiction de pannes bayésienne [Cse90] ou de méthodes non paramétriques [PCF02].
- **Cooccurrence**
  La cooccurrence exploite le fait que certaines pannes apparaissent de manière proche dans le temps et dans l'espace ceci afin d'inférer les futures pannes. Une telle propriété a été exploitée dans [LZJ+06] et [FX07] pour prédire les pannes dans des systèmes distribués.

Observation de symptômes

L'observation des symptômes est une des catégories le plus utilisées pour la prédiction de pannes. Elle s'appuie sur l'observation périodique de certaines variables du système (par exemple, la quantité de mémoire libre, la température du processeur) car celles-ci reflètent les effets de bord de la faute qui, dans quelques instants, va entrainer une panne. En effet, les symptômes d'un problème dans le système sont bien souvent visibles avant que cela n'aboutisse à une vraie panne. Quatre approches principales se démarquent :

- **Approximation de fonction**
  Les méthodes d'approximation de fonction essayent de prédire l'évolution de certains paramètres (par exemple, la mémoire disponible) afin de prédire le moment où celui-ci attendra une valeur synonyme de panne (par exemple, une température de processeur supérieure à cent degrés). Les techniques sont multiples, utilisant des modèles stochastiques [VT99, LVT02], la régression [AS07] et même l'apprentissage [HTM07, FX07, AG05].
- **Classificateurs**
  Les classificateurs sont des mécanismes qui décident, en analysant les paramètres du système, si celui-ci est dans un état sujet aux pannes ou pas. Pour cela, il doit d'abord disposer d'un historique de pannes avec les valeurs des paramètres observés correspondants, afin de définir quels états sont sujets aux pannes et lesquels ne le sont pas. La prédiction consiste ensuite à savoir auquel des deux ensembles (sujet aux pannes ou non), l'état actuel appartient. Deux types de classificateur sont le plus communément utilisés : les classificateurs bayésiens [HE01, PSBDG98] et les classificateurs flous [TA03, BAV03, MHKD03, BFB+05].
- **Modèles de système**
  La modélisation de systèmes permet de représenter tous les états autorisés par le système qui ne sont pas sujets aux pannes. A ces états correspondent des intervalles de valeurs pour les paramètres observés. Si les valeurs courantes des paramètres observés diffèrent trop de ceux appartenant aux modèles non sujets aux pannes, une panne est prédite. Les modèles d'instances [EKA03, HMKDE02, CKF+02, BFB+05], de regroupement d'instances [MHKD03], stochastiques [WGR98, CAK+04], de graphe [KF05] et de la théorie du contrôle [GBB02, CGM02] sont les approches qui ont été étudiées dans ce domaine.

– **Analyse de séries temporelles**
Cette méthode traite une séquence de paramètres observés comme une série temporelle. Des échantillonnages successifs de la valeur des paramètres observés sont analysés, soit pour décider si une panne est imminente, soit pour prédire l'intervalle de temps avant l'occurrence de la panne. Les techniques utilisées sont la régression [GVMVT98, CHH+01, CWT+05], l'analyse componentielle [CSC02, SCC+03] et la prédiction de séries temporelles [HZS99, VAH+02, SOR+03, CWT+05].

Rapport d'erreurs détectées

La catégorie des méthodes basées sur l'analyse des rapports d'erreurs, c'est-à-dire des erreurs détectées par le système, contrairement à la surveillance de symptômes, n'effectue pas de *monitoring* de manière active. Elle est entièrement rythmée par l'apparition aléatoire des rapports d'erreurs. Nous avons donc un fonctionnement complètement différent de celui basé sur les symptômes, qui agit en réaction à la détection d'une ou plusieurs erreurs. Les méthodes d'analyse des rapports d'erreurs comportent cinq grandes catégories :

– **Approche basée sur des règles**
Ce type de prédiction [HKM+96, Wei99, VM02] s'appuie sur un ensemble de règles et de conditions qui devront être remplies pour déclencher une prédiction de panne. La difficulté est donc de construire ce système de règles en se basant sur des données historiques contenant les pannes et les erreurs.
– **Cooccurrence**
Similairement au mécanisme utilisé dans la catégorie de suivi des pannes, la technique utilise la proximité (dans le temps et l'espace), mais cette fois-ci entre les erreurs détectées par le système, comme présenté dans les articles [LC03, NA85, LC98, LLR04].
– **Reconnaissance de modèles**
Les séquences d'erreurs peuvent former des modèles d'erreurs. L'objectif de la reconnaissance de modèles est d'identifier les modèles qui indiquent une panne imminente et de vérifier lors de l'apparition de nouvelles erreurs, leurs similarités avec les modèles sujets aux pannes. Les travaux de Vilalta et al.[VAH+02] et de Salfner et al. [SSM06, SM07] utilisent cette technique pour la prédiction de pannes.
– **Tests statistiques**
Cette catégorie regroupe toutes les techniques utilisant des tests statistiques pour analyser les rapports d'erreurs comme les travaux de Levy et Chillarege [LC03].
– *Classificateur*
Le principe de la classification, utilisée notamment par Domeniconi et al. [DPVM02], est similaire à la classification des symptômes, sauf que celle-ci s'effectue sur les erreurs détectées.

Audit d'erreurs non détectées

La dernière grande catégorie de techniques de prédiction est beaucoup plus lourde puisqu'elle implique une recherche active des erreurs. Cela permet notamment de ne pas attendre les rapports d'erreurs et donc de gagner un temps précieux dans le processus de gestion proactive. On peut citer à titre d'exemple, la recherche d'incohérence dans la structure du système de fichiers par la vérification complète de la consistance du système de fichier. Cette méthode est en quelque sorte un mélange entre les deux catégories précédentes, puisqu'elle travaille sur des erreurs comme la catégorie utilisant les erreurs détectées, mais utilise une recherche active, un peu à la manière des dispositifs actifs de surveillance de symptômes.

Bien que l'ensemble de ces techniques contribuent, à leur échelle, à fournir une prédiction de pannes efficace, les techniques basées sur les journaux d'erreurs (*log*) et sur le *monitoring* de symptômes, sont les plus populaires et celles qui semblent les plus adaptées aux contraintes des réseaux, comme la section suivante l'illustre au travers des paramètres utilisés pour la prédiction.

1.6.2.4 Les paramètres annonciateurs de panne

Quelles que soient les techniques utilisées pour la prédiction, et même si certaines techniques donnent de meilleurs résultats que d'autres, elles sont toutes fortement dépendantes des données qu'elles analysent. En effet, ces techniques s'appuient grandement sur une étape de *monitoring* (même si celle-ci ne fait qu'observer les alarmes rapportées au gestionnaire de réseau), dont les éléments surveillés ont une très grande influence sur le résultat de la prédiction. La section a donc pour but de faire un tour d'horizon des paramètres à prendre en compte dans un système de prédiction de pannes pour les réseaux. L'état de santé du réseau peut être analysé *via* une multitude de paramètres dont la plupart sont déjà disponibles au milieu des centaines d'informations et d'alarmes déjà rapportées au gestionnaire de réseau. Ils sont donc, pour la plupart, bien connus des équipes opérationnelles. De plus, des exemples concrets montrent que l'ensemble des paramètres pertinents dépendent du type de panne (voir Sec. 1.6.1) visé dans la prédiction.

Un des paramètres les plus parlant pour illustrer une dégradation lente est la température. C'est une des raisons pour laquelle nous avons choisi ce paramètre pour notre démonstrateur [KAB+11] (voir annexe D) implémentant le mécanisme proposé au Chap. 3. La température peut être mesurée sur différents éléments d'un équipement, notamment le processeur, les contrôleurs, la mémoire, la carte mère, les cartes de ligne, les cartes d'extension telles que les cartes de contrôle ou de DPI [1], les disques durs, le châssis, les alimentations, etc. Une augmentation ou une variation anormale de température peut être le résultat d'une panne dans le système de climatisation, d'une anomalie dans les éléments de refroidissement internes à l'équipement, de filtres encrassés, qui vont, dans un avenir proche, obliger l'équipement à s'éteindre par mesure de sécurité, ou même pire, créer une surchauffe dommageable pour les composants. Dans ce cas, des seuils doivent être mis en place afin de détecter les températures anormalement élevées, ou des augmentations de température trop rapides, symptomatiques d'un problème sérieux. La température est une valeur facilement récupérable dans les systèmes d'exploitation d'aujourd'hui *via* le standard ACPI [2] ou les cartes d'extension de type BMC [3] [LGLS07]. Par exemple le système d'exploitation Linux donne accès à toutes les sondes ACPI *via* l'application lm_sensor [Lin11a]. De plus, il est même possible de brancher des sondes directement sur le réseau, ce qui permet de connaitre la température ambiante et ainsi de renforcer le modèle avec une information supplémentaire relative à la température.

Le système de refroidissement interne à l'équipement, s'il vient à se détériorer, est une cause de panne à très court terme. En effet, il est composé d'éléments mécaniques qui peuvent s'user, se casser et donc ne plus assurer correctement leur fonction de refroidissement. S'en suit une surchauffe des composants avec les mêmes conséquences que celles décrites au paragraphe précédent. De la même manière que la température, l'ACPI et les cartes BMC permettent d'avoir accès au statut des ventilateurs (marche/arrêt), ainsi qu'à leur vitesse de rotation afin de détecter tout fonctionnement dégradé.

L'alimentation est un élément critique dont la dégradation plus ou moins lente d'un composant peut avoir des répercussions désastreuses sur tout l'équipement. Les valeurs définissant l'alimentation en électricité des éléments de l'équipement, telles la tension ou l'intensité doivent être surveillées afin de détecter toute valeur anormale ou instabilité qui sont symptomatiques d'une dégradation de l'alimentation. Pour cela, il est possible d'utiliser les sondes présentes au niveau de la carte mère ou des cartes additionnelles au travers des protocoles tels l'ACPI, ou bien d'interroger directement les modules UPS [4] *via* les nombreux protocoles de communication existants (par exemple, le protocole Megatec). Ces modules UPS qui ont pour fonction de garantir une alimentation continue malgré les pannes du système électrique, utilisent parfois des batteries de secours. Ces batteries ayant une durée de vie limitée, il est aussi nécessaire d'interroger le module UPS afin de connaitre la puissance disponible et ainsi évaluer le risque de coupure électrique.

Bien que la fiabilité des disques durs des routeurs et autres éléments de réseau soit bien

1. *Deep packet Inspection*
2. *Advanced Configuration and Power Interface*
3. *Baseboard Management Card*
4. *Uninterruptible Power Supply*

meilleure que les disques durs de nos ordinateurs de bureau, ce composant est très sensible aux pannes. En effet, ses éléments mécaniques sont fragiles, et ils peuvent entrainer des dysfonctionnements à cause de secteurs défectueux. Les unités de stockage à base de mémoire flash, et notamment les SSD [1] sont aussi concernées, puisque leur cellules s'usent en fonction du nombre d'accès, et engendrent des blocs défectueux. Heureusement, les disques de stockage d'aujourd'hui implémentent le système S.M.A.R.T. [2] qui fournit des sondes indiquant l'état de l'unité de stockage, la température, la vitesse de rotation des disques, les erreurs et bien d'autres paramètres sur l'état de santé du disque.

Les paramètres de réseau et de qualité du signal tels le nombre de paquets rejetés, la taille des files d'attente, les taux d'erreurs par bit (BER [3]) doivent être observés, principalement pour détecter des problèmes sur les liens ou au niveau des nœuds voisins. En effet, cela permet de détecter la dégradation lente d'un laser, d'un récepteur ou de bien d'autres composants impliqués dans le processus de transmission du signal. De plus, comme cité précédemment, les états de polarisation de la fibre peuvent permettre de prévoir des coupures de fibre lors de travaux, des dégradations de rongeurs ou d'autres évènements, *a priori*, très soudains. [PLR10].

La charge totale des unités de traitement (processeur, processeur de réseau, unité de calcul, contrôleur de disque, contrôleur d'entrée/sortie, etc.) est un paramètre à prendre en compte, puisqu'un *bug*, un piratage informatique ou un processus fou peut être la source d'une utilisation importante d'une de ces unités. L'unité est alors incapable d'assurer le traitement des autres tâches, ce qui perturbe automatiquement le fonctionnement global de l'équipement. Certains routeurs étant configurés pour redémarrer si la charge dépasse un certain seuil, la perturbation n'est généralement pas trop longue, mais une interruption de service est quand même visible.

Bien que la mémoire ne soit pas une unité de traitement, la charge d'utilisation de la mémoire est aussi un indicateur pour la prédiction de pannes. Une consommation importante de l'espace mémoire, surtout pendant de longues périodes, est un comportement anormal qui peut conduire à un ralentissement du temps d'exécution des processus demandant de l'espace mémoire. De plus, la durée d'utilisation et l'âge des composants mémoire ont été identifiés comme des indicateurs d'erreurs mémoire [SPW09]. Il en va de même pour la quantité d'erreurs corrigeables puisqu'il a été observé que la plupart des erreurs non corrigeables par les modules ECC [4] étaient précédées d'erreurs corrigeables [SPW09].

Après avoir vu que la charge globale des composants est un paramètre important à surveiller, une analyse plus fine du taux d'utilisation du processeur, de l'utilisation mémoire, du remplissage des files d'attente des entrées/sorties, pour chaque application, processus et fil d'exécution permet de repérer si un processus particulier demande anormalement l'accès à une ressource. Car la présence de processus utilisant trop l'unité de traitement tend à empêcher les autres processus d'accéder aux processeurs, ne pouvant, au final, plus assurer leurs tâches. De la même manière, des opérations d'entrée/sortie intensives, comme la lecture et l'écriture d'un grand nombre d'octets sur le disque local, affectent tous les autres processus qui ont besoin de communiquer avec les entrées/sorties, et peuvent refléter la présence d'un *bug* logiciel. Enfin, l'empreinte mémoire d'un processus peut être surveillée pour détecter les processus ayant une mauvaise utilisation de la mémoire, avec des problèmes tels que des fuites mémoire.

Comme vu précédemment, bon nombre de pannes proviennent de problèmes logiciels. Beaucoup d'entre eux ont des modèles de comportement identifiables. Par exemple, une boucle infinie causée par une mauvaise mise à jour de l'itérateur est identifiable par un usage intensif du processeur. Le *bug* de fuite mémoire peut aboutir à une erreur de type *out of memory*, ou exécuter un mauvais bout de code. Ce problème est dû à un processus réservant de l'espace mémoire sans jamais le libérer. On peut donc observer ce phénomène en surveillant l'utilisation de la mémoire et le ratio de mémoire virtuelle. De la même façon, le système de fichiers peut être exposé à des fuites de descripteur de fichier, et des corruptions de données avec des conséquences similaires. Un dernier exemple de dégradation logiciel (*software aging*) est le problème d'inter-blocage de verrous où un processus est bloqué par d'autres processus demandant l'accès aux mêmes res-

1. *Solid State Disk*
2. *Self-Monitoring, Analysis and Reporting Technology*
3. *Bit Error Rate*
4. *Error Correction Coding*

sources telles que les verrous de fichiers, les jetons (*token*), les tubes (*pipe*), les sémaphores et tous les dispositifs de communication inter-processus (IPC [1]). D'autres dysfonctionnements logiciels existent et la plupart sont identifiables par des modèles de comportement symptomatiques.

Avec les bonnes performances des mécanismes de prédiction basés sur les erreurs, les exceptions et les alarmes du système d'exploitation doivent sans conteste être incorporées dans le modèle de prédiction, car elles sont bien souvent symptomatiques d'un mal bien plus grand et annonciateur d'une panne imminente : les erreurs lors du chargement de processus, les erreurs de lecture de *socket*, les erreurs de lecture de flux, les erreurs d'alignement, les erreurs d'adressage de donnée, les erreurs d'adressage d'instructions, les erreurs de préchargement du cache, les erreurs de lecture ou d'écriture de données, les erreurs de files d'attente pleines, le *crash* d'un module ou d'un processus et bien d'autres encore.

Les incidents relatifs à la sécurité tels les virus et le piratage informatique sont aujourd'hui relativement bien identifiés à l'avance par des composants tiers tels les antivirus et les systèmes d'IDS. Il est tout à l'intérêt du RAM de communiquer régulièrement avec ces composants externes afin d'être averti de toutes tentatives hostiles qui pourraient entrainer un incident. De même pour le NMS ou l'OSS [2] qui permettent de connaitre les lieux et horaires des activités de mise à jour et de maintenance qui sont susceptibles de générer des erreurs humaines. À plus long terme, une communication avec les systèmes d'alerte météo permettrait aussi d'anticiper des incidents liés aux catastrophes naturelles majeures.

Enfin, et même si la prédiction de pannes que l'on cherche à concevoir est une prédiction à court terme, précédant la panne de quelques secondes, les statistiques long terme sur la disponibilité des composants peuvent permettre de consolider le modèle de risque. D'après [MIB+08], les éléments de réseaux n'ont pas tous la même probabilité de panne, car certains éléments (défectueux) tombent plus souvent en panne que d'autres. De plus, des données constructeur sont disponibles pour modéliser la probabilité de panne d'un équipement, comprenant entre autres le MTTF [3] et le MTBF. Il est donc utile de surveiller la durée de fonctionnement de chaque équipement et de la corréler avec ce modèle, afin de rajouter (encore) un autre paramètre dans le modèle de prédiction.

1.6.2.5 Quelques mots sur les performances

Bien que les performances des méthodes de prédiction actuelles ne se résument pas à la valeur des deux métriques que sont le *Recall* et la *Precision*, mais à une multitude de résultats suivant les conditions, il est important de situer les performances des études présentes dans la littérature.

Tout d'abord, il est utile de préciser que les paramètres de configuration du mécanisme de prédiction, à savoir Δt_d, Δt_l et Δt_p ont un impact direct sur les performances.

En effet, Δt_d a un effet sur les performances puisque, plus la taille de l'échantillon des paramètres observés est grand, plus les performances augmentent. Néanmoins, ce constat est bien moins important que sur les deux autres paramètres et n'est vrai que jusqu'à une certaine limite. Ainsi la prédiction de pannes utilisée au sein des dispositifs proposés dans cette thèse suppose un Δt_d assez grand pour permettre des performances maximales.

Δt_p permet lui aussi d'augmenter les performances du mécanisme de prédiction, car plus cette période est longue, plus une panne à de chance d'apparaître. Néanmoins, une trop grande valeur réduirait automatiquement l'intérêt des mécanismes proactifs qui resteraient actifs sur une trop grande période et perdraient de leur nature dynamique. Pour donner un ordre d'idée, les travaux de Salfner [SSM06, SM07] dont les méthodes de prédiction sont parmi les plus performantes, utilisent un Δt_p d'une minute. Afin de prendre une certaine marge et d'être compatible avec un grand nombre de techniques proposées dans la littérature, cette thèse considère des valeurs situées entre cinq minutes et dix heures, avec une valeur cible d'une heure.

Enfin, plus Δt_l est court, plus les performances sont élevées, car la prédiction concerne un futur plus proche et donc moins indécis. Les mécanismes proactifs de cette thèse permettant

1. Inter-Process Communication
2. *Operating Suport System*
3. Mean Time To Failure

une intervention de l'ordre de la seconde, un Δt_l d'environ cinq secondes permets d'exploiter les dispositifs de prédiction de la littérature dans leur cas le plus favorable et avec leur performance maximale.

Afin de situer le lecteur, avec la configuration précédemment citée, les mécanismes proposés dans la littérature proposent des performances de plus de 90% pour le *Recall* et de 80% pour la *Precision* [SSM06]. En conséquence, cette thèse considère dans un *Recall* et une *Precision* de 20% comme faibles, de 50% comme moyens, et de 80% comme très performants.

1.6.3 Considérations relatives au module de risque

Le module d'évaluation des risques, le RAM, est présent au sein de chaque équipement et au niveau réseau dans le R&S_DE. En charge d'évaluer le risque de panne grâce aux techniques de prédiction de pannes listées précédemment, il se doit aussi de disséminer l'information de risque à toutes les entités intéressées par une telle information (notamment le module FMF du R&S_DE et les modules RAR et AFDT du RM_DE).

Les possibilités d'exploitation d'une information de risque sont nombreuses c'est pourquoi il est utile de définir plusieurs niveaux de risque dans l'information envoyée par le RAM, afin de ne pas empêcher la conception de dispositifs proactifs sensibles à la probabilité de concrétisation de la prédiction. On pourra notamment prendre en exemple le scénario envisagé dans [RNP+11] où un risque de panne moyen déclenche le mécanisme de reroutage du trafic proposé dans le Chap. 3, alors qu'un risque très important déclenche une reconfiguration du routage beaucoup plus importante tel le changement des rôles des routeurs (routeurs de bordure d'air, routeurs de bordure de domaine BGP, etc.). La configuration extrême, représentant le risque de manière continue, avec une valeur précise de la probabilité de panne n'est pas souhaitable. Compte tenu du mode de fonctionnement des mécanismes proactifs de gestion, une valeur précise de la probabilité de panne n'aurait pas de valeur ajoutée par rapport à une valeur discrète. En effet, les mécanismes proactifs doivent alors décider si cette valeur continue de se situer dans leur intervalle de fonctionnement, ce qui revient à discrétiser ce risque. Il semble donc plus cohérent et pertinent que ce soit le module de risque qui se charge de discrétiser ce risque en plusieurs niveaux. Il est aussi important de bien choisir ces niveaux de risque pour permettre une compatibilité avec le plus grand nombre de dispositifs possibles. Pour cela, nous proposons quatre niveaux de risque : normal, alarme, risqué et très risqué dont chacun peut être associé à des seuils ou des intervalles de probabilité de panne.

- le niveau *normal* correspond au comportement standard sans signe avant-coureur de panne ;
- le niveau *alarme* est le premier niveau de risque, avec la présence de quelques signes anormaux, mais sans réelle erreur. Ce niveau est dédié à des services *premium*, où l'on ne veut prendre le moindre risque de panne, même en dépit d'une intervention coûteuse ;
- le niveau *risqué* correspond à un niveau de probabilité de panne importante, où le mécanisme de panne est presque certain de la réalisation d'une future panne. Les alarmes et erreurs ont été détectées, mais il reste suffisamment de temps pour mettre en place des dispositifs proactifs dont la mise en œuvre ne prend pas plus d'une minute ;
- enfin, le niveau *très risqué* est généralement soulevé moins d'une seconde avant la panne ou même pendant son apparition, lors de la quasi-certitude de la panne. Il n'est alors plus possible de prendre des actions préventives, mais il est d'ores et déjà possible de prendre des actions correctives. De plus, ce type d'information de risque est utile pour aider les actions de diagnostic et d'isolation de panne dans leurs tâches de localisation de la faute.

Associé à ce niveau de risque, l'information de risque nécessite de contenir un identifiant de la ressource risquée, qui peut être un élément de réseau tel un routeur, un commutateur, une carte de ligne, une fibre, mais aussi un ensemble logique d'éléments tel qu'un système autonome, une aire OSPF ou un service. Enfin, il peut contenir des paramètres optionnels comme l'origine de la prédiction, à savoir quels paramètres ou alarmes sont à l'origine de la prédiction, afin d'aider les mécanismes de localisation des fautes.

Néanmoins, dans cette thèse, nous ne considérerons que trois mécanismes proactifs pour exploiter cette information de risque. Dans cette optique et compte tenu de leur mode de fonc-

tionnement que nous décrierons plus tard, nous considérerons une information de risque, avec seulement deux niveaux, à savoir, normal lorsqu'aucune future panne n'est envisagée et risqué, lorsque le module de risque pense qu'une panne va apparaître.

De plus, le sujet de cette thèse n'est pas de proposer une nouvelle technique de prédiction de pannes qui ont déjà fait l'objet de nombreux travaux, mais de contribuer à l'évaluation de trois nouveaux dispositifs proactifs appliqués aux réseaux de cœur IP. Pour ce faire, nous considérons le RAM comme une boîte noire, sans se soucier de son fonctionnement interne. Néanmoins, l'étape de prédiction jouant un rôle considérable dans les processus d'autoréparation proactifs, nous considèrons la prédiction de pannes au travers de ses métriques de performances (*Recall* et *Precision*) et des périodes de temps influençant son fonctionnement (plus particulièrement Δt_p). En effet, comme expliqué à la Sec. 1.6.2.1, les trois mécanismes proposés dans les chapitres suivants ont des délais de mise en œuvre similaires et extrêmement rapides, typiquement inférieurs à cinq secondes. Cette thèse considére donc un Δt_l d'environ cinq secondes pour tous les mécanismes proposés par la suite. Pour des raisons de simplicité, le paramètre Δt_d qui concerne le fonctionnement interne du module de risque ne rentre pas en compte dans les mécanismes développés dans la suite. Enfin, le paramètres Δt_p est pris en compte dans les évaluations des mécanismes proactifs, car en définissant la durée de validité d'une prédiction, il influence de manière claire le mécanisme exploitant la prédiction.

Dans le cadre de l'étude de l'autoréparation proactive, tous les dispositifs proposés exploitent en premier lieu un module de prédiction de pannes. Alors que cette thèse se concentre sur les mécanismes proactifs, il est apparu indispensable de détailler les grands principes de cette étape de prédiction qui suscite bien des interrogations quant à sa faisabilité. Bien que ces interrogations paraissent justifiées au regard du peu d'outils de prédiction de pannes disponibles dans les équipements proposés par les constructeurs aujourd'hui, les nombreux travaux présents dans la littérature rassurent sur leur faisabilité dans un futur pas si lointain. En attendant, nous modélisons cette fonction de prédiction au travers des métriques de performance que sont le *Recall* et la *Precision*, ainsi qu'avec le paramètre Δt_p qui définit la durée de validité d'une prédiction.

1.7 Conclusion

Dans ce chapitre, il à été identifié que la gestion actuelle des réseaux souffre d'un certain nombre de maux. Pour y remédier, la communauté de recherche propose d'introduire de l'autonomie afin que le réseau réalise une autogestion visant à sortir l'être humain de la plupart des tâches de gestion. Cette approche laisse envisager de nombreux avantages, notamment dans la gestion des pannes. En effet, lors de l'occurrence d'une panne, la capacité d'analyse et la rapidité d'exécution des systèmes informatiques permettent une gestion bien plus efficace, avec laquelle on peut même anticiper les pannes pour réaliser une gestion proactive. Afin de déployer des fonctions de gestion autonome à grande échelle, il est nécessaire de concevoir une architecture définissant un certain nombre de blocs de base dont les nombreuses propositions ont été recensées. Il en est de même pour la gestion des pannes qui nécessite, au sein de l'architecture d'autogestion, des composants adaptés aux spécificités de la gestion des pannes. Nous proposons donc UAFAReS, une architecture de gestion autonome des pannes, intégrée à l'architecture GANA. Cette architecture définit des composants permettant la gestion des pannes de manière autonome. Elle comprend notamment un module d'évaluation des risques de panne qui exploite la prédiction de pannes pour rendre possible la réalisation de dispositifs d'autoréparation proactifs tels ceux étudié dans les chapitres suivants.

Chapitre 2

Une détection des pannes plus autonome

Sommaire

2.1 Introduction

Les réseaux IP représentent aujourd'hui une grande partie des réseaux déployés dans le monde et de plus en plus d'opérateurs investissent dans des réseaux 100% IP. Ils présentent l'avantage d'être simples et rapides à mettre en place grâce ses nombreux mécanismes d'autogestion [RNCS09], d'être peu couteux, et de permettre une grande flexibilité. Néanmoins, le protocole IP souffre de quelques défauts qui poussent les opérateurs à lui associer le protocole MPLS pour leurs réseaux critiques. Outre l'ingénierie de trafic, le protocole IP pêche dans la gestion des pannes qui est un paramètre essentiel dans la QoS assurée par les opérateurs à leurs clients. La gestion de la topologie et des pannes, s'appuie sur le protocole *Hello* qui n'est pas sans défaut. Le protocole *Hello* est en grande partie responsable de la durée d'interruption de service, causée par une panne, qui s'élève à plusieurs secondes. Une telle indisponibilité n'est pas acceptable pour de nombreux trafics très sensibles à la QoS. Malheureusement, le protocole *Hello* qui détecte les pannes en envoyant des messages de contrôle à intervalles réguliers ne permet une détection rapide qu'en accélérant la fréquence d'envoi de ces messages au dépend de la stabilité du réseau.

Pour résoudre ce problème, ce chapitre propose d'utiliser une information de risque de panne en temps réel pour adapter dynamiquement la fréquence d'envoi des messages *Hello* [VGD11]. Après avoir décrit le contexte, l'état de l'art, ainsi que la proposition, une évaluation théorique du dispositif est proposée *via* un modèle analytique, puis instanciée sur trois réseaux d'opérateurs. Dans un deuxième temps, le mécanisme est implémenté dans un simulateur afin de valider les hypothèses faites dans l'étude théorique et d'étudier son comportement de manière plus fine.

2.2 Gestion des pannes dans les réseaux IP

L'Internet tel qu'on le connaît est un ensemble de réseaux interconnectés par des routeurs qui ont pour fonction de transférer les données vers leur destination. Pour réaliser cet objectif, les routeurs utilisent des protocoles d'échange afin de partager les informations de routage entre eux. Afin de limiter ces échanges, les routeurs sont divisés en zones appelées système autonome AS [1]. À l'intérieur d'un système autonome, un protocole de routage de type IGP est responsable de la construction et de la maintenance des informations de routage. Les protocoles IGP sont de deux types :

- protocole à vecteur de distance
  Chaque routeur échange périodiquement les destinations qu'il peut atteindre. Cela a pour avantage de limiter la quantité d'information échangée au détriment d'un temps de convergence important. Un exemple type est le protocole RIP [2].
- protocole à état de liens
  Chaque routeur diffuse de manière périodique les informations sur ses voisins. Les informations de toute la topologie (*i.e.* liens, nœuds, métriques, etc.) sont diffusées à tous les routeurs afin que chacun puisse en son sein calculer les plus courts chemins vers tous les routeurs de l'AS. L'avantage est une convergence beaucoup plus rapide, mais ceci est réalisé au dépend d'un échange d'informations massif et d'une tâche de calcul plus importante. Les protocoles usuellement utilisés sont IS-IS [Ora90] et OSPF [Moy98]. Les protocoles à état de lien permettent aussi un partitionnement hiérarchique en subdivisant le domaine autonome en aires de routage ce qui permet de contenir le problème de diffusion des informations de topologie à un sous-ensemble du domaine.

Dans les réseaux actuels, les protocoles IGP les plus populaires sont IS-IS [Ora90] et OSPF [Moy98]. Ils sont tous les deux conçus de manière similaire. Dans un souci de cohérence, nous nous focalisons sur OSPF mais tous les concepts étudiés de cette thèse sont également applicables à IS-IS.

1. *Autonomous System*
2. *Routing Information Protocol*

2.2.1 La détection de panne, une étape critique dans le mécanisme de convergence

Dans les réseaux IP, les routeurs utilisent des protocoles IGP afin de construire leurs tables de routage. Chaque routeur d'un IGP à une vue complète de la topologie. Ils peuvent ainsi connaître les plus courts chemins vers n'importe quelle destination. Des métriques de routage sont utilisées pour définir le point de chaque lien. Ces métriques jouent un rôle important dans le routage des flux. Le calcul de route est effectué en exécutant un algorithme de SPF [1] tel que l'algorithme de Dijkstra. Seul le prochain saut est ensuite utilisé pour la commutation des paquets. Pour des raisons de passage à l'échelle et de rapidité de convergence, un domaine autonome est divisé en aires de routage. Les aires de routage sont organisées en étoile autour d'une aire de routage principale nommée « aire 0 »ou *backbone*. Les autres aires de routage sont connectées à l'aire 0 et chaque route ayant une source et une destination dans deux aires différentes doit passer par l'aire 0. Des routeurs ont des interfaces dans plusieurs aires de routage, ce sont des routeurs de bordure d'aire (ABR [2]), alors que les routeurs en bordure de système autonome sont appelés routeur de bordure de système autonome (ASBR [3]). Le découpage du routage en aires multiples permet de ne nécessiter que la vue complète de la topologie de l'aire. Mais la pertinence des chemins dépend de la fiabilité du graphe, et donc de la performance du mécanisme en charge de collecter les informations sur la topologie. Un des aspects critiques est la détection des modifications dans la topologie, notamment dans le cas d'une panne. Chaque routeur observe l'état de ses routeurs adjacents et diffuse des messages d'état de liens dès qu'un changement est observé (par exemple une panne ou une réparation). Lors d'un changement de topologie, les routeurs recalculent les plus courts chemins afin de connaître les prochains sauts à utiliser pour emprunter le plus court chemin. Cela permet de modifier la table de commutation (FIB [4]) en mettant à jour les interfaces qui sont utilisées pour le transfert des paquets.

Aussi bien OSPF qu'IS-IS utilisent le protocole *Hello* de détection de pannes paire-à-paire. Le protocole *Hello* est basé sur l'envoi périodique de messages *Hello* de chaque routeur à l'ensemble de ses voisins. Les différents paramètres impliqués dans la configuration du protocole *Hello* ont une incidence sur le temps de détection d'une panne ainsi que le rétablissement d'une vue cohérente de la topologie du réseau. Pendant cette période, entre la panne et le rétablissement d'une représentation du réseau conforme à la réalité, les paquets IP transférés vers le routeur en panne sont perdus. Il est donc important de réduire cette période afin minimiser la quantité de trafic perdue. De plus, des boucles de routage peuvent apparaître pendant cette période instable et engendrer de la congestion dans le réseau. La détection d'une panne et la mise à jour du routage prend des dizaines de secondes dans les réseaux d'opérateur. Il est utile de voir précisément quels processus et quels paramètres sont impliqués et quels sont leurs incidences sur le temps de rétablissement d'une panne. Le protocole OSPF est ici pris en exemple, mais le protocole IS-IS fonctionne de manière similaire.

Quatre étapes sont nécessaires pour la restauration du routage IP. En premier, la détection de la panne, puis viens ensuite la diffusion de cette panne à tous les routeurs de l'aire de routage par l'envoi de messages LSA [5]. Une fois le LSA reçu, chaque routeur calcule ses nouveaux plus courts chemins et met à jour ses tables de routage et de commutation en conséquence. Le délai total de convergence dépend du temps pour compléter chacune de ces étapes. Mais ces étapes ne sont pas équivalentes en termes de durée, la détection des pannes étant de loin l'étape la plus longue.

La détection de pannes s'appuie sur le protocole d'échange de messages *Hello*. Deux routeurs sont adjacents tant qu'ils peuvent échanger des messages *Hello*. Le voisinage n'est plus valide lorsqu'un routeur ne reçoit plus de message *Hello*. Dans certain cas, le protocole de niveau inférieur permet de détecter la panne de manière plus rapide. Mais ce mécanisme n'est pas toujours disponible et ne permet pas de détecter certaines pannes au sein du routeur telles que

1. *Shortest Path First*
2. *Area Border Routers*
3. *Autonomous System Border Router*
4. *Forwarding Information Base*
5. *Link State Advertisements*

les pannes du contrôleur ou les pannes logicielles. Chaque routeur envoi des messages *Hello* avec une période définie par le paramètre *Hello Interval*. Ce paramètre a une valeur par défaut de 30 secondes mais peut descendre jusqu'à 1 seconde. Il est en général de l'ordre de la dizaine de seconde. Les voisins qui reçoivent ces messages *Hello* décident de supprimer un routeur de la topologie lorsqu'ils ne reçoivent pas de message *Hello* de sa part pendant un intervalle défini par le paramètre *Router Dead Interval*. Ce paramètre est communément configuré à 3 ou 4 fois la valeur du *Hello Interval*. Ceci permet de garantir une certaine fiabilité dans la détection de pannes en attendant la non réception de 3 ou 4 messages *Hello* avant de la déclarer.

Lorsqu'un changement de topologie est détecté, l'information est ensuite diffusée aux voisins par l'envoi de LSA. Chaque routeur maintient une vue complète de l'aire OSPF dans sa LSDB [1] contenant les LSA. Chaque LSA représente un lien du réseau. Les routeurs adjacents s'échangent des messages regroupant plusieurs LSA afin de synchroniser leur LSDB. Lors d'un changement de topologie, les routeurs affectés par la panne doivent générer des LSA qui sont diffusés à tous leurs voisins. À la réception d'un nouveau LSA, les routeurs mettent à jour cette base de données, mais doivent aussi rediffuser le LSA reçu aux autres voisins. La réception d'un nouveau LSA déclenche aussi le calcul des plus courts chemins (SPF), puis la mise à jour des FIB nécessaires au transfert des paquets. Comparé à la détection de panne, les autres étapes sont presque négligeables en terme de durée puisqu'elles sont inférieures une centaine de millisecondes, avec 0,03 secondes pour le temps de diffusion des LSA t_F [SG01] et 0,2 secondes pour le délai de mise à jour des tables de routage et de *forwarding* t_U [GRcF03].

Le temps de calcul des plus courts chemins dépend du nombre de nœuds du réseau qui, selon [GRcF03], est égale à $t_{SP} = 2,47.10^{-6} * |N|^2 + 9,78.10^{-3}$ (Cf Eq. (2.3)) où N est l'ensemble des routeurs du réseau. Bien que des délais aient été ajoutés pour limiter le nombre de calculs de SPF, des études montrent que ces délais peuvent être supprimés [AC02].

Le temps de détection t_D est actuellement l'étape la plus pénalisante dans le processus de convergence des protocoles IGP. Elle peut être bien plus rapide en utilisant les protocoles de détection des couches inférieures mais de tels dispositifs ne sont pas tout le temps disponible et ne permettent pas de détecter des pannes au niveau logiciel du routeur ou du contrôleur. Le protocole *Hello* utilisé dans les protocoles IGP est contrôlé par deux paramètres. Le *Hello*

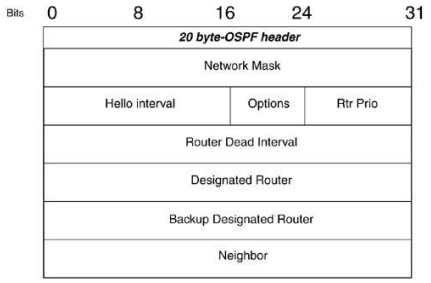


FIGURE 2.1: Message *Hello*.

Interval et le *Router Dead Interval* sont présents dans le message *Hello* représenté dans la Fig. 2.1. Le *Hello Interval* définit la période d'envoi des messages et le *Router Dead Interval*, l'intervalle nécessaire avant de décider de la suppression d'un routeur de la topologie suite à la non réception de messages *Hello*. La définition du protocole *Hello*, telle que présentée dans les Sec. 9.5 et 10.5 de la RFC2328 [Moy98] pour le protocole OSPF, précise que ces deux paramètres doivent être similaires sur toute l'aire de routage et ne peuvent pas changer dynamiquement. Toute nouvelle valeur différente est ignorée par le protocole *Hello*. De plus, il n'est pas possible de définir une valeur inférieure à la seconde pour le *Router Dead Interval* et pour le *Hello Interval*. Néanmoins, la configuration du *Hello Interval* en dessous de la seconde a été implémentée par certains industriels tels que Cisco au sein de leurs routeurs [Sys02]. Les paramètres que nous avons décris concernent le protocole OSPF mais le protocole IS-IS fonctionne de manière tout à fait similaire. Bien que le *Hello Interval* puisse être de seulement une seconde, les opérateurs préfèrent utiliser des valeurs supérieures, entre 3 et 10 secondes [AC02]. En fait, l'utilisation

1. *Link State DataBase*

d'une fréquence élevée pour l'envoi des messages *Hello* engendre de fausses détections de panne, ce qui créé des oscillations dans le routage [OBOM03, BR01, SVKD00].

2.3 Problématique

Le temps de convergence des réseaux IP a une incidence directe sur la disponibilité du trafic. La réduction de ce temps de convergence est un problème qui a largement été étudié par la communauté de recherche. L'étape du processus de restauration la plus pénalisante en termes de temps est la phase de détection. Il est possible de détecter de manière extrêmement rapide une panne, de manière matérielle par l'intermédiaire des protocoles inférieurs ; mais ces mécanismes ne sont pas toujours disponibles et ne permettent pas de détecter l'ensemble des pannes. Par exemple, si une panne implique le processeur de route centrale, mais que les cartes de lignes sont toujours fonctionnelles, la détection matériel est inefficace. C'est pourquoi la détection de pannes s'appuie sur le protocole de détection de pannes *Hello* dont les délais de détection sont de plusieurs secondes. Ce délai correspond à une grande partie de l'interruption de service imputé au processus de convergence, ce qui en fait la cible privilégiée pour l'amélioration du mécanisme de restauration IP. Il est possible d'améliorer la rapidité de détection de pannes en accélérant la fréquence d'envoi des messages *Hello*.

Malheureusement, la réduction du *Hello Interval* pose plusieurs problèmes. La nécessité d'envoyer et de recevoir des messages *Hello* après quelques millisecondes créé une charge additionnelle non anodine pour le contrôleur principal ou le processeur de contrôle. Le temps CPU dédié au traitement des messages *Hello* peut devenir important et venir perturber les autres tâches effectuées par l'unité de calcul.

En effet, la limitation des tâches effectuées par le protocole de routage reste toujours valide. OSPF, en tant que protocole distribué, nécessite l'exécution par les routeurs de certaines opérations dans un temps limité, telles que la génération et le traitement des messages *Hello*. Il est donc essentiel que les routeurs ne soient pas trop chargés, sinon ils pourraient ne pas être en mesure d'assurer correctement leurs fonctions, ce qui pourrait entraîner le dérèglement du routage au niveau global . Pour éviter ce cas de figure, un grand nombre de routeurs possèdent une architecture distribuée où un processeur central exécute le protocole de routage et où le transfert des paquets est assuré par les cartes de lignes. La charge de traitement dédié au protocole OSPF dépend de la taille de l'aire de routage et du nombre de voisin du routeur. Bien que les processeurs des routeurs soient plus puissants que par le passé, la taille et la complexité des domaines de routage rendent la surcharge d'un routeur possible.

De plus, plus le *Hello Interval* est faible, plus la chance qu'une congestion du réseau amène à la perte de plusieurs messages *Hello* consécutifs augmente. Une telle congestion peut se manifester dans des éléments de réseaux intermédiaires, tels que les *switchs*, mais aussi au sein des routeurs au niveau des files d'attente de la carte de lignes ou des files d'attentes de la carte de contrôle, par exemple à cause de la surcharge de travail apportée par le traitement d'un nombre massif de messages *Hello*. Le résultat d'une fausse détection entraîne le retrait erroné d'un voisinage alors que celui-ci est parfaitement fonctionnel. Le LSA généré par la fausse détection amène à de nouveau calcul de route afin de ne plus emprunter le lien faussement supprimé. La fausse alarme est ensuite rapidement corrigée par l'échange des messages *Hello*, ce qui déclenche le rétablissement du voisinage supprimé, la diffusion d'un nouveau LSA ainsi que le re-calcul des tables de routage. En résumé, les fausses détections causent des changements temporaires dans les chemins de routage [OBOM03, BR01, SVKD00] ainsi qu'une charge de calcul supplémentaire et inutile pour les routeurs. Les changements de route fréquents avec intervalle très faible ont un sérieux impact sur la QoS du trafic car les routes temporaires peuvent introduire de mauvais délais de transmission ainsi que des pertes dues à la congestion apportée par ces changements de route fréquents. L'instabilité créée par ces fausses détections étant trop pénalisantes, les opérateurs préfèrent utiliser une fréquence lente d'envoi des messages *Hello* au détriment d'une restauration rapide. Les valeurs les plus couramment utilisées sont de l'ordre de quelques secondes [AC02] et peuvent même atteindre plusieurs dizaines de secondes dans le cas de réseaux particulièrement instables.

2.4 Amélioration du temps de détection des pannes dans les protocoles de routage à état de lien

L'amélioration du temps de convergence est un sujet largement étudié [RMD05, FFEB05]. Parmi les améliorations proposées, la détection de pannes, qui constitue actuellement le point faible de la restauration IP, a été l'objet d'une attention toute particulière. La solution proposée est la diminution du *Hello Interval* jusqu'à des valeurs inférieures à la seconde [GRcF03, AYJ00] afin d'accélérer la convergence des protocoles IGP d'un ordre de grandeur. Mais, comme souligné dans la section précédente, l'accélération de la fréquence d'envoi des messages *Hello* peut amener des instabilités de routage. Cette instabilité, étudiée par Basu et Riecke dans « *Stability issues in OSPF Routing* »[BR01] a le désavantage d'empirer la situation en terme de perte de paquets et peut entraîner une réaction en chaîne qui crée alors un envoi massif de LSA paralysant tout le réseau [Wor01, Pre98, Cho99, Rea01].

En considérant ce problème, des études ont cherché à déterminer quels étaient les réglages optimaux dans le protocole *Hello*, pour le *Hello Interval* et le *Router Dead Interval*. Les simulations de Basu et Riecke montrent par exemple que la réduction du *Hello Interval* de 500 millisecondes à 250 millisecondes ne provoque pas de surcharge significative du processeur mais génère six fois plus d'oscillations de routage à cause des fausses détections. Les travaux de Chouldhury et al. [CMS01] s'attachent à évaluer le problème d'envoi massif en chaîne de LSA (LSA *storm*). Lors d'une ou plusieurs pannes, la génération des LSA entraîne une utilisation intensive des processeurs routeurs et la diffusion en chaîne des LSA dans tout le réseau. Cette période d'activité intense peut engendrer des ralentissements dans le réseau qui, couplés avec une fréquence d'envoi des messages *Hello* élevée, peut entraîner de fausses détections de panne. Ces fausses détections de panne entrainant à leur tour l'envoi de nouveaux LSA, une réaction en chaîne peut se déclencher, déréglant le routage et entrainant des pertes importantes. Malgré le fait que des études aient essayé de définir une valeur optimale pour le *Hello Interval*, celle-ci dépend de nombreux facteurs. Entre autre, l'étude de Goyal et al. [GRcF03] montre que la fréquence des fausses détections de panne augmente avec le niveau de congestion du réseau et avec le nombre de liens dans le réseau. Dans « *On parameter settings of network keep-alive protocol for failure detection* »[Qua09], l'impact d'un faible *Hello Interval* sur le nombre de fausses prédictions est étudié, notamment dans le cas de congestion. L'utilisation d'une fréquence d'envoi élevée n'est recommandée que dans le seul cas où le réseau ne subit pas de ralentissement tel que la congestion. Dans le cas contraire, une fréquence d'envoi faible doit être utilisée, ou un dispositif de suppression des oscillations de routage doit être mis en place. Pour régler ce problème d'oscillation de routage, plusieurs solutions ont été proposées. La majorité de ces solutions prônent l'adaptation dynamique de la fréquence d'envoi à la congestion observée dans le réseau. Dans « *Improving Network Convergence Time and Network Stability of an OSPF-Routed* »[SN05], cinq niveaux de *Hello Interval* sont prédéfinis et associés à cinq niveaux de congestion. Ainsi, lors de la détection d'un certain niveau de congestion, la fréquence d'envoi des messages *Hello* est immédiatement ajustée suivant les valeurs prédéfinies afin d'éviter les fausses détections. Néanmoins, cette solution considère que la sensibilité d'envoi des messages *Hello* à la congestion est similaire sur chaque équipement, ce qui n'est pas le cas dans les réseaux actuels possédant des routeurs très hétérogènes, de vendeur, de qualité et d'ancienneté différente. L'article « *A Route Flap Suppression Mechanism Based on Dynamic Timers in OSPF Network* »[WCLL08] propose une autre stratégie basée sur l'évaluation de l'instabilité du réseau. Dans le cas d'un nombre trop important de fausses détections, la fréquence d'envoi des messages *Hello* est progressivement abaissée sur le lien, ou le routeur concerné, jusqu'à ce que l'instabilité baisse. Cette stratégie permet de conserver une fréquence élevée lorsque c'est possible et d'utiliser une détection moins rapide lorsque cela engendre des oscillations de routage. Le principal défaut est que le mécanisme attend d'observer des oscillations pour entreprendre des actions, ce qui limite le problème mais ne permet pas de l'éradiquer.

Malgré les efforts fournit par la communauté pour définir la fréquence optimale d'envoi des messages *Hello*, les opérateurs conservent des valeurs de l'ordre de la seconde [AC02]. En effet, ces études s'appuient en majorité sur des simulations ou quelques configurations de réseau

spécifiques, peu représentatives de l'étendue des réseaux opérationnels. Dans la pratique, le *Hello Interval* n'est jamais sous la seconde, mais plutôt autour de trois secondes [AC02], allant même jusqu'à une dizaine de secondes, en fonction des caractéristiques de chaque réseau, telles que les performances des routeurs, leur ancienneté, la taille, la charge du réseau, etc.

Un protocole spécifique pour la détection de pannes a été défini par l'IETF afin de dépasser les limites du protocole *Hello* intégré aux protocoles IGP. Le protocole BFD [1] [KW10] permet une détection bien plus rapide en utilisant une fréquence d'envoi de messages *Hello* beaucoup plus rapide. Il peut être utilisé avec les protocoles IGP tel qu'OSPF et IS-IS mais est plus adapté à la détection de pannes le long d'une connexion tel qu'un LSP [2], ou à la détection d'une panne sur un lien bien spécifique, qu'à la détection de pannes sur la topologie de routage dans son ensemble. En effet, il a été créé pour être implémenté de manière matérielle au sein des cartes de lignes afin de décharger le contrôleur du traitement intensif des messages *Hello* que nécessite une fréquence élevée. Néanmoins, il ne permet pas de détecter les pannes qui surviennent au niveau du contrôleur et du logiciel de routage, ce qui limite son utilité. Ses caractéristiques sont néanmoins intéressantes car elles permettent de dépasser les limitations des protocoles *Hello* des protocoles IGP, en autorisant des valeurs inférieures à la seconde et l'utilisation de fréquences d'envoi spécifique à chaque routeur.

2.5 Description de la proposition

Le premier dispositif proposé dans cette thèse exploite l'information de risque de panne fournit par un module RAM pour améliorer l'efficacité de la détection de pannes et par conséquent la rapidité de la restauration IP. Le mécanisme proposé concerne uniquement le protocole *Hello*, et plus particulièrement la période d'envoi de ces messages, définit par le *Hello Interval*. Une fréquence lente d'envoi des messages est nécessaire pour la stabilité, mais une détection rapide permet de réduire sensiblement l'interruption de trafic lors d'une panne. Nous pensons donc qu'il est obligatoire de conserver une fréquence lente mais que les indicateurs annonciateurs de panne peuvent être mis à contribution par les équipements réseaux afin d'accélérer de manière autonome cette fréquence lorsqu'un équipement à des fortes chances de tomber en panne [VGD11]. Nous proposons de conserver un *Hello Interval* semblable à ce qui est utilisé aujourd'hui (supérieur à la seconde) la majorité du temps afin d'assurer un routage stable et lors d'une prédiction de panne, d'accélérer temporairement la fréquence d'envoi des messages *Hello* sur le lien risqué, ou en provenance d'un routeur ou d'une carte de lignes risqués. Cela permet d'être beaucoup plus réactif si une panne a donnée des signes avant-coureurs tout en gardant un réseau stable.

2.5.1 Aperçu général du principe de détection de pannes adaptatif

Dans le contexte des réseaux autonomes, la fonction d'autoréparation AFDT utilise une information de risque de panne afin d'améliorer le temps de détection des pannes. Elle a pour mission d'adapter la réactivité de la détection de pannes au risque de pannes observé afin d'améliorer sensiblement le temps de convergence du protocole IGP tout en maintenant une stabilité acceptable [VGD11].

Comme nous l'avons souligné, le temps de convergence des IGP est fortement couplé au temps de détection d'une panne. Le temps de détection d'une panne peut être extrêmement rapide avec l'utilisation d'une fréquence élevée d'envoi des messages *Hello* équivalente à un *Hello Interval* inférieur à la seconde. Mais la contrepartie est une surcharge inutile des liens et des routeurs, de nombreuse fausses détections de panne ce qui oblige les opérateurs à garder un *Hello Interval* supérieur à la seconde.

L'idée que nous proposons est d'adapter dynamiquement la valeur du *Hello Interval* lorsqu'une prédiction de panne est levée [VGD11]. Grâce à la prédiction de pannes effectuée au sein des RAM, un module de gestion de la fréquence des messages *Hello* localisé au sein du RM_DE a pour mission d'accélérer l'envoi des messages *Hello* par un routeur dont l'un des composants

1. *Bidirectional Forwarding Detection*
2. *Label Switched Path*

(contrôleur, carte de ligne, lien, etc.) va tomber en panne de manière imminente. De façon symétrique, ce module AFDT (Cf Fig. 2.2) est en charge de rétablir une valeur moins agressive de *Hello Interval* que l'on considère comme extrêmement stable lorsqu'aucun risque important de panne n'est détecté.

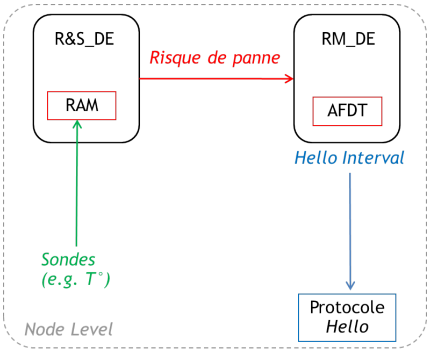


FIGURE 2.2: Architecture fonctionnelle du mécanisme AFDT.

Ce dispositif combine les avantages des deux stratégies de configuration du *Hello Interval*, une détection rapide des pannes lorsque l'environnement est propice aux pannes et un fonctionnement calme et stable lorsque l'environnement est sûr. Car dans une période propice aux pannes et notamment lors d'une panne, l'objectif est d'avoir une table de routage la plus rapidement mise à jour, ce que permet une fréquence élevée d'envoi des messages *Hello*.

En contrepartie, une fréquence élevée est synonyme de surcharge et surtout d'instabilité. En effet une fréquence élevée implique un envoi massif de messages *Hello*, qui ne sont pas forcément nécessaires la majorité du temps, mais qui réquisitionnent du temps de traitement par le contrôleur, et alourdissent donc la charge de ce même contrôleur. De plus, le fonctionnement du protocole *Hello* nécessite la perte de trois ou quatre messages *Hello* pour déclarer une panne. Avec un *Hello Interval* inférieur à la seconde, voire même inférieur à 100 millisecondes, la probabilité pour que ces quatre messages se trouvent ralentis par une congestion temporaire au niveau d'un élément intermédiaire tel qu'un *switch* ou dans le routeur au niveau de la carte de ligne ou du contrôleur lors d'une surcharge importante. Cela créé alors de fausses détections de panne qui créent des instabilités dans le routage (*routing flap*) que les opérateurs ne sont pas prêts à tolérer. Avec un *Hello Interval* supérieur à une seconde pendant les périodes normales, la stabilité est assurée et lors d'une prédiction de panne, l'envoi massif de messages *Hello* est contenu à la durée de la prédiction de panne Δt_p et aux éléments de réseaux risqués.

Un routeur reçoit donc en général des messages *Hello* sur une seule interface et seulement pendant un période limitée. En conséquence, lors d'une panne, on a une restauration beaucoup plus rapide lorsqu'une information de risque a précédé la panne, une restauration semblable à la situation actuelle lorsque la panne n'as pas été anticipée et une faible incidence sur le réseau lorsqu'une fausse prédiction est faite. En effet, lors d'une fausse prédiction, le mécanisme n'a aucune incidence sur le trafic ; seul le protocole *Hello* est modifié et seulement de manière ciblée et limitée dans le temps. Ce mode est sans commune mesure avec l'utilisation d'une fréquence rapide sur tout le réseau pendant 100% du temps. La période de fréquence élevée est réduite aux périodes de risques de panne, ce qui est sans commune mesure avec le fonctionnement standard qui est actif 100% du temps. Le nombre de messages à traiter pour un routeur pendant ces périodes de risque est moindre car la fréquence rapide ne concerne que le lien relié au routeur malade et non les liens avec les autres voisins du routeur. Les voisins du routeur malades étant moins sollicités avec le protocole *Hello*, une surcharge du contrôleur gérant le protocole *Hello* est moins probable, ce qui est positif pour éviter les fausses détections. En somme, l'envoi massif de messages *Hello* est limité dans le temps et l'espace permettant de limiter l'instabilité à une valeur acceptable.

Lors des périodes calmes où aucune panne n'est annoncée, l'objectif est d'avoir une topologie de routage aussi stable que possible, avec peu de messages *Hello* échangés. Une panne dans ce mode génère une interruption de service comparable à ce qui est utilisé dans les réseaux actuels, plus importante qu'en mode réactif mais garantissant une stabilité dans le routage.

2.5.2 Considérations protocolaires

La solution nécessite quelques modifications du protocole *Hello* d'OSPF ou d'IS-IS. Premièrement, il est nécessaire de pouvoir changer dynamiquement les valeurs de *Hello Interval* et de *Router Dead Interval*, ce qui n'est pas possible actuellement, car ces paramètres sont configurés au démarrage du service et ne sont pas prévus pour être modifiés. Un processus de synchronisation de la nouvelle fréquence doit être mis en place, en utilisant le message *Hello* lui-même où en rajoutant des messages spécifiques dédiés à cette synchronisation des *timers*. Deuxièmement, la configuration de ces paramètres doit être spécifique à chaque voisin, afin de pouvoir affecter une fréquence rapide à un seul routeur du réseau, voire à un seul lien. Aujourd'hui les valeurs de *Hello Interval* et de *Router Dead Interval* sont globales à toute une aire de routage de par la spécification du protocole. Enfin, il est aussi nécessaire d'introduire la possibilité de spécifier un *Hello Interval* et un *Router Dead Interval* inférieurs à une seconde.

Le protocole BFD est en grande partie capable d'effectuer ces opérations mais présente le désavantage d'être décorrélé du protocole de routage. De plus, ce protocole a été mis en place afin de pouvoir être implémenté de manière matérielle au sein des cartes de ligne, ce qui lui permet de décharger le contrôleur et donc d'atteindre des fréquences d'envoi et de traitement élevés. En contrepartie, il n'est pas capable de détecter les dysfonctionnements au niveau du contrôleur et du protocole de routage ce qui est un handicap.

Pour notre mécanisme qui permet de limiter l'envoi massif de messages *Hello* dans le temps et dans l'espace, une modification du protocole *Hello* intégré au protocole IGP est la solution la plus performante.

2.5.3 Illustration par l'exemple

Un exemple simple permet de visualiser le mode d'application du dispositif AFDT. La situation stationnaire est représentée à la Fig. 2.3. Il s'agit d'un réseau illustratif comprenant quatre routeurs IP et un flux de trafic allant du routeur 1 au routeur 3 en utilisant le plus court chemin disponible (*i.e.* en passant par le routeur 2). Le protocole *Hello* est utilisé pour maintenir les informations sur la topologie utilisable par le protocole de routage OSPF. Le mode de fonctionnement stationnaire, lorsqu'aucun risque de panne n'est détecté, utilise une période d'envoi des messages *Hello* de trois secondes, ce qui est conforme aux pratiques des opérateurs d'aujourd'hui. Ce mode assure une stabilité maximum avec, dans notre exemple, seulement quarante messages *Hello* par minutes à gérer pour chaque routeur.

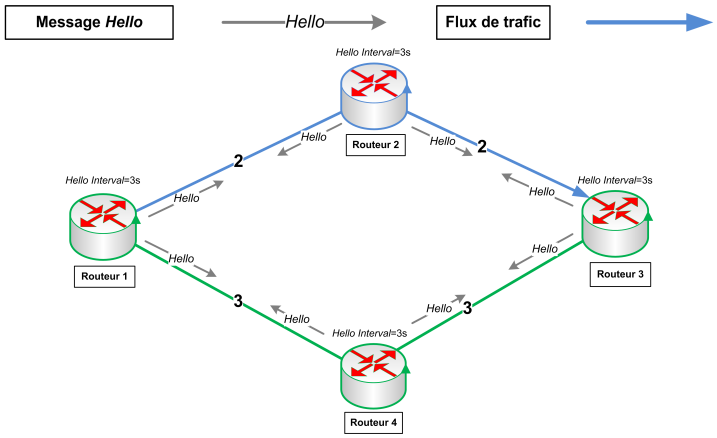


Figure 2.3: Configuration de départ avec une fréquence d'envoi de messages *Hello* lente.

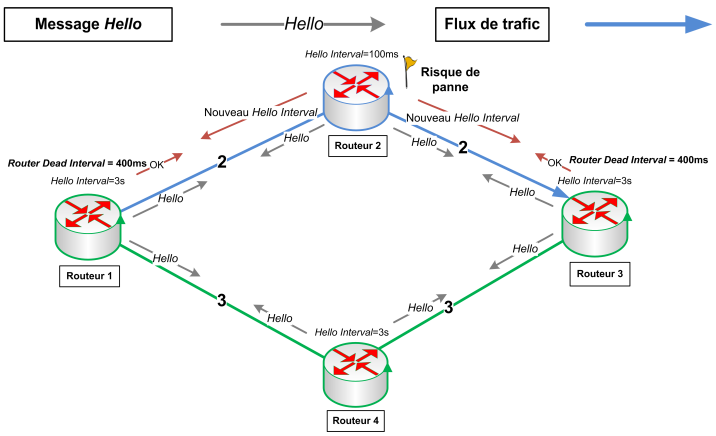


Figure 2.4: Synchronisation de la nouvelle fréquence d'envoi des messages *Hello* vers le routeur risqué.

Ce mode de fonctionnement change de manière autonome lorsqu'un risque de panne est détecté sur un élément du réseau. La Fig. 2.4 illustre l'envoi par l'un des modules d'évaluation du risque RAM présent au sein des routeurs, d'une information de risques de panne, à destination du module AFDT du RM_DE du routeur 2, suite à l'observation d'un comportement symptomatique de panne tel qu'une augmentation anormale de température, une consommation

mémoire anormalement élevée, etc.

Le RM_DE du routeur 2 déclenche alors immédiatement la configuration d'une fréquence d'envoi des messages *Hello* rapide, en se synchronisant avec ses voisins, les informant que sa nouvelle période d'envoi des messages *Hello* est désormais de 100 millisecondes. Les deux routeurs voisins (*i.e.* le routeur 1 et 3) ne modifient par leurs fréquences d'envoi des messages *Hello*, ni la configuration de réception des messages *Hello* en provenance de leurs autres voisins (en l'occurrence le routeur 4), mais doivent mettre à jour la configuration associée à la réception des messages *Hello* en provenance du routeur 2. La mise à jour concerne notamment le *Router Dead Interval* qui est divisé par 30 afin de pouvoir détecter la futur panne en un minimum de temps et ainsi réduire l'interruption de service.

Lors du mode d'envoi à fréquence rapide, si la prédiction concerne un sous ensemble du routeur tel qu'une carte de ligne ou un lien, l'envoi rapide des messages *Hello* ne concerne que les routeurs voisins connectés à ce sous-ensemble. Mais si la prédiction concerne tout le routeur, l'envoi massif de messages *Hello* est adressé à tous les voisins du nœud malade. Au niveau des voisins, seule la fréquence de réception des messages *Hello* en provenance d'un routeur risqué est modifiée, l'envoi des messages *Hello* vers le routeur risqué reste lente, tout comme la fréquence d'envoi et de réception de toutes les autres interfaces non connectées avec le routeur risqué.

Après la synchronisation des valeurs de *Hello Interval* et *Router Dead Interval*, le routeur malade envoie ses messages *Hello* à la nouvelle fréquence rapide, jusqu'à la détection d'une panne, la fin de la validité de la prédiction Δt_p ou bien un changement de comportement faisant re-passer le risque de panne à la valeur stationnaire. Dans ces trois cas, une information de risques de panne annonce le retour à la situation normale, ce qui réinitialise les valeurs du *Hello Interval* et du *Router Dead Interval* à leur valeur initiale. La Fig. 2.5 illustre ce mode de fonctionnement où pendant une période maximum Δt_p, les routeurs 1 et 3 reçoivent 10,66 messages *Hello* par seconde. Dans notre exemple simple, c'est deux fois moins de messages à gérer qu'avec l'utilisation d'une fréquence rapide sur toutes les interfaces.

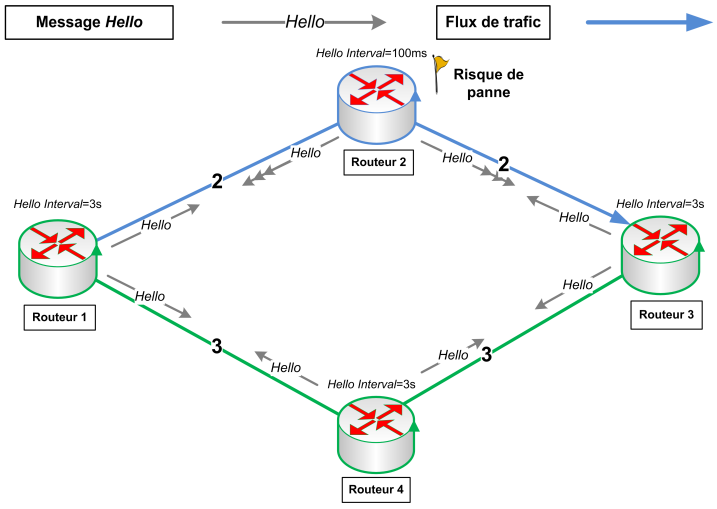


FIGURE 2.5: Envoi des messages *Hello* avec une fréquence élevée par le routeur risqué.

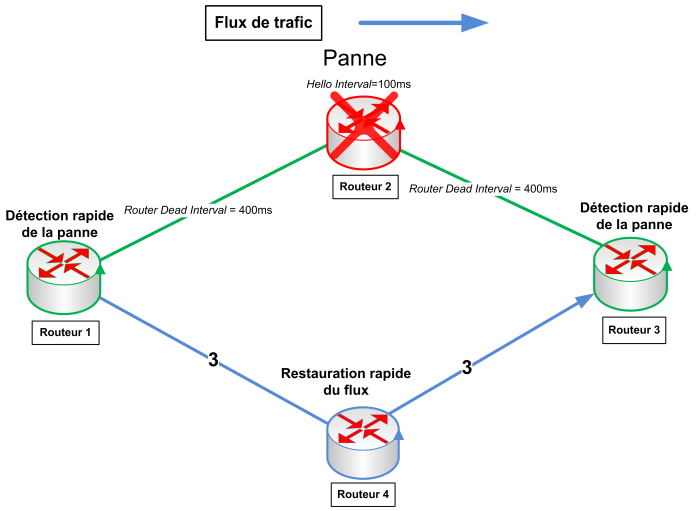


FIGURE 2.6: Restauration IP avec une détection rapide de la panne.

Si la panne apparaît réellement, comme c'est le cas sur la Fig. 2.6, la probabilité de l'avoir prédite est égale au *Recall* et la fréquence rapide d'envoi des *Hello* permet une convergence du protocole IGP rapide et donc une interruption de service minimum. En effet, la détection de la panne est détectée par les routeurs 1 et 3 suite à la non réception d'un message *Hello* pendant seulement 400 millisecondes (au lieu de 12 secondes), et ainsi le routeur 1 peut rapidement terminer sa convergence OSPF et re-router le flux de trafic vers le routeur 4 pour que le service soit rétabli.

Si aucune panne n'apparaît suite à une prédiction de panne, un envoi massif de messages *Hello* aura été fait inutilement pendant une durée Δt_p limitée et sur un périmètre restreint. Dans le cas contraire, la détection de panne aurait nécessité plus de trois secondes, engendrant une

dégradation de service beaucoup plus importante pour l'utilisateur. C'est le cas de figure qui se produit lorsqu'aucune prédiction de panne n'est faite. La topologie de routage reste dans un état stable, avec peu de temps de calcul dédié au traitement des messages *Hello* mais la restauration IP est beaucoup plus longue.

2.5.4 Algorithme

L'Algorithme 1 résume le déroulement du mécanisme AFDT suite à l'envoi d'une information de risque de panne par le RAM au RM_DE. Cette information de risques de panne peut concerner trois types d'équipements, le routeur (noté $n \in N$ où N est l'ensemble des routeurs du réseau), un lien du routeur (noté *link*) ou l'interface à laquelle est rattaché ce lien sur le routeur (noté IF [1]).

algorithme 1 Modification dynamique de la fréquence d'envoi des messages *Hello* en fonction du risque de pannes.

```
1: for each router n_i in N do
2:     if n_i.new_risk > n_i.old_risk then
3:         for each link_j in n_i.links do
4:             if link_j.t_HI = t_SHI then
5:                 configure new Hello Interval with t_FHI
6:                 configure new Router Dead Interval with t_FRDI
7:                 synchronize Hello protocol parameters with the neighbour
8:                 start sending Hello with new frequency
9:             end if
10:        end for
11:    else if n_i.new_risk <n_i.old_risk then
12:        for each link_j in n_i.links do
13:            if link_j.new_risk = normal AND link_j.If.new_risk = normal then
14:                configure new Hello Interval with t_SHI
15:                configure new Router Dead Interval with t_SRDI
16:                synchronize Hello protocol parameters with the neighbour
17:                start sending Hello with new frequency
18:            end if
19:        end for
20:    else if n_i.new_risk = normal then
21:        for each link_j in n_i.links do
22:            if (link_j.new_risk > link_j.old_risk) OR (link_j.If.new_risk > link_j.If.old_risk) then
23:                if link_j.t_HI = t_SHI then
24:                    configure new Hello Interval with t_FHI
25:                    configure new Router Dead Interval with t_FRDI
26:                    synchronize Hello protocol parameters with the neighbour
27:                    start sending Hello with new frequency
28:                end if
29:            else if (link_j.new_risk < link_j.old_risk) OR (link_j.If.new_risk < link_j.If.old_risk) then
30:                if link_j.new_risk = normal AND link_j.If.new_risk = normal then
31:                    configure new Hello Interval with t_SHI
32:                    configure new Router Dead Interval with t_SRDI
33:                    synchronize Hello protocol parameters with the neighbour
34:                    start sending Hello with new frequency
35:                end if
36:            end if
37:        end for
38:    end if
39: end for
```

1. InterFace réseau

2.6 Modélisation analytique

2.6.1 Définitions et notations

Le réseau est représenté par un graphe orienté $G=(N,E)$ où N est l'ensemble des nœuds (routeurs) du réseau et E l'ensemble des liens orientés. Le trafic est représenté par l'ensemble F de flux de trafic transporté par le réseau G où chaque flux de trafic $f \in F$ est définit par son routeur source $In(f) \in N$, son routeur destination $Out(f) \in N$ et son débit $\mu(f)$ en bits/s. Pour chaque flux de trafic $f \in F$, le protocole de routage (OSPF ou IS-IS) définit le plus court chemin (*shortest path*) composé des routeurs de transit par le sous-ensemble $sp(f) \subseteq N$.

Dans ce modèle, les conséquences d'une panne sur les nœuds source et destination ne seront pas prise en compte puisque les mécanismes de résilience intra-domaine n'ont aucun moyen de rétablir une telle situation. Néanmoins, dans les réseaux d'opérateurs, de telles circonstances sont gérées par des techniques de *multi-homing* impliquant une redondance des nœuds d'extrémité et l'intervention de plusieurs réseaux.

La comparaison entre le comportement des trois stratégies de détection de pannes (fréquence lente, fréquence rapide et fréquence adaptative, *i.e.* AFDT) varie suivant deux critères : la probabilité de panne d'un nœud et la prédiction de pannes.

Pour commencer, la probabilité de panne est caractérisée par $MTBF(n)$, le temps moyen entre deux pannes du nœud $n \in N$ et $MTTR(n)$ le temps moyen de réparation du nœud $n \in N$. Le temps de réparation est très inférieur au temps entre deux pannes $MTTR(n) << MTBF(n)$. Pour un processus ergodique stationnaire, la probabilité qu'un nœud $n \in N$ soit en panne $P_{node}(n)$ est :

$$P_{node}(n) = \frac{MTTR(n)}{MTBF(n) + MTTR(n)} << 1 \tag{2.1}$$

Lorsque les routeurs sont considérés comme identiques, le paramètre n est omis dans les notations de tous les paramètres.

Concernant le dispositif AFDT, celui-ci est paramétré par les trois variables de la prédiction de pannes (Cf Sec. 1.6.3) que sont le *Recall*, la *Precision* et Δt_p. Alors que Δt_p est une constante du mécanisme de prédiction, *Recall*(n) et *Precision* sont les variables associées à chaque routeur $n \in N$. Enfin, tous ces paramètres sont considérés comme non modifiables dans le temps.

Lors d'une panne, le protocole de routage intra-domaine enclenche le processus de convergence, qui vise à restaurer les flux de trafic impactés par la panne. Pendant la durée de ce processus noté t_C, les flux $f \in F$ qui passent par le nœud $n \in N$ en panne ne délivrent plus les données à leur destination. Ce processus de convergence est décomposé en quatre étapes. L'étape de détection de la panne t_D, l'étape de diffusion de l'information de la panne à tout le réseau t_F, l'étape de calcul des plus courts chemins t_{SP} et l'étape de mise à jour de la table de routage et de la table de *forwarding*. Compte tenu de toutes ces étapes, le temps de convergence est obtenu en sommant chacune des étapes :

$$t_C = t_D + t_F + t_{SP} + t_U \tag{2.2}$$

Si les étapes t_F et t_U ont des valeurs constantes, ce n'est pas le cas de l'étape de calcul des plus courts chemins et de détection de la panne. Le temps de calcul des plus courts chemins dépend du nombre de nœud du réseau qui selon l'article [GRcF03] est exprimé par la formule suivante pour un routeur de type Cisco 3600 :

$$t_{SP}(N) = 2,47.10^{-6} * |N|^2 + 9,78.10^{-3} \tag{2.3}$$

Néanmoins, pour plus de clarté dans les équations suivantes, on omettra la dépendance avec N en notant t_{SP} le temps de convergence du réseau.

L'étape de détection de la panne dépend de la période entre deux envois de message *Hello* (*Hello Interval*) noté t_{HI} ainsi que de la valeur du compteur *Router Dead Interval* noté t_{RDI}. Pour des raisons de simplicité, le compteur *Router Dead Interval* est bien souvent un multiple de *Hello Interval* tel que $t_{RDI} = 4 * t_{HI}$ afin qu'une panne soit détectée suite à la non réception

de quatre messages *Hello*. En considérant que l'occurrence d'une panne est uniforme entre deux messages *Hello*, le temps moyen de détection d'une panne est :

$$t_D = (t_{RDI} - t_{HI}) + \frac{t_{HI}}{2} \tag{2.4}$$

Dans cette thèse, sachant que $t_{RDI} = 4 * t_{HI}$, l'Eq. (2.4) peut être remplacée par

$$t_D = (3 * t_{HI}) + \frac{t_{HI}}{2} \tag{2.5}$$

Néanmoins dans ce chapitre, les trois stratégies de gestion du *Hello Interval* (lente, rapide et adaptative) impliquent l'utilisation de deux valeurs pour t_{HI} et donc pour t_{RDI}. Pour la fréquence rapide d'envoi des messages *Hello*, t_{FHI} représente la valeur du *Hello Interval* tandis que t_{FRDI} constitue la valeur du *Router Dead Interval* associé. De la même manière, la fréquence lente est définie par t_{SHI} et t_{SRDI}.

Avec t_{FC} le temps moyen de convergence avec une fréquence de messages *Hello* rapide et t_{SC} avec une fréquence lente, les deux formules exprimant le temps de convergence moyen lors d'une panne sur le réseau G sont :

$$\begin{aligned} t_{FC} &= (t_{FRDI} - t_{FHI}) + \frac{t_{FHI}}{2} + t_F + t_{SP} + t_U \\ t_{SC} &= (t_{SRDI} - t_{SHI}) + \frac{t_{SHI}}{2} + t_F + t_{SP} + t_U \end{aligned} \tag{2.6}$$

2.6.2 Données de la comparaison

2.6.2.1 Estimation de l'indisponibilité du réseau

En considérant que le fonctionnement du mécanisme de convergence des protocoles IGP tels que OSPF et IS-IS soient conforme à l'Eq. (2.2), l'indisponibilité moyenne du flux $f \in F$ suite à la panne du routeur $n \in N$ est exprimée par la formule suivante :

$$\begin{aligned} U_{IGP}(f,n) &= P_{node}(n).t_C/MTTR(n) \\ &= P_{node}(n).((t_{RDI} - t_{HI}) + \frac{t_{HI}}{2} + t_F + t_{SP} + t_U)/MTTR(n) \end{aligned} \tag{2.7}$$

Adapté aux stratégies d'envois des messages *Hello* l'Eq. (2.8) donne les formules pour le cas rapide U_F et le cas lent U_S :

$$\begin{aligned} U_F(f,n) &= P_{node}(n).((t_{FRDI} - t_{FHI}) + \frac{t_{FHI}}{2} + t_F + t_{SP} + t_U)/MTTR(n) \\ U_S(f,n) &= P_{node}(n).((t_{SRDI} - t_{SHI}) + \frac{t_{SHI}}{2} + t_F + t_{SP} + t_U)/MTTR(n) \end{aligned} \tag{2.8}$$

Pour le dispositif AFDT, la probabilité conditionnelle est divisée en une somme de deux probabilités concernant d'une part le cas d'une panne prédite (*Recall*) avec une convergence rapide grâce à une fréquence élevée, et d'autre part le cas d'une panne non prédite ($1 - Recall$) où la fréquence moins élevée entraîne une indisponibilité plus grande :

$$\begin{aligned} U_{AFDT}(f,n) =& P_{node}.Recall(n).t_{FC}/MTTR(n) \\ &+ P_{node}(n).(1 - Recall(n)).t_{SC}/MTTR(n) \end{aligned} \tag{2.9}$$

Ce mécanisme permet d'obtenir, lors de la prédiction d'une panne, une indisponibilité égale à celle de la stratégie d'envoi rapide de messages *Hello*, *i.e.* $U_F(f,n)$ et dans les autre cas, (lorsque la panne n'est pas anticipée), une indisponibilité égale à celle de la stratégie d'envoi lente de messages *Hello*, *i.e.* $U_S(f,n)$:

$$U_{AFDT}(f,n) = (1 - Recall(n)).U_S(f,n) + Recall(n).U_F(f,n) \tag{2.10}$$

À partir de l'Eq. (2.10), il est possible de retomber sur les formules du fonctionnement standard des protocoles IGP de l'Eq. (2.8) puisqu'une valeur de $Recall = 0$ donne l'indisponibilité U_F alors qu'une valeur de $Recall = 1$ permet d'obtenir U_S.

En faisant ressortir les valeurs des *Hello Interval* et des *Router Dead Interval*, l'Eq. (2.9) devient :

$$\begin{aligned} U_{AFDT}(f,n) = \frac{P_{node}(n)}{MTTR(n)}.((Recall(n).((t_{FRDI} - t_{FHI}) + \frac{t_{FHI}}{2})) \\ + ((1 - Recall(n)).((t_{SRDI} - t_{SHI}) + \frac{t_{SHI}}{2})) + t_F + t_{SP} + t_U) \end{aligned} \tag{2.11}$$

Ainsi, en considérant que les pannes des routeurs sont des événements indépendants, les indisponibilités moyennes du flux $f \in F$ pour chaque stratégies (F, S, AFDT) sont :

$$\begin{aligned} U_{AFDT}(f) &= 1 - \left(\prod_{n \in sp(f)} (1 - U_{AFDT}(f,n)) \right) \approx \sum_{n \in sp(f)} U_{AFDT}(f,n) \\ U_F(f) &= 1 - \left(\prod_{n \in sp(f)} (1 - U_F(f,n)) \right) \approx \sum_{n \in sp(f)} U_F(f,n) \\ U_S(f) &= 1 - \left(\prod_{n \in sp(f)} (1 - U_S(f,n)) \right) \approx \sum_{n \in sp(f)} U_S(f,n) \end{aligned} \tag{2.12}$$

Cette approximation est valide puisque, suivant l'Eq. (2.1), la probabilité de cas de pannes simultanées d'au moins deux routeurs dans un même réseau est suffisamment petite pour être négligée.

Enfin, pour calculer l'indisponibilité du réseau, il est nécessaire de définir une moyenne de l'indisponibilité de chaque flux f pondérée par le débit de chaque flux $\mu(f)$. X prenant la valeur de chacune des stratégies de gestion du protocole *Hello* ($X = AFDT$, F et S), l'indisponibilité moyenne est donnée par U_X :

$$U_X(G,F) = \frac{\sum_{f \in F} \mu(f).U_X(f)}{\sum_{f \in F} \mu(f)} \tag{2.13}$$

2.6.2.2 Estimation du nombre de messages *Hello* reçus

Une configuration extrême des compteurs *Hello Interval* et *Router Dead Interval* permet une convergence plus rapide afin de fournir une disponibilité plus grande. Malheureusement, lorsque ces valeurs passent en dessous d'une seconde, le protocole *Hello* est beaucoup plus sujet à de fausses détections de panne en partie à cause des messages *Hello* deviennent beaucoup plus exposés à des retards dus au trafic (congestion), ou à la surcharge des éléments de traitement de ces messages de contrôles (file d'attente, processeur du plan de contrôle). Ce problème, bien connu de la communauté de recherche [BR01, SVKD00, OBOM03, WCLL08], est néanmoins difficile à quantifier. En effet, l'hétérogénéité des équipements dont certains sont équipés d'un plan de données séparé du plan de contrôle, les différentes politiques d'ingénierie de trafic, de gestion des files d'attentes, etc ; rendent très difficile la quantification de l'impact des valeurs affectées au compteurs du protocole *Hello* sur l'instabilité du routage et plus particulièrement sur le nombre de fausses détections de panne. Malgré quelques travaux [BR01, SVKD00, OBOM03] mesurant ce phénomène, il nous est apparu préférable, afin d'éviter des approximations vis-à-vis du comportement de la stabilité du routage par rapport aux messages *Hello*, de mesurer le nombre de messages *Hello* reçu par chaque routeur. En effet, l'instabilité a pour principale cause, la réception d'un trop grand nombre de messages *Hello* par seconde par les routeurs. Cela permet de ne pas faire de mauvaise approximation quant à l'instabilité engendrée, tout en fournissant une information utile pour les opérateurs qui pourront, à partir de cette donnée et des performances observées sur leur infrastructure réseau, en déduire l'instabilité que cela pourrait engendrer.

Pour calculer le nombre de messages *Hello* reçus par seconde, il est nécessaire de faire la moyenne des messages reçus par tous les nœuds. Pour cela il est nécessaire de sommer tous les

messages reçus ($\sum_{n\in N} d^-(n)$) et de soustraire les messages n'ayant pas été envoyés pendant les pannes ($\sum_{n\in N} P_{node}(n)*d^+(n)$). Pour un fonctionnement conforme aux standards des protocoles IGP avec une période de *Hello* t_{HI}, cela donne :

$$H_{IGP}(G) = \frac{\sum_{n\in N} d^-(n) - (P_{node}(n).d^+(n))}{|N| * t_{HI}} \tag{2.14}$$

Pour les deux stratégies d'envoi des messages *Hello* standard, l'Eq. (2.14) se traduit par une formule pour le cas rapide H_F et une formule pour le cas lent H_S :

$$\begin{aligned} H_F(G) &= \frac{\sum_{n\in N} d^-(n) - (P_{node}(n).d^+(n))}{|N| * t_{FHI}} \\ H_S(G) &= \frac{\sum_{n\in N} d^-(n) - (P_{node}(n).d^+(n))}{|N| * t_{SHI}} \end{aligned} \tag{2.15}$$

Dans le cas de routeurs identiques, ayant la même probabilité de panne ($P_{node} = \frac{MTTR}{MTBF+MTTR}$), l'Eq. (2.14) peut se simplifier de la manière suivante :

$$H_{IGP}(G) = \frac{|E|.(1 - P_{node})}{|N| * t_{HI}} \tag{2.16}$$

Avec le fonctionnement du mécanisme AFDT, il est nécessaire de considérer le cas des TP et des FP en faisant intervenir le *Recall*, la *Precision* et Δt_p. Avec le *Recall* et la *Precision* définis par l'Eq. (1.1) et (1.2) les probabilités de TP et de FP pour le nœud n sont :

$$\begin{aligned} P_{TP}(n) &= P_{node}(n).Recall(n).\left(\frac{\Delta t_p/2}{MTTR(n)}\right) \\ P_{FP}(n) &= P_{node}(n).Recall(n).\left(\frac{1}{Precision} - 1\right).\left(\frac{\Delta t_p}{MTTR(n)}\right) \end{aligned} \tag{2.17}$$

Les formules de l'Eq. (2.17) permettent de prendre en compte que :

- lors d'une mauvaise prédiction (FP), le nœud n envoie des messages *Hello* à la fréquence rapide ($1/T_{FHI}$) pendant toute la durée de la prédiction définie par Δt_p ;
- lorsqu'une panne est prédite à l'avance (TP), le nœud n envoie des messages à la fréquence $1/T_{FHI}$ uniquement avant l'apparition de la panne. Cette période commence à la date du début de la prédiction (début de Δt_p) et se termine à l'apparition de la panne. En considérant que l'occurrence des pannes est uniformément distribuée pendant Δt_p, *i.e.* entre 0 et Δt_p, la durée moyenne d'envoie des messages *Hello* à la fréquence rapide est $\Delta t_p/2$;
- le reste du temps les messages *Hello* sont envoyés à la fréquence lente $1/T_{SHI}$, excepté lors d'une panne où aucun message n'est envoyé.

Il est alors possible de calculer le nombre de messages *Hello* reçus en moyenne par seconde et par nœud :

$$\begin{aligned} H_{AFDT}(G) = &\sum_{n\in N} \frac{d^+(n)(1 - (p_{node}(n) + P_{TP}(n) + P_{FP}(n))}{|N| * t_{SHI}} \\ &+ \sum_{n\in N} \frac{d^+(n).(P_{TP}(n) + P_{FP}(n))}{|N| * t_{FHI}} \end{aligned} \tag{2.18}$$

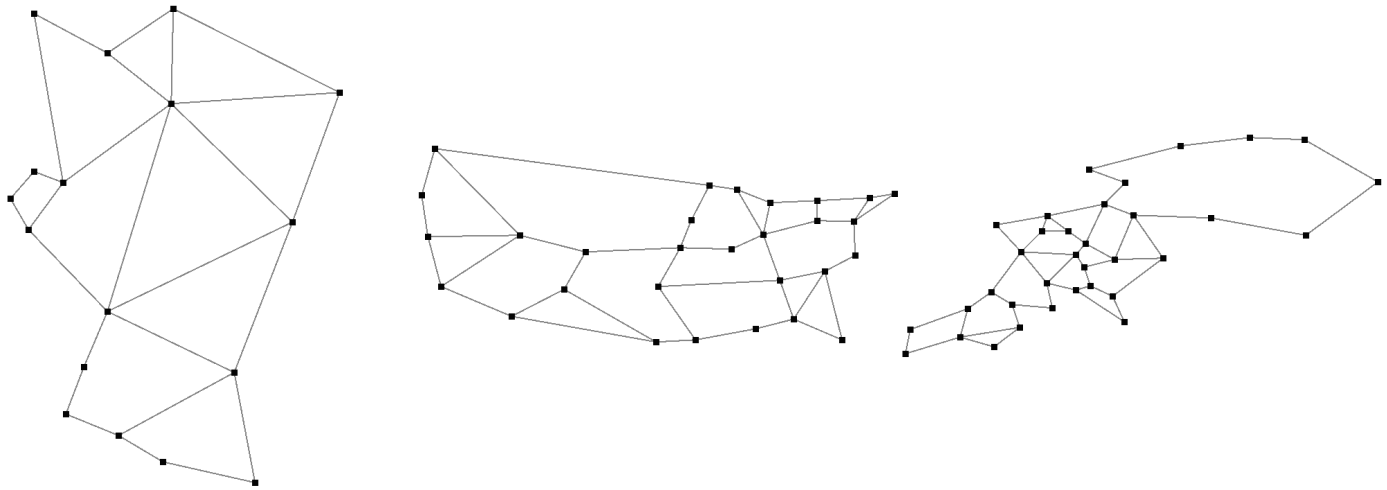
Avec une probabilité de panne unique ($P_{node} = \frac{MTTR}{MTBF+MTTR}$) signifiant que l'on considère les routeurs identiques, l'Eq. (2.18) peut s'écrire de manière simplifiée :

$$H_{AFDT}(G) = \frac{|E|}{|N|} * \left(\frac{1 - (P_{node} + P_{TP} + P_{FP})}{t_{SHI}} + \frac{(P_{TP} + P_{FP})}{t_{FHI}}\right) \tag{2.19}$$

Fort des formules permettant d'estimer à la fois l'indisponibilité du réseau mais aussi la quantité de messages de contrôle *Hello* que reçoit un routeur chaque seconde, il est maintenant possible de comparer les trois stratégies de gestion du protocole de détection de pannes *Hello* sur des configurations réseaux représentatives de la réalité des opérateurs.

2.7 Étude de cas : trois réseaux de classe opérateur

Afin d'analyser le comportement du dispositif AFDT proposé, vis-à-vis du mécanisme standard de restauration OSPF grâce au modèle analytique, trois configurations de réseaux cœurs (Cf Fig.2.7) permettent de mesurer l'impact des variables du modèle. La première topologie est un réseau national allemand (Fig. 2.7a) de 17 nœuds, similaire au réseau utilisé dans [MMJ08]. Le deuxième réseau est la topologie NSF-Net à 29 nœuds (Fig. 2.7b) enfin, la dernière topologie est un réseau européen de 34 nœuds (Fig. 2.7c).



(a) A (b) US (c) EU

FIGURE 2.7: Topologies de réseaux de cœurs étudiées.

En outre, le tableau 2.1 permet de synthétiser quelques caractéristiques additionnelles de chaque topologie telles que le nombre de nœuds ($|N|$), le nombre de liens ($|E|$) le degré moyen ($\bar{d}$), la densité ainsi que le diamètre du réseau ($d(G)$). Une matrice de trafic bien définie est associée à chaque réseau (voir les tableaux. A.2, A.7 et A.12). La matrice de trafic du réseau allemand contient 242 flux pour un total de 1363 Gbit/s, celle du réseau américain 812 flux pour un total de 485 Gbit/s et celle du réseau européen contient 1122 flux pour un total de 1554 Gbit/s (Cf. Tab. 2.1). Enfin la dernière colonne informe sur le nombre d'interfaces de 10 Gbits/s que contient chaque réseau. Le lecteur pourra se référer à l'annexe A et plus particulièrement aux tableaux A.3, A.8 et A.13 pour connaitre le détail de la capacité de chaque configuration réseau.

Réseau	$\|N\|$	$\|E\|$	$\bar{d}$	Densité	$d(G)$	$\|F\|$	Trafic (Gbit/s)	Dimensionnement
A	17	26	3,06	0,19	8	242	1363	626 IF
US	29	44	3,03	0,11	9	812	485	440 IF
EU	34	49	2,88	0,09	14	1122	1554	1244 IF

TABLE 2.1: Caractéristiques des réseaux étudiés.

Enfin, à ces trois topologies sont associées des métriques nécessaires au protocole de routage. Ces métriques, proportionnelles à la distance entre les nœuds, sont décrites en annexe A dans les tableaux A.2, A.7 et A.12.

2.8 Application numérique du modèle analytique

Le modèle de calcul de l'indisponibilité et de la quantité de messages *Hello* reçus a été appliqué aux trois topologies susmentionnées, afin d'étudier l'impact des deux paramètres caractérisant les pannes que sont le MTBF et le MTTR [1] sur le comportement du mécanisme AFDT. De même, les répercussions du *Recall*, de la *Precision* et de Δt_p ont été analysées.

Pour cela, des conditions de références ont été utilisées avec un MTBF de 5000 heures ainsi qu'un MTTR de 5 heures identiques pour chaque routeur. De même, la prédiction de pannes est considérée comme identique sur chaque routeur et avec une durée de validité de référence d'une prédiction Δt_p d'une heure. Bien que l'état de l'art propose des valeurs de Δt_p de quelques minutes [ST08], nous avons constaté qu'une période d'une heure n'engendrait pas de différence significative dans le comportement de notre mécanisme. Étant donné que plus Δt_p est grand, plus la prédiction de pannes est performante, cette valeur d'une heure permet une compatibilité avec la quasi-totalité des dispositifs de prédiction de pannes, et laisse une marge suffisamment importante pour espérer rester compatible avec les futures fonctionnalités de prédiction que proposerons les équipements de réseau. Ce Δt_p d'une heure est donc utilisé pour les neuf configurations de prédiction de pannes de références formées par la combinaison d'un *Recall* et d'une *Precision* de 20%, 50% et 80%.

Enfin, les différents délais des étapes de la convergence du protocole IGP ont été affectés suivants les valeurs relevées dans la littérature [SG01, GRcF03]. Ainsi, 0,03 secondes sont considérées pour le temps de diffusion t_F [SG01] et 0,2 secondes pour le délai de mise à jour des tables de routage et de *forwarding* t_U [GRcF03]. Pour le délai de détection de panne qui nous intéresse, le nombre de messages *Hello* nécessaires pour détecter une panne est fixé à 4, tel que $t_{RDI} = 4 * t_{HI}$. La période d'envoi des messages *Hello*, quant à elle, dépend de la stratégie modélisée. Pour la stratégie utilisant une fréquence lente d'envoi de messages *Hello* la période utilisée correspond à une valeur assurant une stabilité garantie tout en permettant une convergence de l'ordre de 10 secondes. Cela correspond à une période t_{SHI} de 3 secondes. À l'opposé, la période utilisée pour la stratégie de fréquence rapide utilise une période très inférieure à la seconde permettant une convergence extrêmement rapide au détriment de la stabilité. Cette période t_{SHI} est de 100 millisecondes. Comme expliqué précédemment, le mécanisme AFDT utilise alternativement ces deux stratégies en fonction du risque de panne observé par le module RAM.

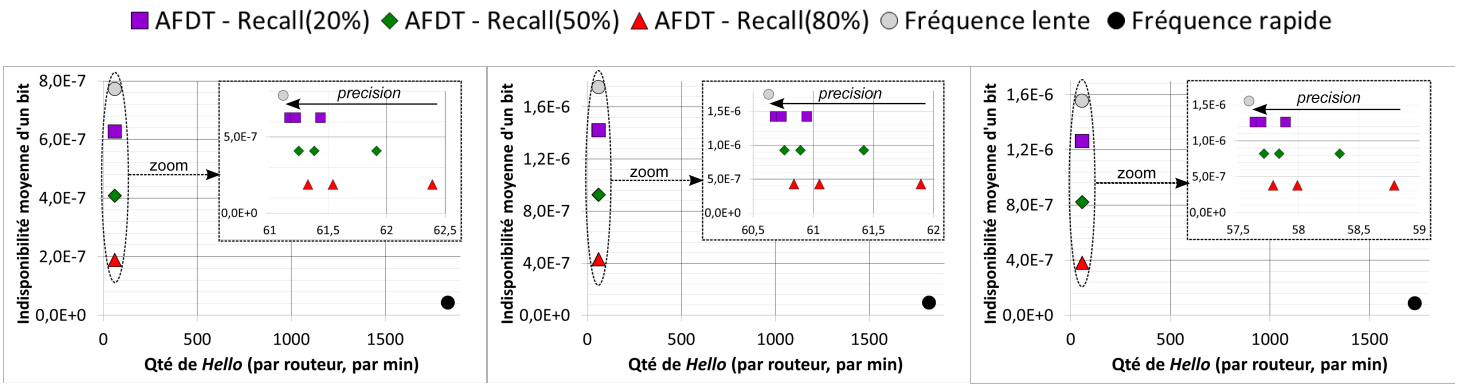
2.8.1 Analyse conjointe de la disponibilité et de la quantité de messages *Hello* reçus

La Fig. 2.8 représente l'indisponibilité et le nombre de messages *Hello* théorique reçus chaque minute dans les conditions de référence décrites à la Sec. 2.8 pour les trois topologies. Les combinaisons des trois valeurs de *Recall* et de *Precision* de 20%, 50% et 80% pour le mécanisme AFDT sont comparées à l'utilisation du protocole *Hello* avec une fréquence lente et une fréquence rapide. Cette figure illustre l'intérêt du dispositif AFDT par rapport au mécanisme utilisé par les protocoles IGP actuels en terme de *ratio* stabilité et disponibilité. Car nous avons vu précédemment qu'un nombre important de messages *Hello* avait une influence sur la stabilité du routage.

Le *Recall* a un impact significatif sur la disponibilité, au contraire de la *Precision*. Concernant la quantité de messages *Hello* à traiter, celle-ci augmente lorsque la *Precision* diminue (voir encadré Fig. 2.8), mais cette augmentation reste négligeable au regard de la quantité à traiter lors de l'utilisation d'un t_{HI} de 100 millisecondes. Cette illustration des conditions de référence pose les bases de l'avantage du mécanisme AFDT sur le comportement des protocoles standards IGP avec un gain d'indisponibilité supérieur à 4 pour le mécanisme AFDT pour un *Recall* de 80%, de quasiment 2 pour un *Recall* de 50% et de 1,4 pour un *Recall* de 20%. Le dispositif AFDT ne permet pas d'assurer une disponibilité égale à l'utilisation continue d'un *Hello Interval* de 100 millisecondes, mais il permet des performances intermédiaires permettant de s'en rapprocher jusqu'à un facteur 4.

1. *Mean Time To Repair*

En contrepartie, l'observation de la quantité moyenne de messages *Hello* à traiter de chaque mécanisme met en valeur l'excellente performance du mécanisme AFDT qui reste très proche de l'utilisation d'une fréquence lente, permettant ainsi de diviser par 30 la quantité de messages à traiter par rapport à la fréquence rapide. Le faible impact de la *Precision* sur la moyenne des messages reçus n'est visible que lorsque l'on zoom sur la Fig. 2.8, car l'effet de l'utilisation d'une fréquence rapide pendant les faux positifs est largement contenue dans le temps et dans l'espace. À la vue de ces premiers résultats, l'utilisation d'un RAM avec un *Recall* le plus important possible au détriment de la *Precision* semble le plus adapté au dispositif AFDT.



(a) A (b) US (c) EU

FIGURE 2.8: *Ratio* indisponibilité / nombre de message *Hello* avec la configuration de référence pour les trois topologies.

2.8.2 Influence de la probabilité de panne

Il est intéressant d'étudier plus particulièrement l'impact de certains paramètres sur le comportement de notre mécanisme AFDT. Bien que les trois topologies aient été étudiées, les différents résultats sont similaires dans leur comportement. Dans un souci de clarté, nous illustrons notre analyse avec une seule topologie, à savoir la topologie européenne. Néanmoins, les résultats complets pour chaque topologie sont disponibles en annexe B.1.

Il est intéressant d'évaluer l'impact théorique de tous les paramètres du modèle sur le comportement du mécanisme AFDT, à commencer par les paramètres caractérisant les pannes, à savoir le MTBF et le MTTR. Pour cela, l'influence du MTBF dans une plage comprise entre 1000 et 10000 heures, ainsi que celle du MTTR pour des valeurs allant de 1 à 10 heures sont analysés en termes de disponibilité et de quantité de messages *Hello* reçus sur chaque routeur.

La Fig. 2.9 montre clairement l'impact de la fréquence des pannes sur la disponibilité. On remarque une augmentation qui s'accélère en dessous d'un MTBF de 3000 heures. Cependant même si l'écart avec la protection s'agrandie, l'ordre et le *ratio* entre chaque mécanisme restent les mêmes que ceux constatés sur la configuration de référence (Cf Fig. 2.8c). La Fig. 2.10 confirme que la durée des pannes ne change en rien l'indisponibilité car dans cette thèse, nous ne nous intéressons qu'à l'indisponibilité due aux délais de convergence des protocoles IGP.

Mais les choses sont légèrement différentes pour la quantité de messages *Hello*. Tout d'abord, le nombre de messages *Hello* que doit traiter un nœud avec la stratégie de fréquence rapide étant beaucoup plus importante que pour les autres mécanismes, cette donnée n'est pas représentée dans les figures illustrant la quantité de messages *Hello*. Néanmoins, son comportement vis-à-vis des messages *Hello* est en tout point similaire à la stratégie utilisant une fréquence lente si ce n'est qu'elle à une valeur 30 fois plus importante. Le lecteur peut se référer à la Fig. 2.8c où il peut noter que chaque nœud de la topologie européenne doit en moyenne traiter 1727 messages *Hello*. Cette valeur correspond à une moyenne importante, trop exposée au moindre ralentissement de traitement aussi bien au niveau du réseau que du contrôleur du routeur. Une augmentation du nombre de pannes ne modifie pas grandement le nombre de messages reçus. Néanmoins, lors d'une panne, un routeur n'est plus en mesure d'envoyer ses messages *Hello*, ce qui est responsable de la baisse visible sur la Fig. 2.11 lorsque le nombre de pannes devient

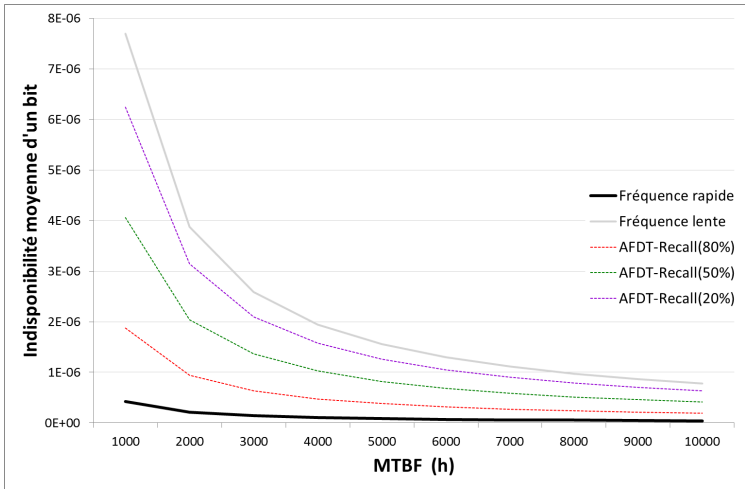


Figure 2.9: Impact du MTBF sur la disponibilité avec la topologie européenne.

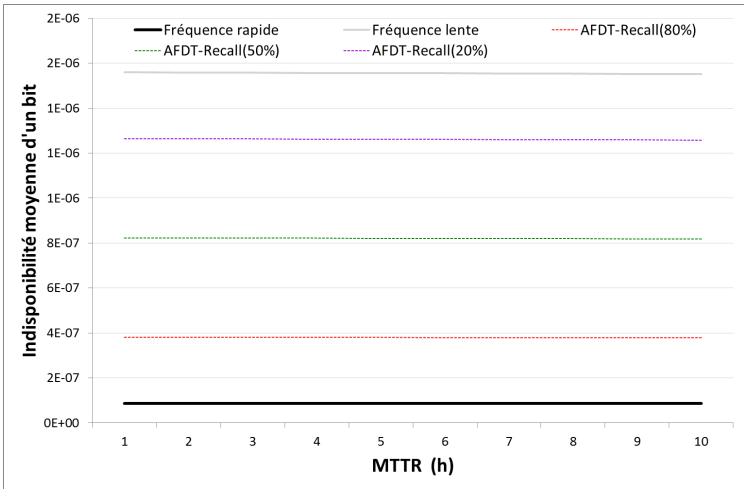


Figure 2.10: Impact du MTTR sur la disponibilité avec la topologie européenne.

important (*i.e.* avec un MTBF faible). Le dispositif AFDT possède un comportement opposé car il accélère la fréquence d'envoi de ses messages *Hello*, avant chaque panne anticipée et lors de chaque fausse prédiction. Le surplus de message *Hello* reste extrêmement faible ($< 4\%$) sur les configurations du mécanisme AFDT ne générant pas trop de faux positif comme le montre la Fig. 2.11. Le surplus d'envoi de messages *Hello* n'est réellement visible que sur les configurations possédant une faible *Precision* (*i.e.* 20%) et un *Recall* pas trop faible (50% et 80%). Néanmoins, ce surplus reste inférieur à 4 % jusqu'à un MTBF de 3000 heures, pour rester en dessous de 64 avec le MTBF minimum (1000 heures), ce qui reste négligeable comparé à la moyenne supérieure à 1500 messages *Hello* observée avec la stratégie de fréquence rapide.

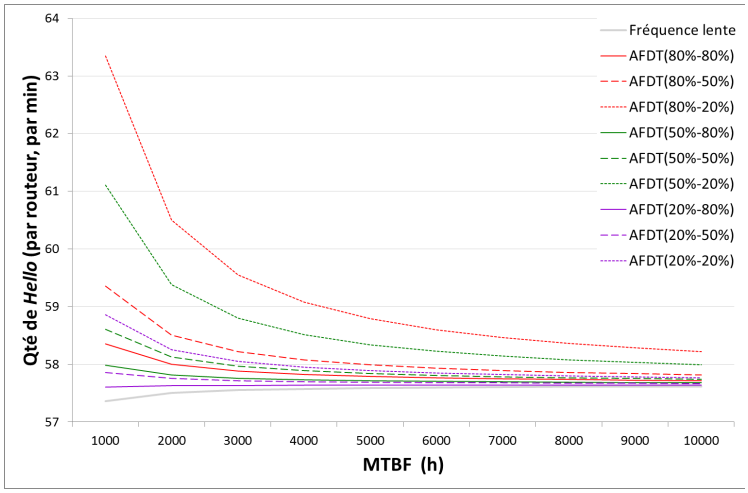


Figure 2.11: Impact du MTBF sur le nombre de messages *Hello* avec la topologie européenne.

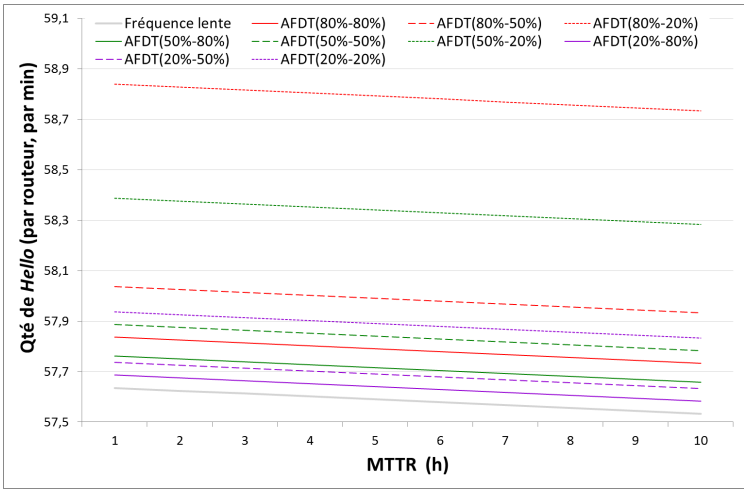


Figure 2.12: Impact du MTTR sur le nombre de messages *Hello* avec la topologie européenne.

La durée de panne n'a qu'une influence minime puisque, lors d'une panne, le routeur hors service arrête d'envoyer des messages *Hello*. Cela engendre un comportement similaire quel que soit le *Recall* ou la *Precision* visible sur la Fig. 2.12 avec une légère baisse (inférieure à 0,2%) sur toute la plage de variation du MTTR de 1 à 10 heures.

2.8.3 Les conséquences de la prédiction de pannes

L'effet des paramètres de prédiction de pannes est l'une des questions importantes de cette thèse. Étant donné qu'aucune méthode de prédiction de pannes associée à notre mécanisme n'est proposée, il est indispensable d'identifier les contraintes que doit posséder le module RAM, ainsi

que le comportement qui en découle, en fonction de ses performances. Pour cela, nous analysons l'incidence de toutes les valeurs du *Recall* et de la *Precision* associées à nos conditions de références, ainsi que le comportement du dispositif AFDT avec un Δt_p variant de 5 minutes à 10 heures.

Le paramètre qui, de prime abord, semble le plus important vis-à-vis du comportement de notre solution, en terme de disponibilité est, le *Recall*. Celui-ci caractérise la proportion de pannes anticipées par le RAM. La Fig. 2.13 montre fort logiquement une augmentation linéaire des performances en fonction du *Recall*. En commençant avec des performances similaires à la stratégie de fréquence faible pour le *Recall* le plus faible, le mécanisme AFDT atteint les performances de la stratégie utilisant la fréquence rapide lorsque toutes les pannes sont anticipées (*i.e.* $Recall = 1$). Le modèle ne considérant aucune incidence de la *Precision* sur la disponibilité, la Fig. 2.13 est donc valable quelle que soit la *Precision* du mécanisme AFDT.

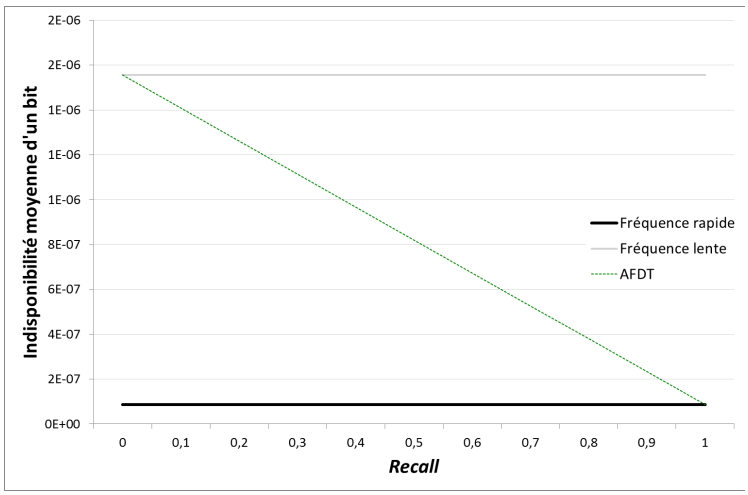


FIGURE 2.13: Impact du *Recall* sur la disponibilité avec la topologie européenne.

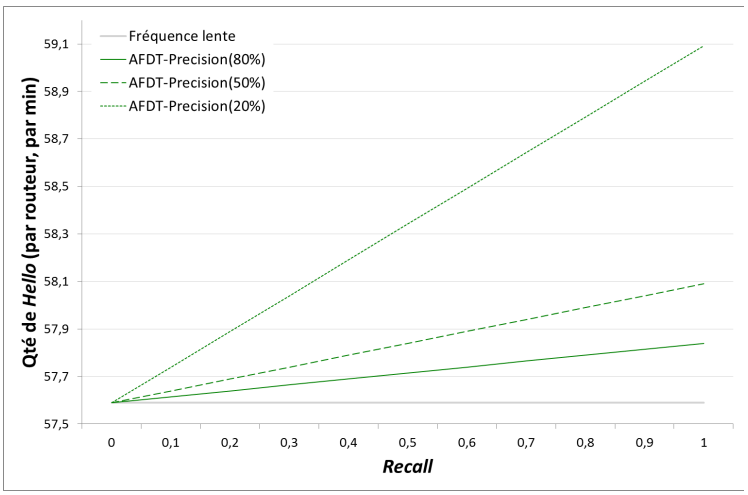


FIGURE 2.14: Impact du *Recall* sur le nombre de messages *Hello* avec la topologie européenne.

En revanche, la situation est différente pour la quantité de messages *Hello* puisque la fréquence s'accélère lors d'une prédiction. Cette accélération temporaire se fait, en moyenne, pendant une durée $\Delta t_p/2$ lors d'une prédiction correcte. Mais elle est aussi présente lors des fausses prédictions. Le nombre de fausses prédictions est caractérisé par la *Precision*, elle-même dépendante du *Recall* (Cf Eq. (1.2)). Cette relation est illustrée sur la Fig. 2.14 qui montre une augmentation du nombre de message avec le *Recall*. Cette augmentation n'est pas seulement due au surplus de messages envoyés pendant $\Delta t_p/2$ secondes avant les pannes anticipées, mais plutôt aux fausses prédictions. En effet on peut observer que l'augmentation est plus forte pour les *Precision* faibles et notamment pour le mécanisme AFDT avec une *Precision* de 20%. Mais cette augmentation ne représente qu'une augmentation de 0,26% par rapport à la fréquence lente dans le pire cas, soit une valeur toujours 30 fois moins importante qu'avec le mécanisme utilisant une période de 100 millisecondes.

La Fig. 2.15 montre l'incidence de la *Precision* sur le nombre de fausses prédictions et donc sur le nombre moyen de messages *Hello* à traiter avec une accélération en dessous de 30%. Le paramètre clé étant la quantité de fausses prédictions, le *Recall* influence indirectement les trois courbes. À *Precision* égale, le mécanisme avec le *Recall* le plus important provoque plus de fausses prédictions et donc une quantité de messages *Hello* plus importante. Mais conformément aux autres observations, cette quantité reste de 60 messages *Hello* par nœud et par minute dans le pire des cas, encore bien loin de la stratégie agressive utilisant un faible *Hello Interval* et ses 1727 messages par minute.

Dans notre configuration de référence, nous avons délibérément choisi un Δt_p avec une marge afin d'éviter de contraindre notre mécanisme à un sous-ensemble trop restreint de mécanismes de prédiction. D'après notre modèle, le Δt_p ne modifie pas la disponibilité mais implique une durée plus longue pendant laquelle la fréquence d'envoi des messages *Hello* est rapide. La Fig. 2.16 met en évidence une variation allant jusqu'à 21% entre un Δt_p variant de 5 minutes à 10 heures

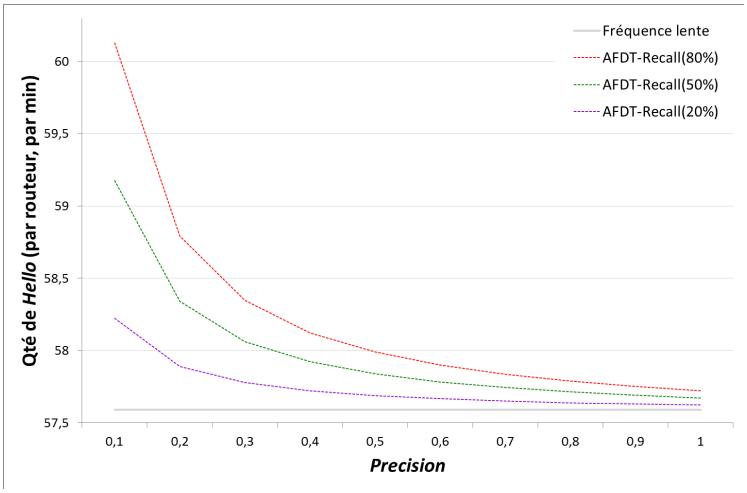


Figure 2.15: Impact de la *Precision* sur le nombre de messages *Hello* avec la topologie européenne.

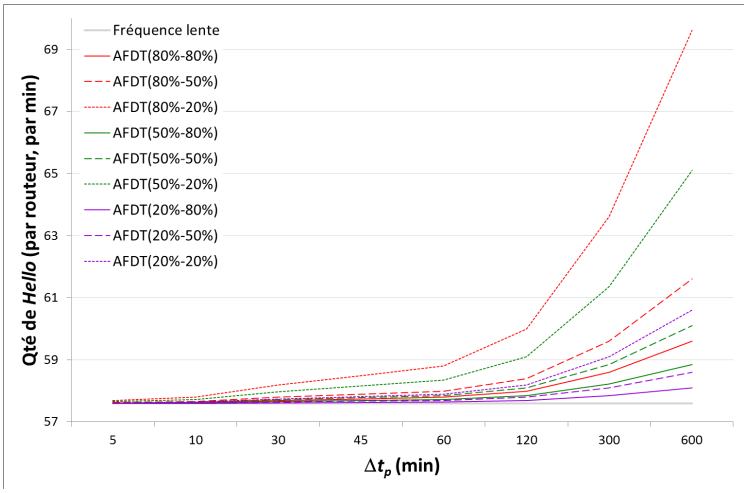


Figure 2.16: Impact de Δt_p sur le nombre de messages *Hello* avec la topologie européenne.

pour le pire des cas. Cette surconsommation qui reste contenue à 3% jusqu'à 2 heures subit une augmentation rapide, notamment dans le cas avec de nombreuse fausses prédictions, *i.e.* avec une *Precision* faible associée à un *Recall* pas trop faible. Il est important de noter que le nombre de messages reçus est ici une moyenne et qu'avec une augmentation de 20%, il est possible que certains nœuds expérimentent de trop nombreuses périodes de forte activité qui pourraient provoquer de l'instabilité au niveau du routage. C'est une des limites de notre modèle qui à été comblée par l'étude du mécanisme *via* des simulation à la Sec. 2.9.2.

2.8.4 Les enseignements de l'étude théorique

Les enseignements de l'application de notre modèle analytique aux trois configurations réseau illustratives soulignent la prépondérance du *Recall* dans les bénéfices qu'apporte le dispositif AFDT. Bien que la probabilité de panne ait une influence sur la disponibilité et la quantité de messages à traiter sur chaque nœud, celle-ci est globalement comparable aux comportements observés sur le comportement standard des protocoles IGP quel que soit la stratégie de configuration du *Hello Interval* utilisée. La seule exception concerne l'influence de la durée de panne sur le nombre de messages *Hello*. Alors que les mécanismes standards affichent une baisse du nombre de messages *Hello* lorsque le MTTR augmente, le mécanisme AFDT souligne une augmentation qui reste néanmoins faible, notamment lorsque le nombre de fausses prédictions reste contenu (*i.e.* avec une *Precision* correcte). Aucune limite n'a pour l'instant été identifiée quant à l'application du mécanisme AFDT par rapport à certaines probabilités de panne. Mais la simulation de notre dispositif dans les mêmes conditions à pour objectif de vérifier cette observation. Le mécanisme AFDT, avec différentes valeurs de *Recall*, de *Precision* et de Δt_p permet, avec un nombre faible de messages *Hello*, de fournir une disponibilité comprise entre les deux stratégies de configuration du *Hello Interval*. Le mécanisme AFDT permet, en fonction du *Recall*, de se rapprocher de la disponibilité assurée avec une fréquence élevée, tout en ne devant traiter qu'un nombre de messages *Hello* 30 fois moins important. Il est notamment intéressant de souligner que le dispositif AFDT avec un *Recall* performant de 80% et une *Precision* d'au moins 50% est un compromis très intéressant entre une fréquence d'envoi de messages *Hello* assurant une stabilité garantie (OPEX) et une très bonne disponibilité.

Les performances observées du mécanisme AFDT plaident pour le remplacement du mécanisme standard de détection de pannes des protocoles IGP par le dispositif AFDT car avec pratiquement une quantité de messages *Hello* reçus très proche de notre configuration stable (*i.e.* avec $t_{HI} = 3s$) et donc ne causant pas d'instabilité, il est possible, grâce à la prédiction de pannes, d'atteindre une disponibilité beaucoup plus compétitive. Il est même possible de s'approcher des performances assurées par des configurations n'étant pas atteignables avec les mécanismes standards des protocoles IGP sans engendrer une forte instabilité.

Le nombre de fausses prédictions et la durée de validité des prédictions Δt_p semble avoir une incidence négligeable vis-à-vis de la quantité moyenne de messages *Hello* reçus par chaque nœud, mais il est nécessaire de s'assurer que les périodes d'accélération du protocole de détection de pannes restent modérées en observant le comportement de l'implémentation de notre mécanisme dans un environnement simulé. De plus, cela permet de vérifier la validité de notre modèle de manière pratique.

2.9 Implémentation

L'implémentation du dispositif AFDT dans un simulateur a deux objectifs. Premièrement, celui de valider le modèle analytique proposé à la Sec. 2.6.1 en reproduisant les conditions étudiées dans la Sec. 2.8. Deuxièmement, de vérifier le comportement du mécanisme dans des conditions extrêmes, c'est-à-dire lorsque le Δt_p est important, ou lorsque les fausses prédictions sont nombreuses (*i.e.* avec une *Precision* faible associée à un *Recall* élevé). En effet, dans ces conditions, la durée d'utilisation du protocole *Hello* en version rapide, ainsi la quantité de messages à traiter par un nœud pendant cette période peut créer une instabilité que les opérateurs veulent à tout prix éviter.

2.9.1 Implémentation du simulateur

La simulation de notre mécanisme a été faite en utilisant la version 3 du simulateur à événements discrets le plus populaire [nsm11]. Nous avons donc utilisé les implémentations du routage et de gestion des nombres aléatoires, à laquelle nous avons ajouté la prise en compte du protocole *Hello* de détection de pannes, ainsi que l'implémentation du mécanisme AFDT d'envoi des messages *Hello* avec une fréquence adaptative.

2.9.1.1 Gestion des événements

Les événements qui rythment une simulation sont les pannes, les prédictions de pannes (bonnes ou mauvaises), ainsi que les actions de re-convergence du protocole de routage.

Les traces des pannes survenant sur les réseaux d'opérateur sont des données sensibles, qu'il est très difficile d'obtenir. En l'absence de telles données, nous avons choisi d'appliquer la théorie générale de la fiabilité [OL09] afin de générer des pannes aléatoires. Le temps entre deux pannes sur un même nœud est supposé suivre une distribution exponentielle dont la moyenne est égale au MTBF souhaité. Cela revient à générer une distribution exponentielle de paramètre $\lambda = 1/MTBF$. De même, on suppose que la durée d'une panne suit une loi log-normale dont la moyenne par nœud est égale au MTTR et l'écart type égale à $0.6 * MTTR$. La distribution généré pour chaque nœud suit donc une loi en $\ln N(\mu, \sigma^2)$ avec $\mu = \log(MTTR) - ((0.5) * \log(1 + ((0.6 * MTTR)^2/MTTR^2)))$ et $\sigma = \sqrt{\log(1 + ((0.6 * MTTR)^2/MTTR^2))}$.

Deux générateurs de distribution (exponentielle et log-normal) sont donc attribués à chaque nœud et d'être utilisés à chaque panne afin de déterminer la durée de celle-ci ainsi que la date de la prochaine. Il n'est donc pas possible de connaître à l'avance le nombre de pannes afin de respecter le *Recall* et la *Precision* de manière exacte. Lors de chaque panne, une variable suivant une distribution uniforme est utilisée afin de déterminer selon la valeur du *Recall* si celle-ci est anticipée ou non par le RAM. De plus une autre variable comprise entre 0 et Δt_p et suivant une distribution uniforme est utilisé afin de déterminer la date d'occurrence de la prédiction.

Un mécanisme différent est utilisé pour générer les fausses prédictions. Dès le début de la simulation, en fonction du MTTR et du *Recall* un estimation du nombre de pannes qui devraient être prédite est effectuée et utilisée pour calculer le nombre de fausses prédictions pour chaque routeur. Il ne reste plus alors qu'à utiliser une distribution uniforme pour chaque routeur afin de distribuer les fausses prédictions de manière uniforme tout au long de la simulation.

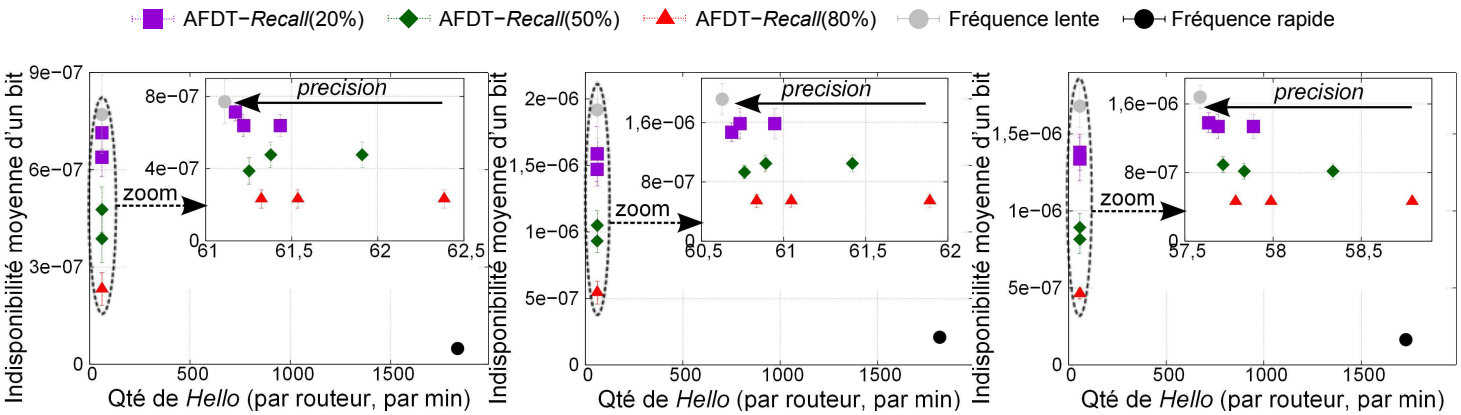
Enfin, afin de se mettre dans les mêmes conditions que notre modèle, le *multi-homing* doit être simulé. Il est donc nécessaire de ne pas inclure l'indisponibilité des flux suite à une panne sur le routeur source $In(f)$ ou le routeur destination $Out(f)$ en arrêtant temporairement les flux concernés pendant la durée de la panne.

2.9.1.2 Scénario des expérimentations

L'utilisation d'un simulateur de niveau paquet tel que NS-3 a pour avantage une précision plus grande au détriment d'une durée de simulation extrêmement longue. Les simulations réalisées utilisent les mêmes configurations et conditions que celles définies pour l'application du modèle analytique à la Sec. 2.8 à ceci près que nous avons dû réduire le débit des flux afin de pouvoir exécuter les simulations dans un temps raisonnable. Elles utilisent donc les topologies et matrices de trafic définies à la Sec. 2.7 dont les débits des flux ainsi que les capacités des liens sont divisés par 102400. Cette opération n'a pas d'influence sur l'observation des résultats de notre mécanisme, mais permet un gain substantiel de temps, indispensable à notre étude. Néanmoins, chaque simulation reste longue, c'est pourquoi nous avons choisi de limiter le temps simulé à $5 * (MTBF + MTTR)$ et d'effectuer seulement sept essais avec une graine différente à chaque fois. Bien qu'un nombre plus important de simulations ainsi qu'une période de simulation plus longue auraient permis d'obtenir des résultats plus précis, les résultats obtenus affichent des intervalles de confiance satisfaisants. L'intervalle de confiance affiché sur chaque graphique est de 99%. Avec les mêmes réglages du protocole *Hello* que dans l'application numérique précédente (*i.e.* $t_{SHI} = 3s$ et $t_{SHI} = 0.1s$), nous avons simulé chaque configuration appliquée au modèle analytique, ce qui permet d'étudier l'impact du MTBF, du MTTR, du *Recall*, de la *Precision* et du Δt_p sur la disponibilité et sur la quantité de messages *Hello* à traiter sur chaque nœud.

2.9.2 Résultats des simulations

2.9.2.1 Analyse conjointe de la disponibilité et de la quantité de messages de contrôle à traiter



(a) A (b) US (c) EU

Figure 2.17: *Ratio* indisponibilité / nombre de message *Hello* avec configuration de référence pour les trois topologies.

La Fig. 2.17 représente l'indisponibilité et la quantité de messages *Hello* reçus par les routeurs dans les conditions de référence décrites à la Sec. 2.8 pour les trois topologies. Les combinaisons des trois valeurs de *Recall* et de *Precision* de 20%, 50% et 80% pour le dispositif AFDT sont comparées à l'utilisation du protocole *Hello* avec une fréquence lente et une fréquence rapide. Cette figure confirme les résultats obtenus lors de l'application du modèle analytique aux intervalles de confiance près. Les enseignements sont donc les mêmes, à savoir, un *ratio* disponibilité / nombre de messages *Hello* (et donc stabilité) meilleur que le mécanisme standard des protocoles IGP existant. Pour une quantité de messages *Hello* similaire à la stratégie stable, le mécanisme AFDT permet d'assurer une disponibilité autrefois uniquement atteignable au détriment de la stabilité du routage.

Néanmoins, ces premiers résultats mettent en évidence un phénomène visible sur l'ensemble des simulations, à savoir des résultats en terme de disponibilité différents en fonction de la *Precision*. Ceci n'est pas dû au mécanisme en lui-même, mais à l'effet de bord de la gestion des nombres aléatoires. Nous reviendrons sur ce phénomène lors de l'analyse de l'impact de la

Precision sur le comportement du dispositif AFDT, mais il est important de noter que cette différence de disponibilité qui reste faible est due à l'utilisation de générateurs de nombres aléatoires supplémentaires afin de répartir correctement les fausses prédictions lorsque celles-ci sont peu nombreuses. L'utilisation de ces variables supplémentaires a pour effet de modifier la distribution du TBF [1], du TTR [2] et des pannes prédites et aboutissant logiquement à une disponibilité différente. Mais nous reviendrons plus en détail sur ce sujet dans la suite de l'analyse.

2.9.2.2 Influence de la probabilité de panne

Pour les mêmes raisons que lors de l'analyse des résultats analytiques, notre analyse n'est illustrée qu'avec une seule topologie, à savoir la topologie européenne. Néanmoins les résultats complets des simulations pour chaque topologie sont disponibles en annexe C.1.

L'intérêt de cette section est de vérifier que le comportement du mécanisme AFDT est conforme à nos résultats analytiques, notamment dans les valeurs extrêmes, c'est-à-dire pour un MTBF faible (exemple : 1000 heures) et un MTTR élevé (exemple : 10 heures). Pour cela, des simulations sont effectuées avec les conditions étudiées par le modèle analytique, c'est-à-dire un MTBF compris dans une plage de 1000 à 10000 heures, ainsi qu'un MTTR prenant des valeurs comprises entre 1 et 10 heures. La Fig. 2.18 confirme le comportement affiché par le modèle

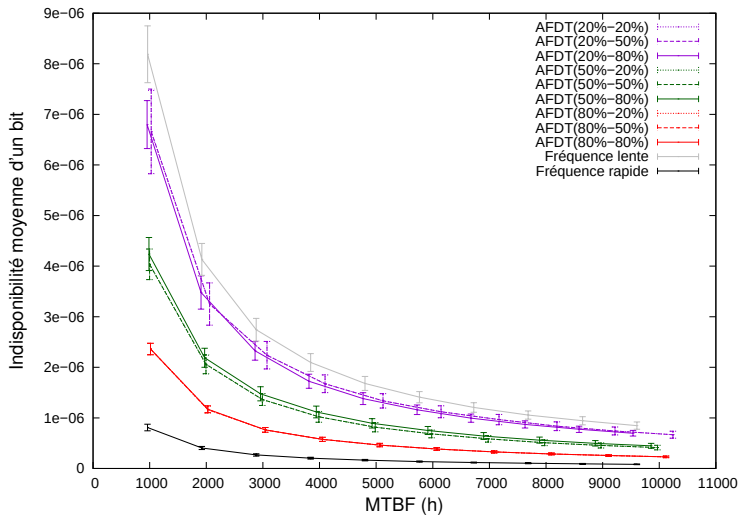


FIGURE 2.18: Impact du MTBF sur la disponibilité avec la topologie européenne.

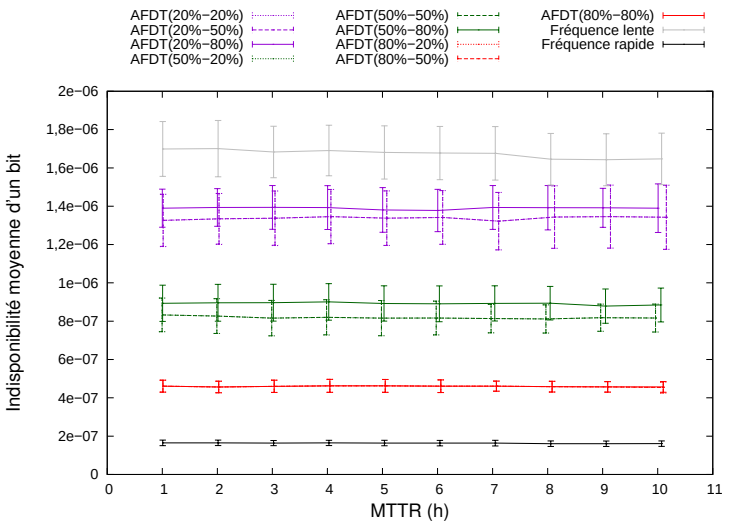


FIGURE 2.19: Impact du MTTR sur la disponibilité avec la topologie européenne.

analytique, même avec une probabilité de panne importante due à un MTBF de 1000 heures. Il en est de même pour le MTTR qui, comme observé grâce au modèle analytique, n'a pas d'influence sur la disponibilité, comme l'illustre la Fig. 2.19. Les résultats de l'incidence du MTBF et du MTTR sur la quantité de messages *Hello* reçus des simulations sont représentés sur les Fig. 2.20 et 2.21. Les valeurs mesurées sont similaires aux résultats analytiques et confirment en partie l'intérêt du mécanisme AFDT même dans une plage de probabilité de panne importante.

Néanmoins, la valeur moyenne du nombre de messages *Hello* ne permet pas de s'assurer complètement du comportement du dispositif AFDT vis-à-vis de la stabilité. En effet, il serait faut s'assurer que la durée d'envoi des messages *Hello* avec une fréquence élevée est faible et que cette fréquence n'est pas trop importante. Pour cela, la Fig. 2.22 représente l'impact du MTBF sur la période de fonctionnement du mécanisme AFDT en mode accéléré. Ce mode reste en grande partie limité à une période inférieure à 0,4% du temps total, ce qui, avec une fréquence moyenne de 643 messages *Hello* à traiter par un routeur en une minute ne devrait pas générer d'instabilité. Cette période peut aller jusqu'à 1% du temps mais uniquement dans le cas d'une *Precision* faible (*i.e.* 20%) avec une probabilité de panne importante ($MTBF < 2000H$). Mais,

1. *Time Between Failure*
2. *Time To Repair*

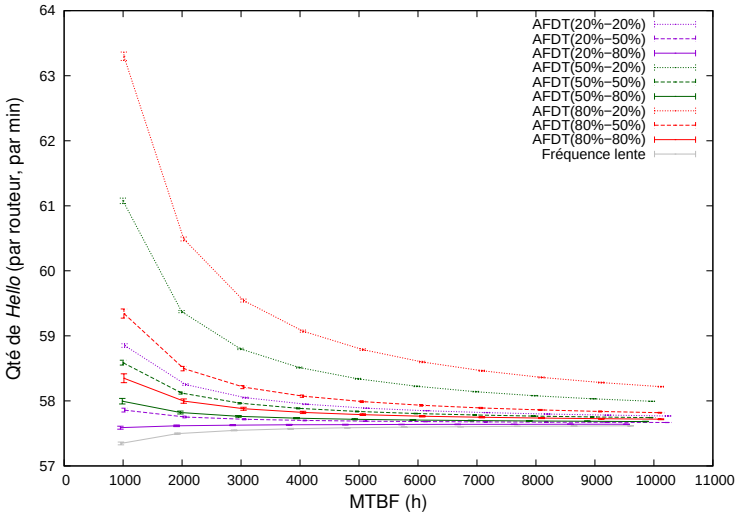


FIGURE 2.20: Impact du MTBF sur le nombre de messages *Hello* avec la topologie européenne.

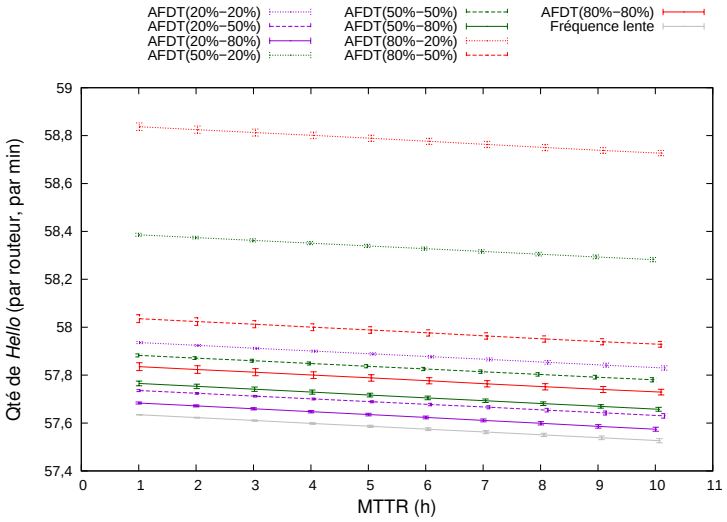


FIGURE 2.21: Impact du MTTR sur le nombre de messages *Hello* avec la topologie européenne.

là encore, même si la stabilité peut être impactée, cela est sans commune mesure avec l'instabilité engendrée par le traitement de 1527 messages *Hello* par minute en continue.

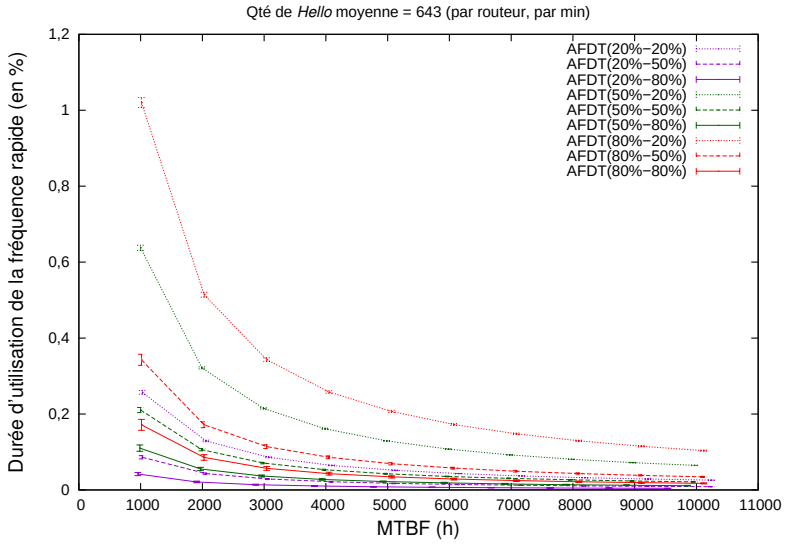


FIGURE 2.22: Impact du MTBF sur la durée de réception des messages *Hello* à une fréquence élevée pour la topologie européenne.

2.9.2.3 Les conséquences de la prédiction de pannes

Après avoir eu la confirmation que la probabilité de panne ne déréglait pas la stabilité du mécanisme AFDT, il est important de valider les caractéristiques de la prédiction de pannes compatible avec le dispositif AFDT. Pour cela, nous simulons la même configuration que celle étudiée avec le modèle analytique. En commençant par le *Recall*, on peut noter que le cas $Recall = 0$ n'est pas traité. En effet, dans ce cas de figure, le nombre de fausses prédictions à générer pour respecter la *Precision* définie par l'Eq. (1.2) n'as plus aucun sens. Mis à part cette spécificité, la Fig. 2.23 montre des résultats de disponibilité identiques à la Fig. 2.13 aux intervalles de confiance prés (99% dans ce cas).

La quantité de messages *Hello* affichée sur la Fig. 2.24 montrent des résultats conformes aux résultats analytiques avec de très bons intervalles de confiance qui confirment la justesse de notre modèle. De plus la Fig. 2.25 met en évidence l'impact quasi-nul du *Recall* sur la période de stress du protocole de détection de pannes. Avec une fréquence de 643 messages par minute

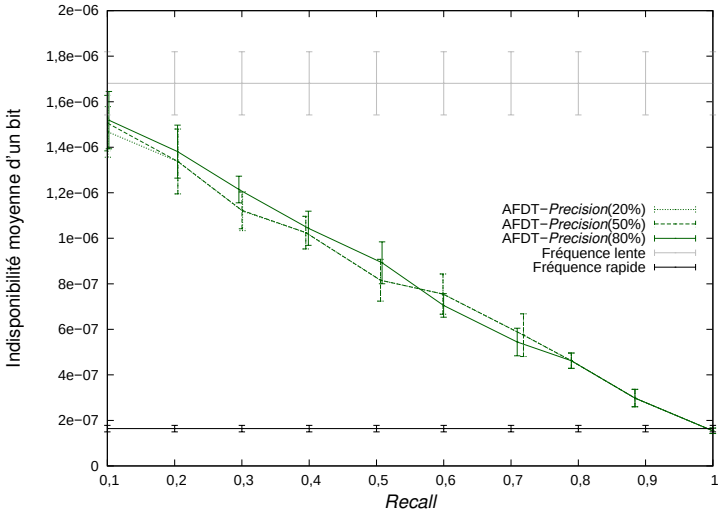


FIGURE 2.23: Impact du *Recall* sur la disponibilité avec la topologie européenne.

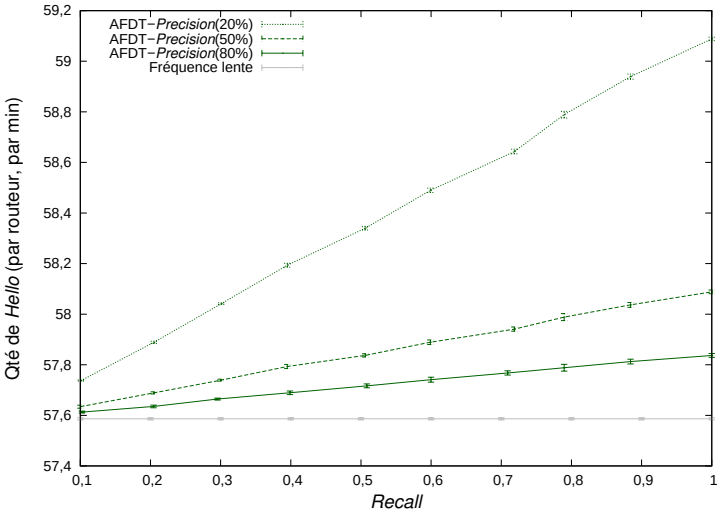


FIGURE 2.24: Impact du *Recall* sur le nombre de messages *Hello* avec la topologie européenne.

pendant une période inférieure à 0,1% du temps et un *Recall* d'au moins 50%, (et ceci est aussi valable pour une période de 0,25% avec un *Recall* de seulement 20%), la stabilité du routage du mécanisme AFDT doit être similaire à celle constatée avec une fréquence lente.

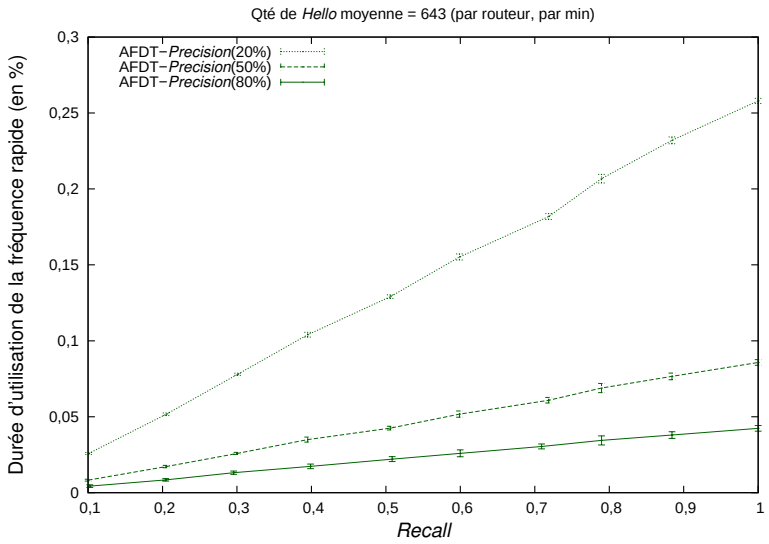


FIGURE 2.25: Impact du *Recall* sur la durée de réception des messages *Hello* à une fréquence élevée pour la topologie européenne.

La Fig. 2.26 devait permettre de constater que la *Precision* ne joue aucun rôle dans l'indisponibilité, car les fausses prédictions n'ont d'incidence que sur la fréquence d'envoi des messages *Hello* qui n'est pas lié à l'indisponibilité. Néanmoins, la Fig. 2.26 met en évidence le comportement relatif aux nombres aléatoires que nous avons révélé précédemment. On peut voir distinctement que les résultats d'indisponibilité ne sont plus stables dans certain cas de *Precision* importante. Pour être plus précis, on peut noter que ce phénomène n'apparaît pas aux mêmes valeurs de *Precision* suivant le *Recall*. Il est même non présent dans le cas d'un *Recall* de 80%. La raison est que ce phénomène ne se déclenche que lorsque le nombre de fausses prédictions devient très faible. Dans le cas normal, les simulations utilisent un même nombre de variables générant des nombres aléatoires ce qui implique pour une probabilité de panne identique et une même graine, des simulations subissant exactement les mêmes pannes. Il est donc logique d'observer des résultats de disponibilité similaires lorsque le paramètre n'influence pas le mécanisme de convergence comme c'est le cas sur la Fig. 2.29. Mais dans le cas où le nombre de fausses

prédictions est faible, il est nécessaire de répartir correctement les quelques fausses prédictions de manière aléatoire et uniforme. Pour cela, notre implémentation créé dynamiquement de nouvelles variables suivant des distributions uniformes le cas échéant. Le résultat de cette opération crée un décalage dans la gestion des nombres aléatoires d'une simulation avec une même graine, responsable d'un déroulement différent de la simulation notamment vis-à-vis des pannes, mais aussi de la prédiction de pannes. Ce comportement n'enlève en rien la validité de notre implémentation, mais il est nécessaire de comprendre son origine, afin de ne pas faire de mauvaise interprétation des résultats obtenus. Compte tenu de cette information, la Fig. 2.26 confirme l'indépendance entre la disponibilité et la *Precision* utilisée par le dispositif AFDT. L'évolution maîtrisée de la quantité moyenne de messages *Hello* pour des valeurs de configuration générant de nombreuses prédictions, mis en valeur par la Fig. 2.27, suit en tout point le comportement mis en évidence avec notre modèle analytique.

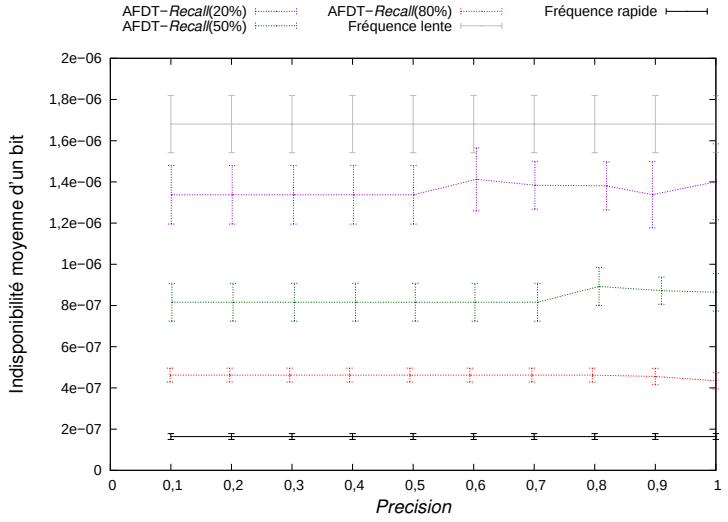


Figure 2.26: Impact de la *Precision* sur la disponibilité avec la topologie européenne.

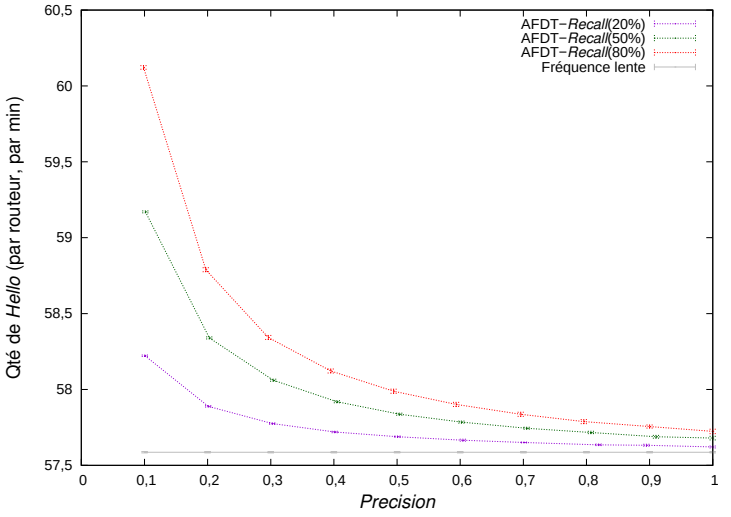


Figure 2.27: Impact de la *Precision* sur le nombre de messages *Hello* avec la topologie européenne.

Il est néanmoins nécessaire de s'attarder sur le comportement du mécanisme AFDT en période de stress, en fonction de la *Precision*. Avec une fréquence de 642 messages par minute, soit presque trois fois moins qu'avec un protocole IGP et un *Hello Interval* de 100 millisecondes, la Fig. 2.28 montre une augmentation significative de la durée de stress lorsque la *Precision* passe sous les 30%, et ce de manière encore plus visible lorsque le *Recall* est important. On peut considérer qu'au-dessus d'une *Precision* de 30% la stabilité est garantie avec une durée de fonctionnement du mécanisme AFDT en mode stress inférieur à 0,1%. Avec une *Precision* inférieure, la durée d'utilisation de la fréquence rapide monte jusqu'à 0,45% ce qui ne devrait pas créer d'instabilité importante. Il ne reste plus qu'à vérifier que la durée de prédiction reste un paramètre sans grande incidence sur le fonctionnement du mécanisme proposé. La Fig. 2.29 montre une disponibilité constante quelle que soit la durée de la prédiction, ce qui conforte le choix de ne pas inclure ce paramètre dans le modèle de calcul de la disponibilité.

La quantité de messages *Hello* reçus illustrée par la Fig. 2.30 est conforme aux observations de notre application numérique. Elle souligne une augmentation plus importante que sur les autres figures qui mérite d'être analysée de manière plus précise en isolant la période d'utilisation de la fréquence élevée.

La Fig. 2.31 montre que le dispositif AFDT contient la durée d'utilisation du *Hello Interval* de 100 millisecondes sous les 0,5% jusqu'à un Δt_p de 2 heures mais qu'ensuite celle-ci devient beaucoup plus importante. Cette durée grimpe jusqu'à 2% du temps, ce qui, même si elle reste loin des 100% de traitement de 1527 messages par minute, est susceptible de générer une instabilité que certains opérateurs exigeant ne sont pas prêts à tolérer. Cette simulation permet ainsi de valider les hypothèses de notre modèle analytique, et d'assurer que le mécanisme AFDT reste efficace pour les plages de paramètres testées.

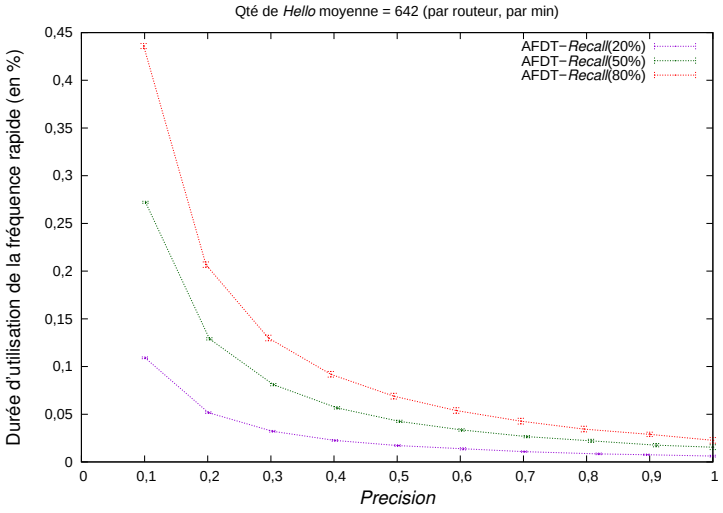


FIGURE 2.28: Impact de la *Precision* sur la durée de réception des messages *Hello* à une fréquence élevée pour la topologie européenne.

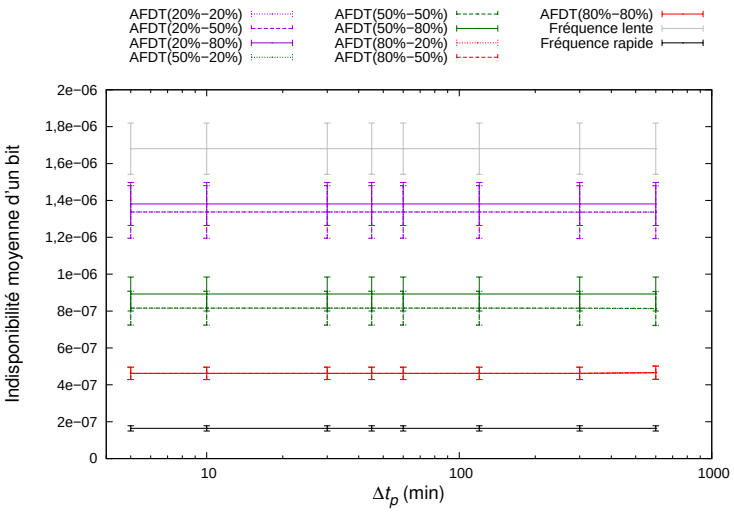


FIGURE 2.29: Impact de Δt_p sur la disponibilité avec la topologie européenne.

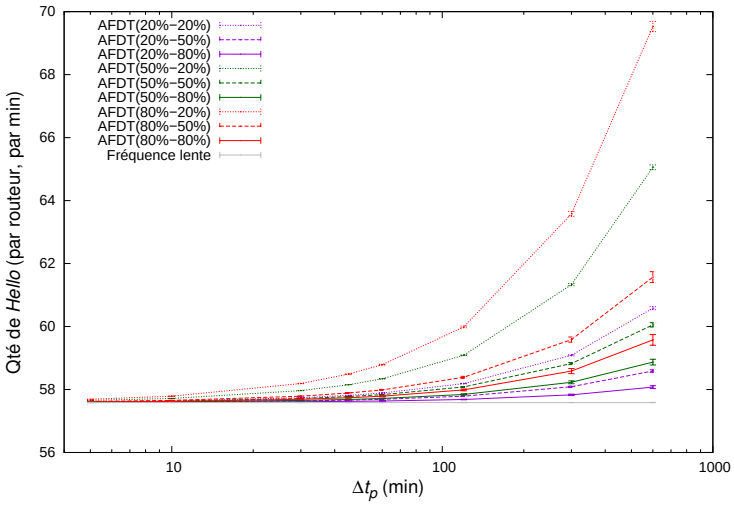


FIGURE 2.30: Impact de Δt_p sur le nombre de messages *Hello* avec la topologie européenne.

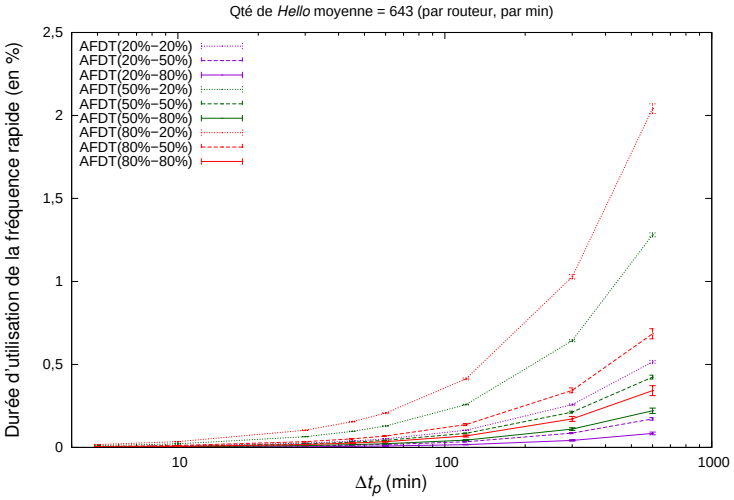


FIGURE 2.31: Impact de Δt_p sur la durée de réception des messages *Hello* à une fréquence élevée pour la topologie européenne.

2.9.2.4 Les enseignements apportés par la simulation

L'utilisation du simulateur NS-3 permet une certaine confiance dans l'implémentation des mécanismes utilisés pendant cette simulation. La simulation a fourni des résultats avec un intervalle de confiance de 99% permettant une analyse précise.

De manière générale, les résultats de simulations attestent de l'exactitude du modèle analytique. L'interprétation de ces résultats confirme l'intérêt du dispositif AFDT pour fournir des performances qui n'étaient auparavant atteignables qu'au détriment de la stabilité. Le mécanisme proposé permet, avec une stabilité égale à l'utilisation d'un *Hello Interval* de trois secondes, d'assurer des performances de disponibilité jusqu'à quatre fois plus importantes avec un *Recall* de 80%.

De plus, la simulation permet de vérifier que la période de stress du mécanisme AFDT est bien contenu dans le temps et dans l'espace afin de garantir une stabilité même pour des valeurs extrêmes de probabilité de panne et de prédiction de pannes. Fort de cette dernière vérification, l'enseignement à retenir est le lien fort entre performance de disponibilité et *Recall*, validant l'importance du *Recall* comme premier critère dans le choix d'implémentation de la fonction de

prédiction de pannes, mais en prenant bien soin d'utiliser si possible un Δt_p inférieur à 2 heures et une *Precision* d'au moins 50%.

2.10 Conclusion

Le dispositif d'adaptation de la fréquence d'envoi des messages de détection de pannes au risque de pannes (AFDT) proposé dans ce chapitre a pour but d'améliorer le temps de convergence du protocole IGP. La détection de panne est l'étape la plus pénalisante lors de la restauration IP. Elle peut être accélérée en augmentant la fréquence d'envoi des messages au dépend de la stabilité du réseau. L'observation de l'état de santé des équipements permet d'utiliser l'information de risque de panne en temps réelle pour n'accélérer cette fréquence que pendant un temps limiter et un sous-ensemble d'équipement restreint. Le comportement de ce mécanisme a été modélisé de manière analytique afin de quantifier ses conséquences sur l'indisponibilité ainsi que sur la surcharge de travail des routeurs responsables de l'instabilité. Les résultats de cette étude ont ensuite été confrontés au comportement de l'implémentation du mécanisme au sein d'un simulateur à événements discrets afin de vérifier que celui-ci était conforme à nos prévisions. Les résultats permettent de constater que le mécanisme permet d'améliorer la disponibilité avec peu d'inconvénients. Les gains sur la disponibilité sont proportionnels à la proportion des pannes qui ont été anticipées par le module de prédiction de pannes du RAM. Les incidences sur la stabilité du routage restent mineures quelle que soit la topologie de réseau car l'accélération de la fréquence d'envoi des messages ne concerne que les équipements risqués. Néanmoins, des conditions extrêmes, regroupant à la fois une faible *Precision* et un *Recall* important (*i.e.* un nombre de prédictions important), ainsi qu'un taux de pannes élevé ou une durée de prédiction très longue (*i.e.* Δt_p) montrent des limites qui peuvent dissuader certains opérateurs d'utiliser le dispositif dans de telles conditions. L'intérêt de ce premier mécanisme proactif est qu'il exploite une information de risque de panne pour régler de manière autonome un paramètre permettant d'améliorer la disponibilité du réseau mais qui est actuellement utilisé de manière statique à cause de la gestion des réseaux par des opérateurs humains. Les conséquences de cette intervention ne modifie pas le flux de trafic ce qui permet de limiter l'incidence des fausse prédictions sur la qualité de service. Cependant, une interruption de service résiduelle persiste. La suppression de cette interruption requière une intervention plus lourde telle que celle proposée dans le chapitre suivant.

Chapitre 3

Proposition pour un routage sensible au risque de pannes

Sommaire

3.1 Introduction

La demande pour des services fiables oblige les opérateurs de réseaux à fournir une qualité de service et une disponibilité toujours plus importante pour rester compétitif. Puisque la disponibilité n'est pas une option, ils doivent faire face aux incidents en utilisant des mécanismes tels que la restauration ou la protection. La protection permet de rétablir un chemin de manière extrêmement rapide, mais est très coûteuse en terme de ressources, alors que la restauration est beaucoup moins chère, mais génère une plus longue interruption de service. De plus, même si les performances sont différentes, tous ces mécanismes sont réactifs et donc ne permettent pas de masquer complètement les effets d'une panne aux utilisateurs finaux. Ce chapitre, s'intéresse au mécanisme de restauration IP fournit par les protocoles IGP tel qu'OSPF et IS-IS. Le délai entre l'occurrence d'une panne et la fin de la convergence d'OSPF implique des pertes de paquets, des boucles de routage temporaires ou des trous noirs de routage qui perturbent la topologie de routage de l'opérateur. En effet, le mécanisme de restauration n'intervient qu'après l'occurrence de la panne ce qui ne permet que de limiter l'impact d'une panne sur le trafic, mais pas de la supprimer.

Nous proposons une approche complémentaire qui élimine ce défaut en anticipant les pannes grâce au module RAM d'évaluation du risque de pannes afin d'outrepasser la lenteur des mécanismes de gestion des pannes. Cela permet de supprimer totalement l'effet d'une panne sur le trafic en effectuant des actions de reconfiguration quelques secondes avant la panne. L'utilisation de la prédiction de pannes permet d'ajuster dynamiquement les métriques de routage afin d'obliger le trafic à ne pas emprunter les éléments de réseau avec un risque de panne identifié et donc d'isoler les pannes avant qu'elles n'occasionnent des dégâts sur le trafic [RNP+11, VC10, CPP+10, KAB+11].

Le chapitre commence tout d'abord par décrire le contexte, ainsi que les travaux connexes, pour ensuite détailler le principe de la contribution. La deuxième partie de ce chapitre est dédiée à l'évaluation du dispositif. Dans un premier temps, un modèle analytique est proposé pour quantifier l'apport du dispositif RAR. Puis, l'implémentation du mécanisme dans un simulateur mécanisme permet d'observer son comportement de manière plus fine. Enfin, un prototype expérimental a été conçu afin de vérifier la faisabilité du concept dans un environnement réseau réel.

3.2 La restauration IP

Le contexte de ce chapitre est similaire au chapitre précédent. Dans les réseaux IP, le routage intra-domaine est effectué à l'aide des protocoles IGP tel qu'OSPF et IS-IS. Avec OSPF, chaque routeur d'une aire de routage découvre et construit une vue complète de la topologie du réseau. Chaque routeur OSPF doit signaler sa connectivité par l'intermédiaire de LSA [1]. Afin de réduire le nombre de messages échangés, plusieurs LSA sont regroupés au sein d'un même paquet. Ces LSA sont diffusés aux autres routeurs du réseau, afin qu'ils puissent construire une vue complète de la topologie. L'ensemble des LSA qui forment le graphe de la topologie est stocké dans une LSDB [2]. Basé sur cette topologie, chaque routeur calcule un arbre de plus court chemin dont il est la racine en utilisant un algorithme à état de liens tel que l'algorithme de Dijkstra. Il applique ensuite le résultat pour construire sa table de commutation (FIB [3]). Lors d'un changement de topologie, typiquement une panne, le processus de convergence est déclenché. Ce processus décrit à la Sec. 2.2.1 nécessite une période t_C pour rétablir les flux de trafic impactés par une panne. Cette convergence se fait en quatre étapes [PIM11] :

- l'étape de détection de panne avec t_D le temps de cette détection ;
- l'étape de diffusion des LSA d'une durée t_F ;
- le calcul des plus courts chemins dont la durée est notée t_{SP} ;

1. *Link State Advertisements*
2. *Link State DataBase*
3. *Forwarding Information Base*

– et l'écriture du nouveau routage dans les tables de routage et de commutation qui dure t_U secondes.

En conséquence, on obtient une durée de convergence $t_C = T_D + T_F + T_{SP} + T_U$ (voir Eq. (3.2)) durant laquelle la topologie de routage n'est pas cohérente. En l'état actuel, la durée d'indisponibilité des flux touchés par une panne due au processus de convergence est de l'ordre de plusieurs secondes, ce qui n'est pas acceptable pour les trafics *premium* tel que la VoIP, la vidéo, les services de télétravail interactifs, etc.

3.3 Problématique

La gestion des pannes dans les réseaux IP est effectuée au travers du processus de restauration IP. Une reconvergence complète du réseau est déclenchée suite à la détection d'une panne afin de mettre à jour les routeurs avec des chemins valides. Malheureusement, lors d'une panne dans le réseau, le protocole IGP prend du temps pour détecter la panne et rétablir une vue cohérente de la nouvelle topologie du réseau. Durant cette transition, le trafic transféré vers les équipements en panne sera perdu. De plus, des boucles de routage peuvent apparaître et engendrer de la congestion. Ces pertes engendrent une interruption de service de plusieurs secondes, ce qui n'est pas acceptable pour le trafic sensible à la QoS (VoIP, Vidéo, etc.). Des techniques ont été proposées afin de réduire le délai de convergence (Cf Sec. 3.4), notamment pour accélérer l'étape de détection de panne traitée au Chap. 2. Néanmoins, le délai de convergence reste impactant pour les trafics nécessitant une QoS garantie.

La nature réactive du dispositif de restauration permet d'adapter le routage du réseau à une panne de manière optimale, contrairement aux mécanismes de protection, comme peut le proposer le protocole GMPLS (voir Sec. 4.2.3.1). La simplicité d'utilisation et la nature relativement autonome du protocole IP sans connexion [RNP+11] en font un candidat crédible par rapport à GMPLS dans bien des cas. Il est donc important d'améliorer ses défauts, notamment sa gestion des pannes dont la nature réactive oblige à conserver une interruption de service pendant le processus de reconvergence, ce qui pénalise le trafic des utilisateurs.

3.4 Amélioration du délai de restauration IP

Les travaux précédents se sont focalisés sur la réduction du temps d'exécution de ces étapes de convergence [FFEB05, RMD05]. Par exemple, plusieurs études se sont intéressées au problème de la détection de pannes étudiée au Chap. 2 [GRcF03]. La configuration du protocole de détection de pannes *Hello* avec une fréquence élevée amène une instabilité dans le routage et une fréquence lente augmente l'interruption de service. Ce problème de stabilité oblige les opérateurs à utiliser une période d'envoi des messages *Hello* supérieure à une seconde au dépend d'un rétablissement rapide du service.

Les autres étapes de la convergence ont aussi été l'objet d'amélioration notamment l'étape de calcul de route, et de nombreuse techniques de *Fast ReRoute* [SB10] ont été proposées afin d'outrepasser les délais $t_F + t_{SP} + t_U$ en pré-calculant à l'avance certains chemins de secours. L'étape de diffusion des LSA, bien que négligeable dans le processus de convergence a été l'objet d'amélioration [PE05, RBGK03, SWL+03, Nar00], notamment afin d'optimiser et de restreindre sa diffusion. Cependant c'est l'étape de calcul des plus courts chemins qui a été la plus étudiée [NST01, XN98, ZXW07, JXLP09]. Cette étape, la deuxième la plus longue après la détection de panne, nécessite une puissance de calcul importante, et est d'autant plus importante que la topologie est grande. Plusieurs travaux proposent une amélioration avec des algorithmes de calcul de route incrémental [NST01, FMSN94] ne nécessitant que de recalculer la portion affectée par la panne afin de réduire le temps de calcul des plus courts chemins. Un calcul parallèle dynamique des plus courts chemins est proposé dans [ZXW07] pour affecter le calcul partiel des plus courts chemins au routeur concerné sur différents processeurs. Xiao et al. [XN98] propose de diviser la topologie du réseau en aires indépendantes pour des processeurs différents selon la LSDB de chaque nœuds, afin d'accélérer le calcul de route dans des regroupements de routeurs. Enfin, l'utilisation de routeurs distribués exécutant plusieurs instances de calcul des plus courts chemins

est proposé dans [JXLP09] afin d'accélérer l'étape de calcul. Néanmoins, ces améliorations, bien que permettant de réduire le temps de rétablissement d'un chemin, ne permettent pas de rendre une panne imperceptible par le trafic, laissant le champ libre des approches différentes.

L'approche complémentaire qui a suscité le plus d'engouement ces dernières années est celle du *IP Fast ReRoute* (IP FRR) [FB05] où les routeurs, au plus près de la panne, utilisent des routes pré-calculées afin de rétablir les flux de trafic rapidement, sans en informer tout le réseau et attendre la reconvergence totale du protocole IGP. IP FRR [1] fait référence à un ensemble de propositions dont le but est de fournir un reroutage rapide en n'utilisant seulement les technologies de base du routage IP. Comme étudié par la suite dans le Chap. 4, des initiatives similaires ont également été proposées pour la technologie MPLS sous le nom de MPLS FRR [2]. L'idée à l'origine de IP FRR est que, lorsqu'une panne apparaît, les chemins des routeurs de la zone autour de la panne sont fortement perturbés mais les routeurs plus éloignés possède des routes toujours valides. Il est donc nécessaire de mettre en place des mécanismes permettant de les utiliser afin de continuer à transférer les paquets jusqu'à leur destination en attendant la mise à jour complète des tables de routage du réseau. Cela permet d'outrepasser les délais $t_F + t_{SP} + t_U$ en fournissant un chemin temporaire utilisant les tables de routage disponibles (*i.e.* obsolète depuis la panne), en attendant les nouveaux chemins définitifs qui seront déployés à la fin du processus de convergence.

Une des premières techniques proposée pour rétablir rapidement un chemin est l'utilisation de plusieurs chemins avec des coûts identiques (ECMP [3]). Dans le cas d'une panne, il reste toujours l'autre plus court chemin de même poids afin de continuer à transférer le trafic. Une alternative est l'utilisation de *Loop Free Alternate* (LFA) [AZ08] où le routeur directement lié à l'équipement en panne, transfert le trafic vers un de ses voisins directs qui possède un chemin sans panne pour la destination du trafic. Lorsqu'un chemin LFA [4] n'est pas disponible sur les voisin direct, le routeur recherche un routeur dans un périmètre élargi à plusieurs sauts si celui-ci possède un chemin sans panne. Ce mécanisme appelé réparation multi-saut [SB10] est décliné suivant plusieurs techniques. L'utilisation d'une adresse spéciale mentionnant le routeur à éviter est appelée *Not-via address* [SBP11, ESRC09]. Il est aussi possible d'utiliser la technique du *U turn alternate* [Atl06] qui utilise un voisin considérant ce routeur en question comme prochain saut principal pour la destination du paquet et qui possède également un LFA pour la destination de ce paquet, qui ne passe pas ce routeur. Une autre proposition de IP FRR est la possibilité pour un routeur d'inférer une possible panne sur un lien en recevant un paquet sur une interface inhabituelle [NLY+07, WN07] et de transférer le paquet vers un chemin évitant ce lien possiblement en panne. Le papier « *Relaxed multiple routing configurations for IP fast reroute* »[CHK+08] propose l'utilisation par les routeurs de plusieurs configuration de routage, en maintenant une FIB pour chaque cas de panne. Lorsqu'un routeur détecte une panne, il marque le paquet qui aurait autre fois été transféré vers le routeur en panne, pour que chaque routeur puisse utiliser la configuration de routage correspondant à la panne sur le routeur en question. Enfin Xi et Chao [BFPS07] propose l'utilisation de tunnel IP (*IP-in-IP* ou GRE [5]) pour joindre un routeur possédant un chemin sans panne jusqu'à la destination, pour ensuite utiliser le routage IP classique de ce routeur jusqu'à la destination.

Il est important de noter que les chemins temporaires empruntés grâce au IP FRR ne sont pas définitifs et qu'en parallèle, le processus de convergence s'exécute et viens remplacer les chemins temporaires lorsqu'il est achevé. Néanmoins, les techniques de routage rapides introduisent quand même une interruption de service puisqu'elle met un certain temps à se mettre en place, même si celui-ci est beaucoup plus court que la convergence OSPF. Le temps de détection qui a le plus d'incidence n'est pas traité par cette technique, et les chemins temporaires ne sont pas optimaux et peuvent introduire des boucles de routage et de la congestion dans certain cas.

Une autre approche connexe à notre proposition est l'utilisation du risque de pannes dans le routage. Afin de minimiser l'incidence d'une panne sur le trafic, la prise en compte du risque

1. *IP Fast ReRoute*
2. *MPLS Fast ReRoute*
3. *Equal-cost multi-path routing*
4. *Loop Free Alternate*
5. *Generic Routing Encapsulation*

de pannes dans le routage a été étudiée au travers plusieurs initiatives. Des études ont proposé la prise en compte du risque de pannes dans le routage. Pour commencer, des propositions de routage basées sur la disponibilité du chemin [ZZZ+07, LQL08, MFH08] ont été utilisées afin de s'assurer de respecter un certain SLA. De même le risque de pannes a déjà été utilisé afin de minimiser le nombre de pannes potentielles touchant un chemin [LY07, YVJ05]. Mais le risque de pannes ici utilisé est un risque statique, c'est-à-dire qui ne varie pas dans le temps et qui provient de statistiques de fiabilité à long terme qui ne correspondent en rien au risque de pannes envisagé dans cette thèse par le module RAM. De manière similaire, dans [XTMM10], des statistiques longs termes sont utilisées afin de choisir le meilleur chemin capable de satisfaire une disponibilité précise. Ce chapitre propose une approche assez différente puisque notre risque de pannes est une donnée dynamique calculée en temps réel en observant l'état de santé des éléments de réseau.

3.5 Description de la proposition

Un des avantages majeurs de la restauration IP est son coté dynamique qui permet d'adapter de manière sur-mesure la configuration du réseau à la situation de la panne en cours. Mais le désavantage est le côté réactif. Les systèmes de gestion des pannes, notamment dans les télécommunications, traitent les incidents de manière réactive car c'est la façon la plus simple de régler le problème. Mais une approche réactive par essence implique une réaction après l'occurrence de la panne et ne permet donc pas de supprimer toutes les conséquences d'une panne sur le trafic. En effet, la restauration IP amène à des pertes de paquets tant que le processus de convergence d'OSPF n'est pas achevé. Les dispositifs d'IP FRR permettent de réduire ce délai mais pas de le supprimer et peuvent induire des boucles de routage et des topologies sous-optimales.

Nous avons exploré une nouvelle approche complémentaire, en ajoutant un mécanisme préventif aux mécanismes de résilience actuels. Cette approche proactive évalue le risque de panne en temps réel afin de créer une fenêtre de temps dans laquelle des actions préventives peuvent être prises. En l'occurrence, il s'agit d'adapter le comportement du routage afin d'éviter l'impact néfaste de la panne imminente en forçant le trafic à contourner la future panne. Si l'on considère que les réseaux des opérateurs sont dimensionnés et configurés pour supporter au moins une panne de façon transparente pour le trafic [FT02, NSB+03, NBTD07, RCT+11] une telle approche proactive semble tout à fait envisageable. La stratégie proposée est le changement temporaire de la métrique des liens ayant un fort risque de panne avec une valeur dissuadante pour le trafic (*i.e.* bien plus grande que les autres métriques).

3.5.1 Aperçu général du principe de re-routage proactif

En collectant une information de risque de panne temps réel du RAM, il est possible pour un agent autonome tel que le RM_DE, de prendre en compte ce risque dans le routage. Nous proposons donc au travers d'un module de *Risk-Aware Routing* (RAR) au sein des RM_DE (voir Fig. 3.1) d'utiliser le temps précieux précédant une panne pour préventivement obliger le trafic à éviter les éléments de réseaux risqués. Cela permet d'atténuer les effets d'une panne sur le trafic de l'utilisateur. Dans les réseaux avec un protocole IGP tel que IS-IS ou OSPF, cela se traduit par l'ajustement des métriques des liens de manière à dissuader les flux d'utiliser les liens risqués [RNP+11, VC10, CPP+10, KAB+11] (*i.e.* avec une valeur de métriques très grande et égale à $RiskyMetric(link_i)$ définit à la Sec. 3.5.2).

Les avantages de cette méthode sont multiples. Premièrement, si le lien fait partie du seul chemin vers une destination, il est toujours disponible. Cela permet notamment de limiter l'incidence des fausses prédictions par rapport à une stratégie ou le lien serait désactivé complètement.

Lorsqu'une prédiction de panne se vérifie, le trafic à préventivement été envoyé sur un autre chemin, la panne ne perturbe donc pas le trafic (sauf pour les rares cas où le lien risqué faisait partie du seul chemin disponible, mais il est alors impossible de faire quelque chose dans ce cas extrême). Dans ce cas, le dispositif RAR permet de supprimer complètement l'interruption de service et ainsi d'assurer une meilleure QoS pour l'utilisateur.

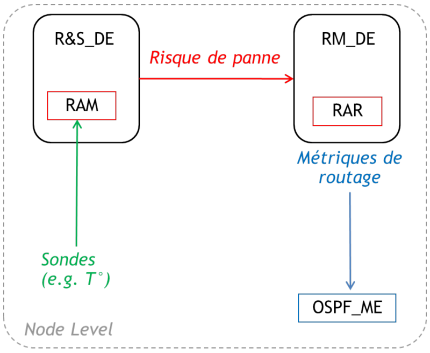


FIGURE 3.1: Architecture fonctionnelle du mécanisme RAR.

Si la panne n'est pas anticipée, la panne est gérée par le processus de convergence du protocole IGP ; l'indisponibilité est donc celle de la restauration IP. S'il y a une fausse prédiction de panne, le trafic est inutilement rerouté. Mais les réseaux sont actuellement configurés pour supporter un tel reroutage, notamment dans le cas d'une panne [FT02, NSB+03, NBTD07, RCT+11], les conséquences sont donc minimes sauf dans le cas d'un trop grand nombres de faux positifs. C'est pourquoi la Sec. 3.9.2 s'attache à évaluer l'impact des faux positifs, notamment sur la congestion que peuvent impliquer les nombreux reroutages. De plus, alors qu'un reroutage avant une panne ne fait qu'anticiper la situation du routage après la panne, un reroutage suite à une fausse prédiction créé deux changements de routage inutile. Or (Cf Chap. 2) les opérateurs sont très attachés à maintenir la stabilité dans leur routage. Avec un nombre de faux positifs raisonnable, cela ne devrait pas modifier sensiblement la donne néanmoins il convient d'étudier l'incidence des fausses prédictions sur la stabilité du routage pour en être certain (Cf Sec. 3.8 et 3.9.2).

Un des avantages importants du mécanisme RAR est qu'aucun processus de standardisation n'est nécessaire et que la compatibilité avec les protocoles actuellement déployés est totale. Il suffit de modifier localement sur le bon routeur le poids d'un lien et d'exploiter les capacités de diffusion de la topologie de routage d'OSPF (et d'IS-IS) pour propager le nouveau schéma de routage augmenté de la prise en compte du risque à tout le réseau et ses flux.

Ensuite, il est essentiel de ne pas bousculer les pratiques actuelles utilisées par les opérateurs en configurant les métriques des liens comme précédemment, afin d'optimiser la bande passante, le délai, l'équilibrage de charge, etc. L'intervention préventive reste restreinte à une courte période de temps pendant laquelle un important risque de panne est détecté, et ne concerne que le poids des liens risqués.

Bien qu'il soit possible de choisir un ensemble plus complexe de niveaux de risque afin d'utiliser ce risque directement comme métrique, cela aurait limité la stratégie d'ingénierie de l'opérateur à n'optimiser que l'exposition au risque. Au contraire, notre approche composite permet de combiner des métriques définies par l'opérateur la plupart du temps et les métriques utilisant le risque de panne seulement lors de l'anticipation d'une panne. Cette méthode consiste à garder un poids faible pour les liens sûrs et à augmenter de manière significative le poids des liens risqués afin d'orienter le trafic loin d'eux, vers des chemins de coût faible. Dans ce but, le coût des liens risqués doit être assigné avec plusieurs contraintes.

3.5.2 Considérations protocolaires

Premièrement, les protocoles IGP ont des limitations intrinsèques et il est nécessaire de les prendre en compte. Dans le cas d'OSPF, les 16 bits définissant la métrique ne permettent pas d'utiliser une valeur positive supérieur à $2^{16} - 2$, car la valeur maximum est réservée pour la représentation de l'infini. Soit $MaxPossibleCost$ cette valeur.

Deuxièmement, une métrique de lien risqué doit être suffisamment grande pour obliger tous les flux de trafic à préférer n'importe quel chemin sûr à la place d'un lien risqué. Cette contrainte implique d'avoir une valeur de métrique de risque toujours plus grande que le plus court chemin entre les deux extrémités du lien (à l'exception du lien risqué). Mais dans le cas où seul des chemins risqués seraient possibles, il est nécessaire de pouvoir privilégier le chemin avec un nombre

minimum de liens risqués. En d'autres termes, les différentes métriques de liens risqués doivent être le plus proche possible afin de maximiser le nombre de détections de risques simultanés possibles. Soit $\Delta CostR$ la différence maximum entre deux métriques risquées et $Min(CostR)$ la valeur de la plus petite métrique risquée, le nombre maximum de prédictions de panne simultanées possibles tout en conservant la relation d'ordre est $Min(CostR)/\Delta CostR$.

Le résultat de ces contraintes est que la base commune pour tous les liens doit être supérieure au plus long chemin initial du réseau sans boucle. Mais puisque l'accès à cette donnée est un problème difficile, il est préférable d'utiliser $MaxPossibleCost$ pour deux raisons. Premièrement, c'est la valeur autorisée ayant la plus forte probabilité d'être supérieure au plus long chemin. Deuxièmement, cette valeur ne cause pas vraiment de limitation en terme de prédictions simultanées. En effet, le champ *intra-area* supporte jusqu'à 28 métriques risqués ce qui est bien assez.

La quatrième contrainte concerne l'arbitrage entre deux chemins égaux contenant le même nombre de liens risqués. Avec l'utilisation d'une même valeur pour tous les liens risqués, il est impossible de choisir entre ces deux chemins. Mais puisque la métrique initiale reflète le choix de l'opérateur, il est préférable d'ajouter la valeur de la métrique initiale à la base commune afin de rendre l'arbitrage possible. En conséquence, la métrique d'un lien risqué i ($RiskyMetric(link_i)$) correspond à $MaxPossibleCost - MaxInitialCost + InitialCost(link_i)$ avec $MaxInitialCost$ la métrique la plus grande dans la configuration initiale du réseau et $InitialCost(link_i)$ la valeur de la métrique du lien i configurée par l'opérateur.

Une fois que la métrique de risque a été calculée, le processus autonome en charge de modifier la métrique des liens OSPF de la configuration initiale vers la valeur de risque doit s'assurer de le faire d'une manière douce et itérative telle que décrite dans [FB07] afin d'éviter les boucles de routage pendant le processus de convergence.

3.5.3 Illustration par l'exemple

L'exemple décrit dans cette section permet de visualiser le comportement du mécanisme RAR de manière illustrative. Pour cette exemple, on considère la configuration définie par la Fig. 3.2 comme étant la situation stationnaire. Le réseau est composé et huit routeurs utilisant un protocole de routage de type IGP tel qu'OSPF. Pour notre exemple, ce réseau transporte trois flux de trafic, en utilisant les métriques de routage affichées à la Fig. 3.2.

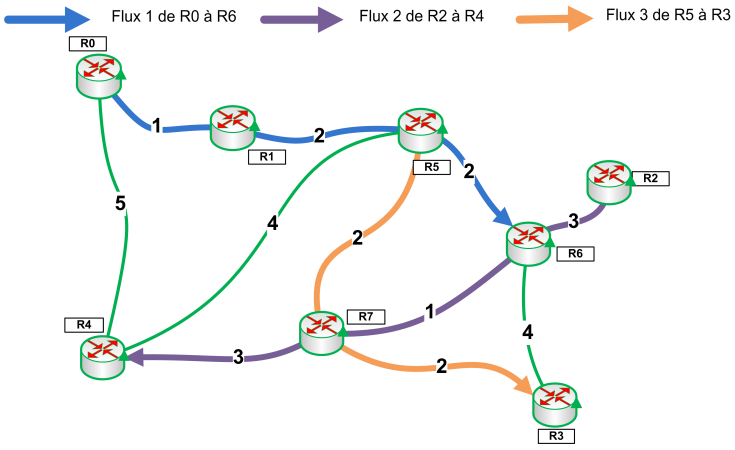


Figure 3.2: Configuration initiale du réseau.

3.5.3.1 Cas d'une panne sans le mécanisme RAR

Afin de visualiser les effets du dispositif RAR par rapport au fonctionnement standard d'OSPF nous commençons par illustrer les effets d'une panne sans notre mécanisme proactif. La Fig. 3.3 montre l'apparition d'une panne sur le routeur R1. Ce routeur faisant partie du chemin emprunté par le flux 1, ce flux est interrompu, car les données envoyées vers R1 sont

perdues. Le protocole *Hello* permet de détecter la panne et d'initier le processus de convergence afin de rétablir le flux du trafic 1.

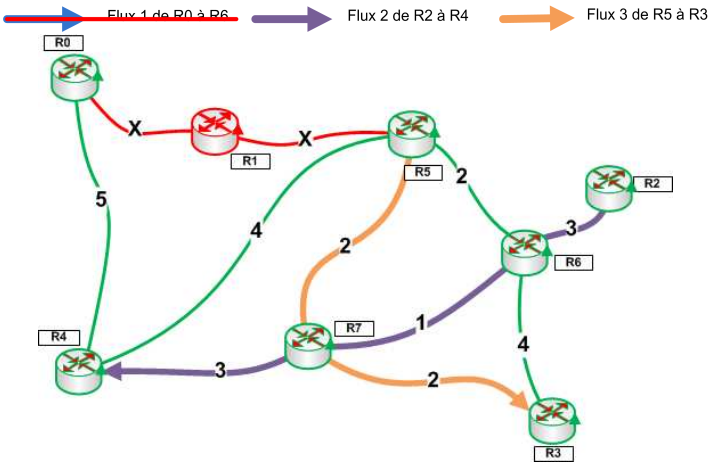


FIGURE 3.3: Panne avec l'utilisation d'un protocole de routage IGP.

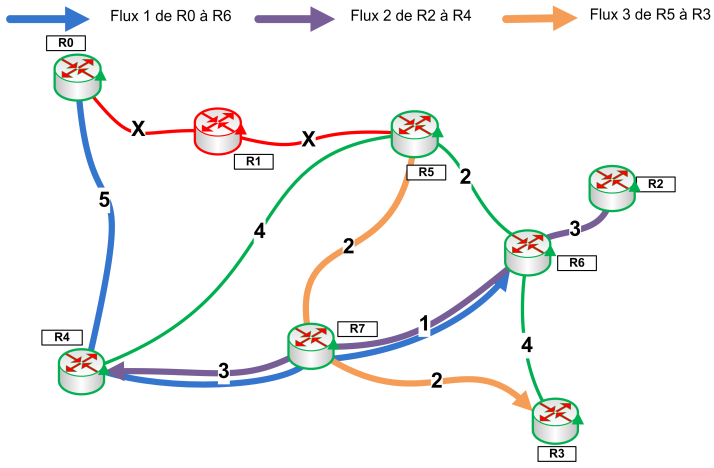


FIGURE 3.4: Situation après la convergence du protocole IGP.

Après plusieurs secondes d'interruption de service, le processus de convergence se termine et permet aux routeurs d'avoir une vision correcte du réseau permettant de rétablir le flux 1. La Fig. 3.4 montre l'état du réseau une fois le flux 1 rétabli en utilisant les routeurs R4 et R7 pour contourner la panne. Les autres flux n'étant pas impactés par la panne, leur routage n'est pas modifié.

3.5.3.2 Cas d'une panne avec le mécanisme RAR

Avec le dispositif RAR, la situation peut être bien différente. Les équipements du réseau sont équipés de modules de prédiction de pannes RAM qui observent l'état du réseau en continue. À partir de la configuration initiale de la Fig. 3.2, la détection d'un risque de panne sur le routeur R1 aboutie à la situation représentée par la Fig. 3.5. Le module RAR présent dans le RM_DE du routeur R1 modifie les métriques de toutes ses interfaces réseaux pour atteindre une valeur suffisamment grande pour obliger le trafic passant par R1 à utiliser un chemin alternatif. Le flux 1 est donc préventivement router vers les routeurs R4 et R7. En conséquence plus aucune donnée n'est commutée par le routeur R1 dont l'état laisse penser qu'une panne est imminente.

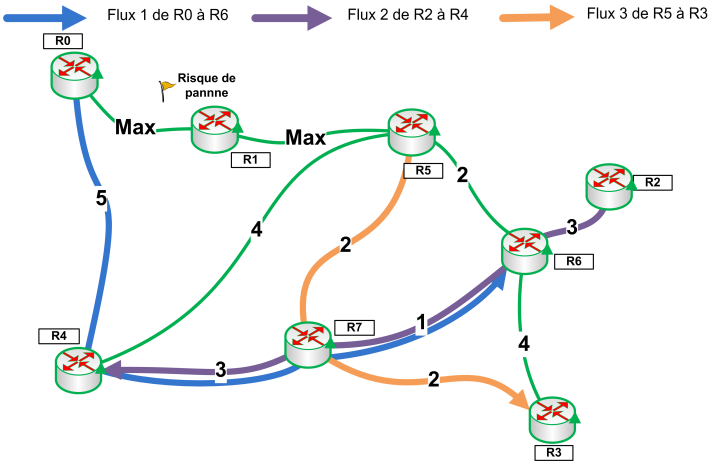


FIGURE 3.5: Situation lors d'une panne avec le mécanisme RAR.

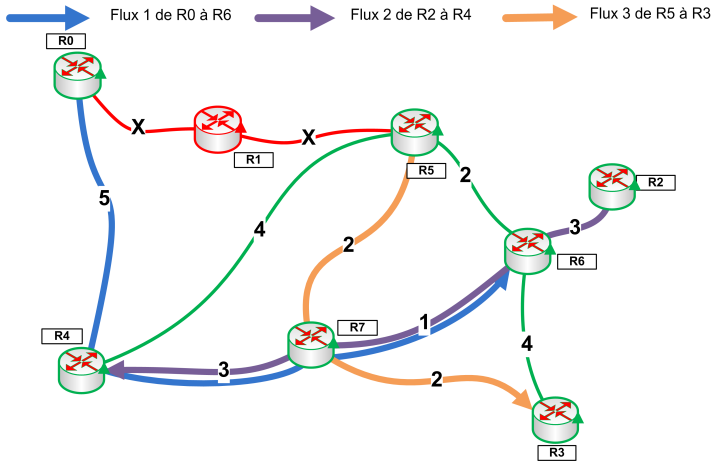


FIGURE 3.6: Situation après la convergence du protocole IGP.

Quelques secondes plus tard, lorsque la panne apparait, celle-ci n'impact pas les flux de données puisqu'aucun flux n'utilise le routeur R1. La convergence du protocole de routage IGP permet de supprimer le routeur R1 de la topologie mais ne modifie pas le routage des flux. En

effet, le mécanisme RAR permet d'anticiper le routage qui sera effectif après le processus de convergence (voir Fig. 3.6), à ceci près que les interfaces du routeur R1 sont toujours utilisables dans le cas où celui-ci serait la seule alternative pour certain flux. Grâce à l'utilisation du mécanisme RAR, aucune dégradation de service n'est constatée par les utilisateurs du flux 1 contrairement à une situation où seule la restauration IP serait utilisée pour gérer les pannes.

3.5.4 Algorithme

L'Algorithme 2 résume le déroulement du dispositif RAR suite à l'envoi d'une information de risque de panne par le RAM au RM_DE. Cette information de risque de panne peut concerner trois types d'équipements : le routeur (noté $n \in N$ où N est l'ensemble des routeurs du réseau), un lien du routeur (noté *link*) ou l'interface à laquelle est rattaché ce lien sur le routeur (noté IF).

algorithme 2 Modification dynamique des métriques de routage en fonction du risque de pannes.

```
1: for each router n_i in N do
2:     if n_i.new_risk > n_i.old_risk then
3:         for each link_j in n_i.links do
4:             if link_j.metric= RiskyMetric(link_j) then
5:                 repeat
6:                     increase link_j.metric without loop [FB07]
7:                     propagate LSA
8:                     compute Shortest Path First
9:                     update Routing table
10:                until link_j.metric = RiskyMetric(link_j)
11:            end if
12:        end for
13:    else if n_i.new_risk <n_i.old_risk then
14:        for each link_j in n_i.links do
15:            if link_j.new_risk = normal AND link_j.If.new_risk = normal then
16:                repeat
17:                    decrease link_j.metric without loop [FB07]
18:                    propagate LSA
19:                    compute Shortest Path First
20:                    update Routing table
21:                until link_j.metric = InitialCost(link_j)
22:            end if
23:        end for
24:    else if n_i.new_risk = normal then
25:        for each link_j in n_i.links do
26:            if (link_j.new_risk > link_j.old_risk) OR (link_j.If.new_risk > link_j.If.old_risk) then
27:                if link_j.metric = RiskyMetric(link_j) then
28:                    repeat
29:                        increase link_j.metric without loop [FB07]
30:                        propagate LSA
31:                        compute Shortest Path First
32:                        update Routing table
33:                    until link_j.metric = RiskyMetric(link_j)
34:                end if
35:            else if (link_j.new_risk < link_j.old_risk) OR (link_j.If.new_risk < link_j.If.old_risk) then
36:                if link_j.new_risk = normal AND link_j.If.new_risk = normal then
37:                    repeat
38:                        decrease link_j.metric without loop [FB07]
39:                        propagate LSA
40:                        compute Shortest Path First
41:                        update Routing table
42:                    until link_j.metric = InitialCost(link_j)
43:                end if
44:            end if
45:        end for
46:    end if
47: end for
```

3.5.5 Quelques mots sur une application à GMPLS

L'applicabilité d'un routage sensible au risque de pannes est étudié dans un environnement IP pur. Mais le concept est également applicable au réseau GMPLS d'une manière différente. Le fonctionnement en mode connecté des réseaux GMPLS n'est pas aussi simple et facile à mettre en place qu'un réseau IP, mais permet grâce aux mécanismes d'ingénierie de trafic de maîtriser complètement le routage des flux. Cela a pour avantage qu'il est alors possible de mettre en place un routage sensible au risque de pannes qui n'engendrerait pas de problème de congestion. Pour cela, il n'est plus utile de modifier la métrique de liens, mais plutôt d'utiliser les mécanismes de *Make-Before-Break* (MBB) [ABG+01] permettant de changer le routage d'un LSP sans perturber le trafic. Le MBB, qui est plus amplement utilisé dans le Chap. 4, permet de modifier un LSP en prenant soin de signaler un nouveau LSP avec le nouveau routage dans un premier temps, pour ensuite transférer le trafic sur ce nouveau LSP pour enfin ne supprimer la ressource de l'ancien LSP qu'une fois le trafic entièrement basculé sur le nouveau tunnel. Afin de contourner les éléments risqués, le protocole GMPLS dispose d'un objet *Exclude Route Object* (XRO) [LFC07] dont l'ensemble des éléments seront exclus du routage. Il est donc nécessaire de placer les éléments risqués dans l'objet XRO pour créer un nouveau LSP contournant les équipements susceptibles de tomber en panne. Grâce à l'option *Shared Explicit* [ABG+01] du MBB, il est possible de réutiliser les ressources de l'ancien LSP lors de la construction du nouveau LSP et ainsi conserver la gestion fine des ressources que permet l'ingénierie de trafic de GMPLS. Enfin, si le reroutage n'était pas possible à cause du manque de ressources, les mécanismes de préemption permettrait d'outrepasser cette limite si telle était la volonté de l'opérateur. Mais dans tous les cas, la congestion ne serait pas possible, même avec de nombreuses fausses prédictions, contrairement à l'application du dispositif RAR au routage IP.

Néanmoins, les fonctionnalités de gestion de pannes du protocole GMPLS permettent d'envisager des alternatives moins extrêmes qu'un reroutage. Le Chap. 4 propose une solution basée sur l'adaptation du mécanisme de résilience en fonction du risque de pannes. Mais ce chapitre est dédié à l'étude du comportement du mécanisme RAR au routage IP et à son évaluation, notamment par l'intermédiaire de la modélisation analytique qui est proposée à la section suivante.

3.6 Modélisation analytique

3.6.1 Définitions et notations

Les notations de la modélisation analytique du dispositif RAR sont en bien des points similaires aux notations du modèle appliqué au mécanisme AFDT et définit à la Sec. 2.6.1, à savoir que le réseau est représenté par un graphe orienté $G=(N,E)$ où N est l'ensemble des nœuds (routeurs) du réseau et E l'ensemble des liens orientés. Le trafic est représenté par l'ensemble F des flux de trafic transportés par le réseau G où chaque flux de trafic $f \in F$ est défini par son routeur source $In(f) \in N$, son routeur destination $Out(f) \in N$ et son débit $\mu(f)$ exprimé en bits/s. Pour chaque flux de trafic $f \in F$, le protocole de routage (OSPF ou IS-IS) définit le plus court chemin (*shortest path*) composé des routeurs de transit par le sous-ensemble $sp(f) \subseteq N$. Avec le mécanisme RAR, suite à une prédiction de panne sur le nœud, de nouvelles métriques sont configurées, modifiant certains plus courts chemins $sp(f)$ également notés $sp_{RAR}(f)$ avec $f \in F$ tel que $n \notin spf_{RAR}(f)$.

Dans ce modèle, les conséquences d'une panne sur les nœuds sources et destinations ne sont pas prises en compte puisque les mécanismes de résilience intra-domaine n'ont aucun moyen de rétablir une telle situation. Néanmoins, dans les réseaux d'opérateurs, de telles circonstances sont gérées par des techniques de *multi-homing* impliquant une redondance des nœuds d'extrémité et l'intervention de plusieurs réseaux.

La comparaison entre le comportement du dispositif RAR avec le comportement standard des protocoles IGP varie suivant deux critères, la probabilité de panne d'un nœud et la prédiction de pannes.

Pour commencer, la probabilité de panne est caractérisée par le $MTBF(n)$, le temps moyen

entre deux pannes du nœud $n \in N$, et le $MTTR(n)$, le temps moyen de réparation du nœud $n \in N$. Le temps de réparation est très inférieur au temps entre deux pannes $MTTR(n) << MTBF(n)$. Pour un processus ergodique stationnaire, la probabilité qu'un nœud $n \in N$ soit en panne $P_{node}(n)$ est :

$$P_{node}(n) = \frac{MTTR(n)}{MTBF(n) + MTTR(n)} << 1 \tag{3.1}$$

Lorsque les routeurs sont considérés comme identiques, le paramètre n est omis dans les notations de tous les paramètres. Concernant le mécanisme RAR, celui-ci est paramétré par les deux variables de performance de la prédiction de pannes (voir Sec. 1.6.3) que sont le *Recall* et la *Precision*. Le $Recall(n)$ et la $Precision(n)$ sont les variables associées à chaque routeur $n \in N$. Enfin, tous ces paramètres sont considérés comme non modifiables dans le temps.

Lors d'une panne, le protocole de routage intra-domaine enclenche le processus de convergence, qui vise à restaurer les flux de trafic impactés par la panne. Pendant la durée de ce processus noté t_C, les flux $f \in F$ qui passent par le nœud $n \in N$ en panne ne délivrent plus les données à leur destination. Ce processus de convergence est décomposé en quatre étapes. L'étape de détection de la panne t_D, l'étape de diffusion de l'information de la panne à tout le réseau t_F, l'étape de calcul des plus courts chemins t_{SP} et l'étape de mise à jour de la table de routage et de la table de *forwarding*. Compte tenu de toutes ces étapes, le temps de convergence est obtenu en sommant chacune des étapes :

$$t_C = t_D + t_F + t_{SP} + t_U \tag{3.2}$$

Si les étapes t_F et t_U ont des valeurs constantes, ce n'est pas le cas de l'étape de calcul des plus courts chemins et de détection de la panne.

Le temps de calcul des plus courts chemins dépend du nombre de nœuds du réseau qui selon l'article [GRcF03] est exprimé par la formule suivante pour un routeur de type Cisco 3600 :

$$t_{SP}(N) = 2,47.10^{-6} * |N|^2 + 9,78.10^{-3} \tag{3.3}$$

Néanmoins, pour plus de clarté dans les équations suivantes, on omettra la dépendance avec N en notant t_{SP} le temps de convergence du réseau.

L'étape de détection de la panne dépend de la période entre deux envois de messages *Hello* (*Hello Interval*) notée t_{HI}, ainsi que de la valeur du compteur *Router Dead Interval*, notée t_{RDI}. Pour des raisons de simplicité, le compteur *Router Dead Interval* est bien souvent un multiple du *Hello Interval*, tel que $t_{RDI} = 4 * t_{HI}$, afin qu'une panne soit détectée suite à la non réception de quatre messages *Hello* . En considérant que l'occurrence d'une panne est uniforme entre deux messages *Hello*, le temps moyen de détection d'une panne est :

$$t_D = (t_{RDI} - t_{HI}) + \frac{t_{HI}}{2} \tag{3.4}$$

Dans cette thèse, sachant que $t_{RDI} = 4 * t_{HI}$, l'Eq. (3.4) peut être remplacée par

$$t_D = (3 * t_{HI}) + \frac{t_{HI}}{2} \tag{3.5}$$

Avec t_C le temps moyen de convergence avec l'utilisation d'un temps de détection moyen t_D, la formule exprimant le temps de convergence moyen lors d'une panne sur le réseau G est :

$$\begin{aligned} t_C &= t_D + t_F + t_{SP} + t_U \\ t_C &= (t_{RDI} - t_{HI}) + \frac{t_{HI}}{2} + t_F + t_{SP} + t_U \end{aligned} \tag{3.6}$$

3.6.1.1 Estimation de l'exposition au risque

En partant du postulat que lors de l'anticipation d'une panne le mécanisme RAR permet au trafic de ne pas expérimenter de perte telle qu'observée avec notre prototype décrit à la Sec. 3.10, il est possible de calculer le risque global de panne sur le trafic à l'instant t d'une prédiction pour

la période Δt_p [VC10]. Soit $P_{node}(n,t)$ la probabilité de panne d'un nœud à l'instant t d'une prédiction. Lors d'une prédiction, et lorsqu'aucun nœud n'est en panne, la probabilité d'une panne d'un nœud non risqué pendant la période Δt_p est $P_{node}(n,t) = (\Delta t_p/MTBF)*(1-Recall)$ et celle d'un nœud risqué est $P_{node}(n,t) = Precision$. En considérant les pannes indépendantes, *i.e.* $pnode(n_i \cap n_j) = pnode(n_i) \cap pnode(n_j)$, la probabilité de panne d'un flux $f \in F$ à un instant $t \in T$ s'écrit :

$$P_{flow}(f,t) = 1 - \left(\prod_{n \in sp(f)} (1 - P_{node}(n,t)) \right) \tag{3.7}$$

Il est utile de noter que pour le mécanisme RAR, lorsqu'une prédiction de panne cible un nœud originel $n \in sp(f)$, l'Eq. (3.7) utilise le nouveau plus court chemin $sp_{RAR}(f)$ comme valeur de $sp(f)$.

À l'échelle du réseau, il est possible de calculer l'exposition au risque moyen d'un bit en utilisant la moyenne des expositions au risque de chaque flux pondérée par leur débit $\mu(f)$:

$$P_{bit}(G,F,t) = \frac{\left(\sum_{f \in F} P_{flow}(f,t).\mu(f) \right)}{\sum_{f \in F} \mu(f)} \tag{3.8}$$

Sachant que $Precision >> (1/MTBF) * (1 - Recall)$, il semble probant que le dispositif RAR qui permet d'éviter les routeurs risqués permette une baisse de l'exposition au risque du trafic réseau.

3.6.1.2 Estimation de l'indisponibilité du réseau

En considérant le fonctionnement du mécanisme de convergence des protocoles IGP tels que OSPF et IS-IS est conforme à l'Eq. (3.2), l'indisponibilité moyenne du flux $f \in F$ suite à la panne du routeur $n \in N$ est exprimée par la formule suivante :

$$\begin{aligned} U_{IGP}(f,n) &= P_{node}(n).t_C/MTTR(n) \\ U_{IGP}(f,n) &= P_{node}(n).((t_{RDI} - t_{HI}) + \frac{t_{HI}}{2} + t_F + t_{SP} + t_U)/MTTR(n) \end{aligned} \tag{3.9}$$

Pour le mécanisme RAR, la probabilité conditionnelle est divisée en une somme de deux probabilités concernant le cas d'une panne prédite ($Recall$) et celui d'une panne non prédite ($1 - Recall$) :

- dans le cas où la panne n'est pas prédite ($1 - Recall$), le trafic s'appuie sur le mécanisme de restauration du protocole IGP avec une indisponibilité du flux qui suit l'Eq. (3.9) ;
- dans le cas où une prédiction de panne anticipe l'occurrence de la panne ($Recall$), le dispositif RAR affecte un nouveau plus court chemin $sp_{RAR}(f)$ au flux f, tel que $n \notin sp(f)_{RAR}$, supprimant de ce fait l'indisponibilité due au délai de convergence. Dans ce cas de figure, on a donc $t_C = 0$;

À partir des deux hypothèses précédentes, l'indisponibilité d'une panne du flux f suite à une panne d'un routeur $n \in sp(f)$ est :

$$\begin{aligned} U_{RAR}(f,n) = & P_{node}.Recall(n).0/MTTR(n) \\ & + P_{node}(n).(1 - Recall(n)).t_C/MTTR(n) \end{aligned} \tag{3.10}$$

Le mécanisme permettant de supprimer l'indisponibilité pour toutes les pannes prédites, l'indisponibilité imputable au mécanisme de résilience du protocole de routage IGP ne subsiste que pour les pannes non prédites :

$$U_{RAR}(f,n) = (1 - Recall(n)).U_{IGP}(f,n) \tag{3.11}$$

L'Eq. (3.11) met en évidence l'importance du $Recall$ dans la disponibilité assurée par le mécanisme RAR, et où le pire cas ($Recall(n) = 0$) conduit à un comportement similaire aux

dispositifs standards des protocoles IGP définis par l'Eq. (3.9) et le meilleur cas ($Recall(n) = 1$) assure une disponibilité totale. Avec la mise en évidences des compteurs de détection de pannes, l'Eq. (3.11) s'écrit :

$$U_{RAR}(f,n) = \frac{P_{node}(n).(1 - Recall(n)).\left(t_{RDI} - t_{HI} + \frac{t_{HI}}{2} + t_F + t_{SP} + t_U\right)}{MTTR(n)} \tag{3.12}$$

Ainsi, en considérant que les pannes des routeurs sont des événements indépendants, les indisponibilités moyennes du flux $f \in F$ pour les deux mécanismes comparés (IGP et RAR) sont :

$$\begin{aligned} U_{RAR}(f) &= 1 - \left(\prod_{n \in sp(f)} (1 - U_{RAR}(f,n))\right) \approx \sum_{n \in sp(f)} U_{RAR}(f,n) \\ U_{IGP}(f) &= 1 - \left(\prod_{n \in sp(f)} (1 - U_{IGP}(f,n))\right) \approx \sum_{n \in sp(f)} U_{IGP}(f,n) \end{aligned} \tag{3.13}$$

Cette approximation est valide puisque suivant l'Eq. (3.1), la probabilité de cas de pannes simultanées d'au moins deux routeurs dans un même réseau est suffisamment petite pour être négligée.

Enfin, pour calculer l'indisponibilité du réseau, il est nécessaire de définir une moyenne de l'indisponibilité de chaque flux f pondérée par le débit de chaque flux $\mu(f)$. X prenant la valeur des deux mécanismes comparés dans ce chapitre ($X = RAR$ et IGP), l' indisponibilité moyenne est donnée par U_X :

$$U_X(G,F) = \frac{\sum_{f \in F} \mu(f).U_X(f)}{\sum_{f \in F} \mu(f)} \tag{3.14}$$

3.6.1.3 Estimation des oscillations du routage

L'avantage du dispositif RAR est une disponibilité plus grande ; mais en contrepartie, les mauvaises prédictions entraînent des modifications du routage. Ces modifications ont pour conséquence un routage temporairement sous-optimal, mais surtout, si elles sont trop nombreuses, elles créent une instabilité néfaste pour les performances du réseau. Les opérateurs étant très attentifs à cette stabilité, il est important de mesurer le nombre d'oscillations du routage qu'implique le mécanisme RAR et de le comparer au comportement avec un protocole IGP standard.

Avec un protocole IGP standard, tel OSPF, les changements de route interviennent lors d'une panne. En effet, lorsqu'un nœud n tombe en panne, le mécanisme de résilience va recalculer la liste des plus courts chemins en prenant bien soin de supprimer le nœud n de la topologie, créant ainsi une modification du routage. Une fois le nœud n réparé, le protocole de routage réintègre le nœud n à la topologie pour recalculer les plus courts chemins afin de revenir à la situation initiale. Une panne engendre donc deux modifications du routage qui, associés à la probabilité de panne, permettent de calculer le nombre d'oscillations par seconde d'un protocole IGP standard :

$$RF_{IGP}(G,F) = 2 * \left(\sum_{n \in N} P_{node}(n)/MTTR(n)\right) \tag{3.15}$$

Avec le dispositif RAR, il est nécessaire de considérer le cas des TP et des FP en faisant intervenir le $Recall$ et la $Precision$:

- lors d'une fausse prédiction, la modification des métriques change le routage une première fois, puis à la fin de Δt_p, la prédiction se termine et les métriques sont réinitialisées à leur valeur initiale, ce qui crée une deuxième modification du routage. Il est donc nécessaire d'ajouter deux oscillations à chaque mauvaise prédiction ;

– lorsqu'une panne est prédite par le RAM, les métriques sont augmentées afin de repousser le trafic hors du nœud critique. Cette opération crée une première modification du routage. Ensuite, lors de la panne, la suppression du nœud en panne crée une modification du routage si faible qu'elle est négligeable. En effet, l'opération de modification des métriques lors de l'anticipation de la panne modifie le routage vers une situation similaire au cas de panne. La quasi-totalité des routes sont identiques. Lors de la panne, seule les routes ayant pour source ou destination le nœud en panne sont modifiées, car elles sont tout simplement supprimées pendant la durée de la panne. Néanmoins, ces flux sont gérés par les mécanismes inter-domaine de *multi-homing*. On peut donc considérer le routage intra-domaine comme stable du début de la prédiction de panne, jusqu'à la fin de la panne. Une fois réparés, les métriques sont réinitialisées à leur valeur d'origine et le nœud anciennement en panne est réintégré à la topologie pour un recalcul du routage similaire au cas sans prédiction. Le cas avec ou sans prédiction engendrant tous deux seulement deux oscillations, il est inutile de prendre en compte les prédictions de panne dans la modélisation.

À partir de ces observations, du *Recall* et de la *Precision* définis aux Eq. (1.1) et (1.2), la fréquence d'oscillations du mécanisme RAR est :

$$\begin{aligned} RF_{RAR}(G,F) =& 2 * \left(\sum_{n \in N} * \left(\frac{1}{Precision(n)} - 1 \right) . \left(\frac{P_{node}(n).Recall(n)}{MTTR(n)} \right) \right) \\ &+ 2 * \left(\sum_{n \in N} P_{node}(n)/MTTR(n) \right) \end{aligned} \tag{3.16}$$

Il est intéressant de noter que le nombre d'oscillations additionnelles du dispositif RAR par rapport au protocole IGP standard est égal à deux fois le nombre de fausses prédictions. Cette constatation est plus évidente en simplifiant l'Eq. (3.16) avec $RF_{IGP}(G,F)$:

$$RF_{RAR}(G,F) = 2 * \left(\sum_{n \in N} \left(\frac{1}{Precision(n)} - 1 \right) . \left(\frac{P_{node}(n).Recall(n)}{MTTR(n)} \right) \right) + RF_{IGP}(G,F) \tag{3.17}$$

3.7 Étude de cas : trois réseaux de classe opérateur

De manière identique au Chap. 2, les trois configurations de réseaux cœurs (voir Sec. 2.7) sont utilisées pour analyser le comportement du mécanisme RAR proposé vis-à-vis du mécanisme standard de restauration OSPF. Pour rappel, la première topologie est un réseau national allemand (Fig. 2.7a) composé de 17 nœuds similaire au réseau utilisé dans [MMJ08]. Le deuxième réseau est la topologie NSF-Net composée de 29 nœuds (Fig. 2.7b). Enfin, la dernière topologie est un réseau européen composé de 34 nœuds (Fig. 2.7c). Les configurations précises de ces trois topologies sont disponible à l'annexe A, où les capacités (Tab. A.3, A.8 et A.13), les métriques de routage (Tab. A.2, A.7 et A.12) et la matrice de trafic (Tab. A.5, A.10 et A.15) de chaque topologie (Fig. A.1, A.2 et A.3) sont indiquées de manière détaillés.

3.8 Application numérique du modèle analytique

Dans un premier temps, une application du modèle de calcul de risque de panne a été faite sur les trois configurations réseaux. L'objectif de cette première analyse est de constater que lors d'une prédiction de panne, l'utilisation du dispositif RAR permet de réduire fortement le risque de panne quel que soit la fréquence des pannes (MTBF) et la durée de validité des prédictions (Δt_p). Puis, dans une étude plus pragmatique, le modèle de calcul de l'indisponibilité et de la stabilité du routage a été appliqué aux trois topologies susmentionnées, afin d'étudier l'impact des deux paramètres caractérisant les pannes que sont le MTBF et le MTTR sur le comportement

du mécanisme RAR. De même, les répercussions du *Recall*, de la *Precision* et de Δt_p ont été analysées.

Pour cela, les mêmes conditions de référence que celles du Chap. 2 ont été utilisées. Il s'agit d'un MTBF de 5000 heures et d'un MTTR de 5 heures identiques pour chaque routeur. De même, la prédiction de pannes est considérée comme identique sur chaque routeur et avec une durée de validité de référence d'une prédiction Δt_p d'une heure. Bien que l'état de l'art propose des valeurs de Δt_p de quelques minutes [ST08], nous avons constaté qu'une période d'une heure n'engendrait pas de différence significative dans le comportement de notre mécanisme. Étant donné que plus Δt_p est grand, plus la prédiction de pannes est performante, cette valeur d'une heure permet une compatibilité avec la quasi-totalité des mécanismes de prédiction de pannes, et laisse une marge suffisamment importante pour espérer rester compatible avec les futures fonctionnalités de prédiction que proposeront les équipements de réseau. Ce Δt_p d'une heure est donc utilisé pour les neuf configurations de prédiction de pannes de références formées par la combinaison d'un *Recall* et d'une *Precision* de 20%, 50% et 80%.

Enfin, les différents délais des étapes de la convergence du protocole IGP ont été affectés suivants les valeurs relevées dans la littérature [SG01, GRcF03]. Ainsi, 0,03 secondes sont considérées pour le temps de diffusion t_F [SG01] et 0,2 secondes pour le délai de mise à jour des tables de routage et de *forwarding* t_U de 0,2 [GRcF03]. Pour le délai de détection de pannes le nombre de messages *Hello* nécessaire pour détecter une panne est fixé à quatre tel que $t_{RDI} = 4 * t_{HI}$ et la période d'envoi des messages *Hello* t_{HI} est de une seconde. L'utilisation d'une telle valeur pour t_{SHI} est volontairement agressive. Elle correspond à une valeur inférieure à la majorité des configurations des opérateurs et est une limite utilisée afin de maintenir une certaine stabilité dans le routage en évitant les fausses détections de panne. Ainsi, la configuration du protocole *Hello* permet de comparer le gain du dispositif RAR par rapport à un protocole IGP ayant une durée de convergence t_C inférieure à la plupart des configurations utilisées par les opérateurs et ainsi de s'assurer de la validité de notre étude dans la majorité des cas.

3.8.1 Minimisation du risque de panne

Même avec des mécanismes de gestion des pannes, les opérateurs préfèrent qu'elles soient évitées. Le mécanisme RAR a pour objectif de re-router le trafic vers des routeurs sains afin que lorsqu'une panne apparaît, celle-ci n'impacte pas le trafic circulant sur le réseau. Afin de quantifier cet avantage, le risque de panne sur le trafic ,que les opérateurs ont tout intérêt à minimiser, est calculé. Le risque de panne dont il est question dans cette section correspond au risque de panne moyen sur un bit lors d'une prédiction de panne. Le mode de calcul de ce risque de panne est décrit à la Sec. 3.6.1.1 ainsi que dans l'article « Proactive fault management based on risk-augmented routing »[VC10]. Il permet de juger de l'utilité du re-routage effectué par le dispositif RAR lors d'une prédiction de panne.

La Fig. 3.7 montre ce risque sur la topologie européenne avec des prédictions dont la durée de validité est de une heure et avec un MTBF variant entre 1000 heures et 10000 heures. Les résultats sont découpés en quatre groupes, dont un groupe avec le risque minimum qui est composé des résultats obtenus avec l'ensemble des configurations du mécanisme RAR. Les trois autres groupes concernent le mécanisme de routage IGP standard où chaque groupe correspond à une valeur de *Precision* du mécanisme de prédiction de pannes. Les résultats sont similaires pour la topologie allemande et pour la topologie US (voir Fig.B.10) et peuvent être consultés en Annexe B.2. On peut observer que la variation de la fréquence des pannes entre un MTBF de 1000 heures et un MTBF de 10 000 heures n'a pas un grand impact sur le risque de panne pendant une prédiction. En effet celui-ci dépendant surtout de la *Precision* du module de prédiction de pannes à l'origine de la prédiction. Mais, même avec une *Precision* de seulement 20%, le risque de panne d'un routeur concerné par une prédiction reste beaucoup plus élevé qu'un routeur non risqué. Cela incite à utiliser le re-routage proposé par le dispositif RAR afin de réduire ce risque. Les résultats soulignent le choix judicieux de routage du mécanisme RAR lors d'une prédiction, en permettant une baisse sensible du risque de panne par rapport au routage effectué par les protocoles IGP actuels.

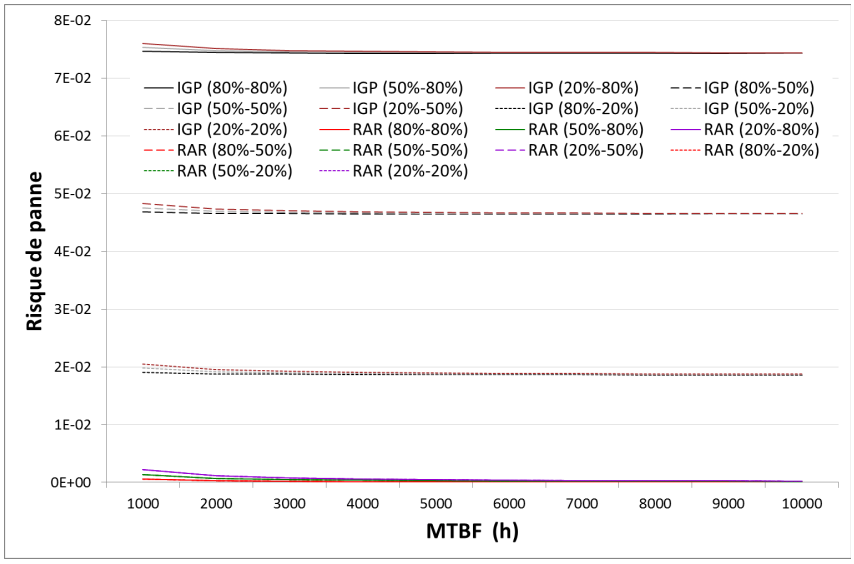


FIGURE 3.7: Impact du MTBF sur le risque de panne avec la topologie européenne.

La Fig. 3.8 montre la faible influence de la durée de cette prédiction Δt_p entre 5 minutes et 10 heures sur le risque de panne. Les résultats de la topologie allemande et de la topologie US (voir Fig. B.11) disponible en Annexe B.2 confirment aussi que le mécanisme RAR permet, lors d'une prédiction, de faire le bon choix de routage permettant de diminuer sensiblement le risque de panne pendant une prédiction.

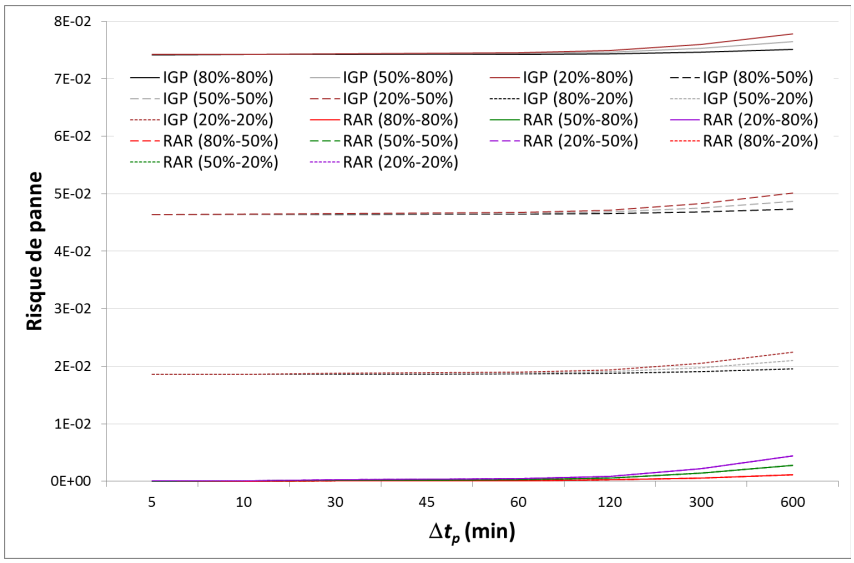


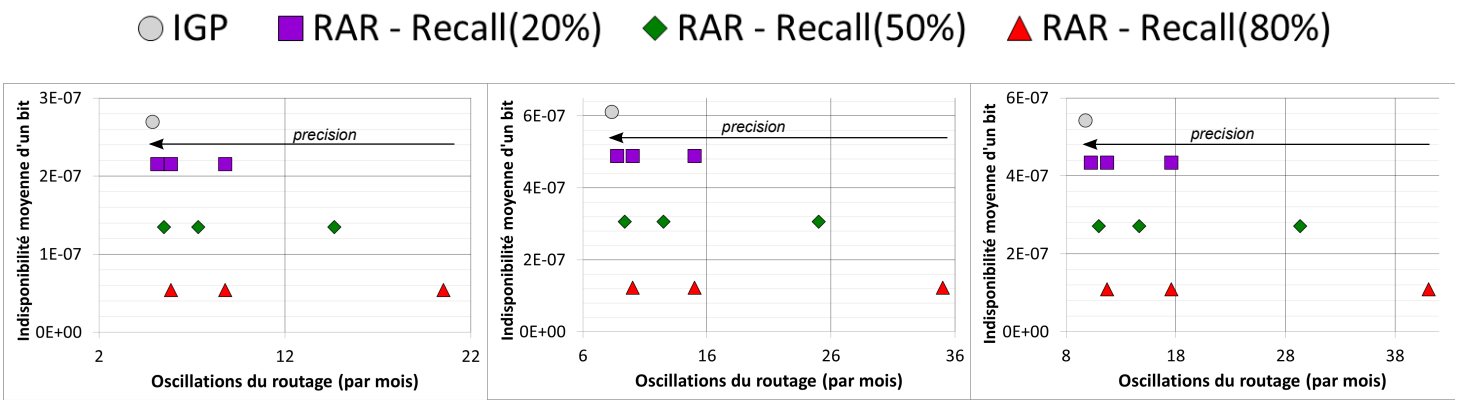
FIGURE 3.8: Impact de Δt_p sur le risque de panne avec la topologie européenne.

Les sections suivantes ont pour objectif de vérifier si cette diminution du risque de panne se traduit concrètement par une meilleure disponibilité des services réseau.

3.8.2 Analyse conjointe de la disponibilité et de la stabilité du routage

La Fig. 3.9 représente l'indisponibilité et le nombre d'oscillations de routage dans les conditions de référence décrites à la Sec. 3.8 pour les trois topologies. Les combinaisons des trois valeurs de *Recall* et de *Precision* de 20%, 50% et 80% pour le dispositif RAR sont comparées à l'utilisation d'un protocole IGP standard tel qu'OSPF. Cette figure illustre l'intérêt du mécanisme RAR par rapport à la restauration utilisée par les protocoles IGP actuels en terme de disponibilité tout en n'apportant pas une instabilité trop importante au routage. Car à l'exception des configurations générant trop de fausses prédictions (*i.e.* avec une *Precision* de 20% et un *Recall* supérieur à 50%), le nombre d'oscillations du routage n'est augmenté que de 70% au

maximum ce qui restreint le nombre de changements de route à 8, 15 ou 17 par mois suivant la topologie. Sachant que ces changement de route sont temporaires et apparaissent exclusivement par deux (autour d'une panne et/ou d'une prédiction), la fréquence moyenne de 2 oscillations tous les mois reste tout à fait acceptable sur un réseau de type opérateur. Le *Recall* a un impact



(a) A (b) US (c) EU

Figure 3.9: *Ratio* indisponibilité / nombre d'oscillations du routage avec la configuration de référence pour les trois topologies.

significatif sur la disponibilité, au contraire de la *Precision*. Concernant le nombre d'oscillations, celui-ci augmente lorsque la *Precision* diminue, mais cette augmentation reste acceptable lorsque la *Precision* est supérieure à 50%. Dans le cas d'une *Precision* de 20%, le nombre de fausses prédictions devient trop important avec un *Recall* de 50% et de 80% engendrant un nombre d'oscillations qui peut devenir supérieur à un changement de routes par jour. La topologie joue ici un rôle important puisque chaque action de re-routage sur un routeur est diffusée à l'ensemble de l'aire de routage. Pour des conditions similaires en terme de probabilité de panne et de performance du module de prédiction de pannes, la topologie avec le moins de nœuds (*i.e.* la topologie allemande) subit moins d'oscillations que les topologies possédant plus de 20 nœuds (*i.e.* les topologies européenne et US). En conséquence, la topologie allemande subit moins de 15 oscillations par mois pour un *Recall* de 50% et une *Precision* de 20% ce qui reste raisonnable, alors que la même *Precision*, et un *Recall* de 80%, entraîne plus de 20 oscillations par mois ce qui reste faible mais peut commencer à être un problème pour certains opérateurs exigeants. Pour les deux autres topologies plus importantes on obtient 25 et 30 changements de routage avec le *Recall* de 50%, ainsi que 35 à 40 changements avec un *Recall* de 80%. Ces valeurs bien qu'encore acceptables, peuvent être rédhibitoires pour des opérateurs n'étant pas prêts à accepter une telle instabilité malgré un gain de disponibilité certain. En effet, cette illustration des conditions de référence pose les bases de l'avantage du mécanisme RAR sur le comportement standard des protocoles IGP avec un gain d'indisponibilité supérieur à cinq pour le mécanisme RAR avec un *Recall* de 80%, de deux pour un *Recall* de 50% et de 1,2 pour un *Recall* de 20%.

Le dispositif permet un gain sensible en disponibilité tout en limitant le nombre d'oscillations de routage à une valeur correcte dans la plupart des cas. Au regard de ces premiers résultats, le choix du RAM semble être restreint par la propension à ne pas trop générer de fausses prédictions. Si les performances du RAM en *Precision* sont faibles, il faudra alors limiter le nombre de prédictions total, et donc le *Recall* afin de ne pas trop perturber le routage. Mais si les performances en *Precision* sont supérieures à 50% (tel que *Recall* $\geq$ 50%), le nombre d'oscillations n'est pas un problème et le mécanisme de prédiction de pannes ayant le meilleur *Recall* est le meilleur choix.

3.8.3 Influence de la probabilité de panne

Il est intéressant d'étudier l'impact de certains paramètres sur le comportement de notre mécanisme RAR. Bien que les trois topologies aient été étudiées, les différents résultats sont similaires dans leur comportement. De manière similaire au Chap. 2, notre analyse est illustrée

avec une seule topologie, à savoir la topologie européenne. Néanmoins, les résultats complets pour chaque topologie sont disponibles en Annexe B.2. Il est intéressant d'évaluer l'impact théorique de tous les paramètres du modèle sur le comportement du dispositif RAR, à commencer par les paramètres caractérisant les pannes, à savoir le MTBF et le MTTR. Pour cela, l'influence du MTBF pour une plage de valeurs comprises entre 1000 et 10000 heures, ainsi que celle du MTTR pour des valeurs allant de 1 à 10 heures, est analysée en terme de disponibilité et de consommation de ressources.

La Fig. 3.10 montre l'impact de la fréquence des pannes sur la disponibilité. On peut noter une augmentation qui s'accélère en dessous d'un MTBF de 3000 heures. L'ordre et le *ratio* entre chaque mécanisme restent les mêmes que ceux constatés sur la configuration de référence (voir Fig. 3.9c). Le mécanisme RAR avec un *Recall* de 80% permet de garder une très bonne disponibilité même lorsque la probabilité de panne est importante. Dans une telle configuration où les oscillations de routage sont déjà fréquentes, l'utilisation du RAR peut permettre de diminuer sensiblement l'indisponibilité. En revanche la durée des pannes ne change en rien l'indisponibilité. Puisque nous ne nous intéressons qu'à l'indisponibilité due aux délais de convergence des protocoles IGP, l'indisponibilité affichée par la Fig. 3.11 reste donc stable quel que soit le MTTR.

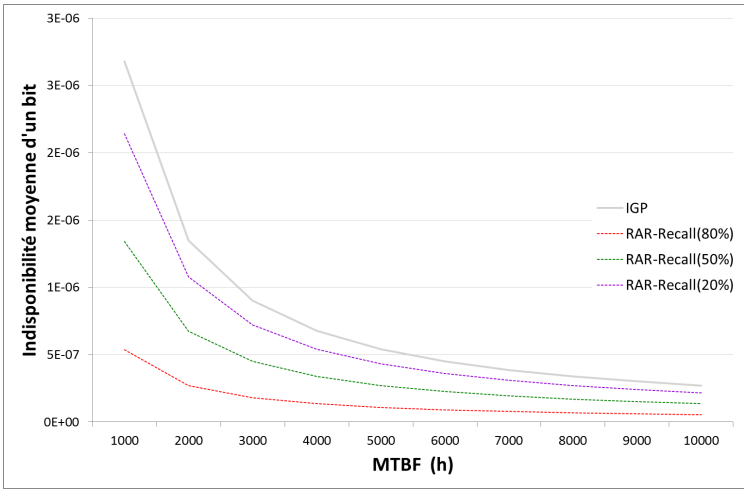


FIGURE 3.10: Impact du MTBF sur la disponibilité avec la topologie européenne.

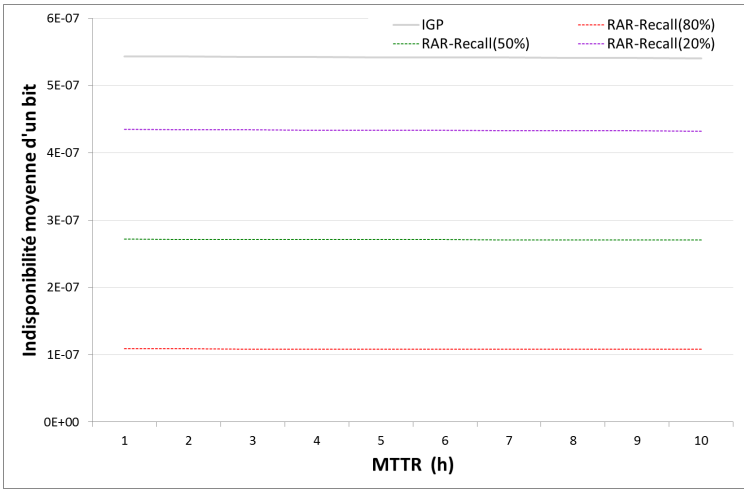


FIGURE 3.11: Impact du MTTR sur la disponibilité avec la topologie européenne.

La mesure de l'instabilité du routage en fonction de la fréquence des pannes de la Fig. 3.12 permet aux opérateurs d'identifier les configurations qui ne respectent pas leur contrainte en terme de nombre de calculs de routes. Le MTBF de 3000 heures est une limite au-delà de laquelle le nombre d'oscillations s'accélére. Le nombre de re-calculs de routes est principalement influencé par les fausses prédictions. Il est logique de retrouver les configurations de *Precision* faible (*i.e.* 20%). Cependant, le *Recall* joue aussi un rôle puisqu'il influence indirectement le nombre de fausses prédictions. La raison est que le nombre de fausses prédictions est caractérisé par la *Precision*, elle-même dépendante du *Recall* (voir Eq. (1.2)). Il est donc logique de retrouver, à *Precision* égale, un nombre supérieure d'oscillation pour les *Recall* importants et même de remarquer qu'avec des *Precision* espacées de 30% , le nombre d'oscillations avec la *Precision* la plus faible et un *Recall* de 20% est égal au nombre d'oscillations avec la *Precision* supérieure et un *Recall* de 80%. Si l'on exclut les valeurs avec une *Precision* faible (*i.e.* 20%), le nombre d'oscillations reste aux alentours de 2 oscillations tous les deux jours maximum jusqu'au MTBF de 3000 heures et ne monte que jusqu'à un peu plus de deux oscillations tous les jours pour un MTBF extrême de 1000 heures. En revanche avec une *Precision* faible et un *Recall* d'au moins 50%, on commence à avoir un nombre élevé d'oscillation dès 4000 heures pour atteindre une oscillation toutes les trois ou quatre heures avec un MTBF de 1000 heures. Il semble donc que lorsque la probabilité de panne est élevée, la précision soit un facteur limitant l'utilisation du dispositif RAR. Mais le grand nombre de re-routages pourrait aussi aboutir à occasionner de la congestion qui anéantirait le gain de disponibilité du mécanisme. Ce comportement difficilement

quantifiable analytiquement est analysé grâce à notre implémentation à la Sec. 3.9.2.

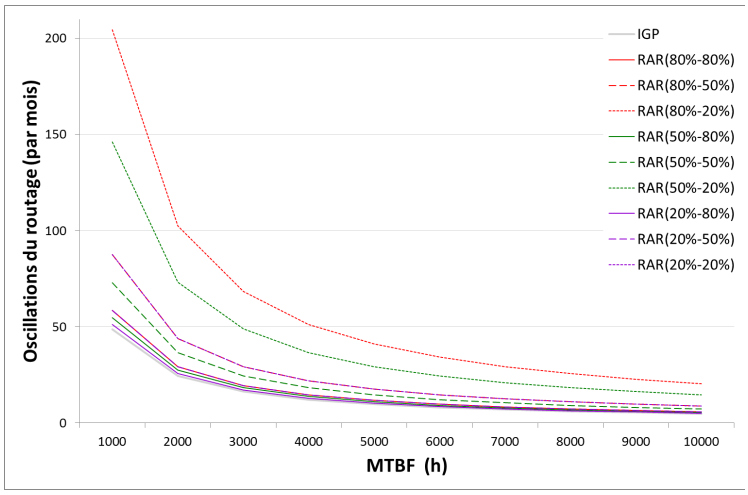


Figure 3.12: Impact du MTBF sur le nombre d'oscillations du routage avec la topologie européenne.

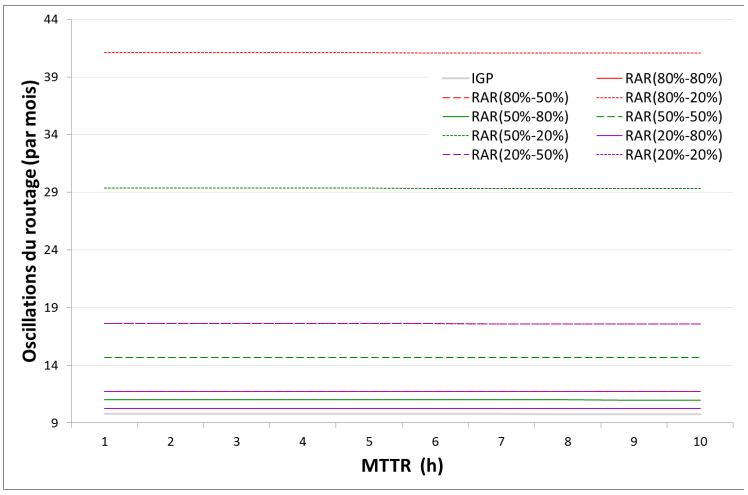


Figure 3.13: Impact du MTTR sur le nombre d'oscillations du routage avec la topologie européenne.

La durée de panne à une influence imperceptible sur le nombre d'oscillations affiché à la Fig. 3.13. Une augmentation du MTTR ne baisse que de manière infime le nombre de panne pour une période donnée et réduit donc le nombre d'oscillations, mais dans des proportions au combien plus faible que le MTBF que ce n'en est pas visible.

3.8.4 Les conséquences de la prédiction de pannes

Il est utile de rappeler qu'aucune méthode de prédiction de pannes associée à notre mécanisme n'est proposée, il est donc indispensable d'identifier les contraintes que doit posséder le module de prédiction de pannes, ainsi que le comportement qui en découle en fonction de ses performances. Pour cela, il est nécessaire d'analyser l'incidence de toutes les valeurs du *Recall* et de la *Precision* associées à nos conditions de références. Le paramètre qui, de prime abord,

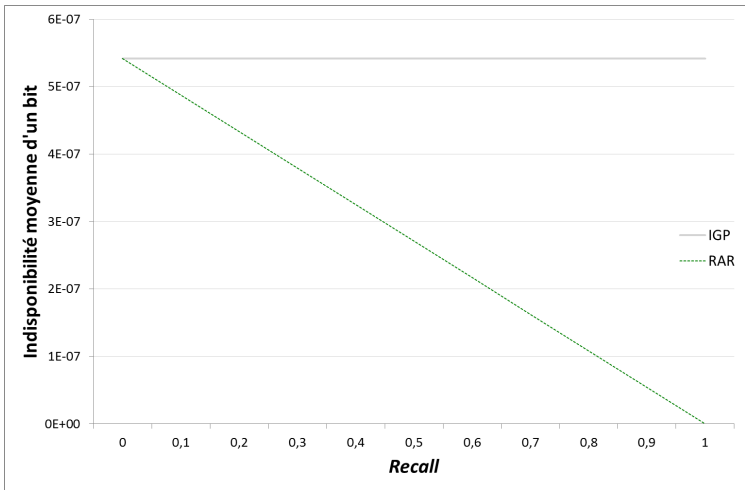


Figure 3.14: Impact du *Recall* sur la disponibilité avec la topologie européenne.

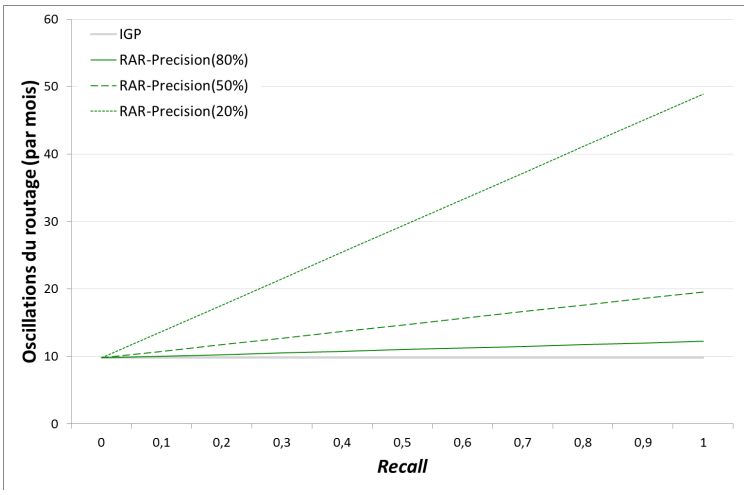


Figure 3.15: Impact du *Recall* sur le nombre d'oscillations du routage avec la topologie européenne.

semble le plus important vis-à-vis du comportement de notre solution en terme de disponibilité est le *Recall*. Celui-ci caractérise la proportion de pannes anticipées par le RAM. La Fig. 3.14 montre fort logiquement une augmentation linéaire des performances en fonction du *Recall*. En commençant avec des performances similaires au dispositif de restauration proposé par les protocoles IGP standards pour le *Recall* le plus faible, le mécanisme RAR assure jusqu'à la disponibilité théorique totale lorsque toutes les pannes sont anticipées (*i.e.* $Recall = 1$). Le mo-

dèle ne considérant aucune incidence de la *Precision* sur la disponibilité, la Fig. 3.14 est donc valable quelle que soit la *Precision* du mécanisme RAR. Un changement de route est effectué lors d'une panne et lors d'une prédiction. Les pannes sont inévitables et le re-routage qui en découle indispensable. Le re-routage d'une bonne prédiction ne fait qu'anticiper le re-routage qui suit une panne, il n'a donc pas d'influence sur la QoS. Mais une fausse prédiction crée une oscillation inutile qui reste acceptable si leurs nombres restent contenus. Le nombre de fausses prédictions est caractérisé par la *Precision*, elle-même dépendante du *Recall* (voir Eq. (1.2)). Cette relation est illustrée sur la Fig. 3.15 qui montre une augmentation du nombre d'oscillations avec le *Recall*.

L'augmentation est plus forte pour les *Precision* faibles et notamment pour le dispositif RAR avec une *Precision* de 20%. En effet, alors que le nombre d'oscillations avec une *Precision* inférieure à 50% ne dépasse pas 20, le mécanisme RAR avec une *Precision* de 20% peut générer jusqu'à 50 oscillations par mois. Enfin, il est intéressant de souligner qu'avec une *Precision* de 80%, le nombre d'oscillations reste très proche du comportement du protocole IGP standard.

La Fig. 3.16 montre l'incidence de la *Precision* sur le nombre de modifications du routage. Le paramètre clé étant la quantité de fausses prédictions, le *Recall* influence indirectement les trois courbes. À *Precision* égale, le mécanisme avec le *Recall* le plus important provoque plus de fausses prédictions et donc un nombre plus important d'oscillations du routage. En fixant à 30 la barrière au-delà de laquelle le nombre d'oscillations peut être un problème, on note qu'avec un *Recall* de 80%, la *Precision* limite est de 30%, qu'avec un *Recall* de 50% elle est de 20%, et que le *Recall* de 20% peut être utilisé avec n'importe quelle *Precision*.

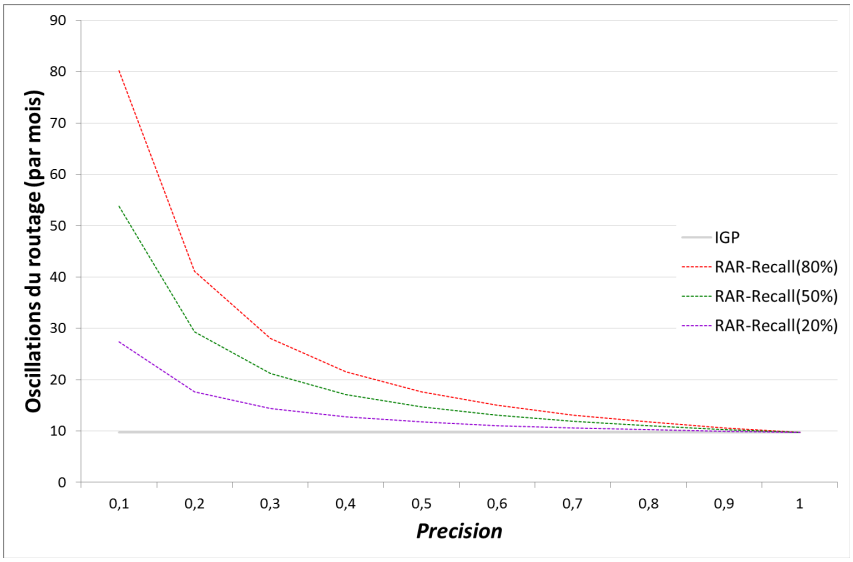


FIGURE 3.16: Impact de la *Precision* sur le nombre d'oscillations du routage avec la topologie européenne.

3.8.5 Les enseignements de l'étude théorique

Le premier enseignement de notre étude est la confirmation que le dispositif de changement des métriques des liens permet de dérouter le trafic et de diminuer sensiblement le risque de panne du trafic lors d'une prédiction de panne. À la vue de ce premier résultat, il paraît intéressant lors d'une prédiction d'utiliser le mécanisme RAR afin de minimiser le risque de panne.

L'application de notre modèle analytique aux trois configurations réseau illustratives souligne l'importance du *Recall* dans les bénéfices qu'apportent le mécanisme RAR et la mise en évidence des difficultés qu'apporterait l'utilisation d'une *Precision* de 20%. Bien que la probabilité de panne ait une influence sur la disponibilité, celle-ci met surtout en évidence une limite du mécanisme avec une *Precision* de 20% en terme d'oscillations du routage, dans les configurations de MTBF inférieur ou égale à 3000 heures. Au contraire, la durée de panne est un paramètre secondaire dont l'impact sur la disponibilité et sur la stabilité du routage est quasiment nul. Mais la

simulation de notre dispositif dans les mêmes conditions permettrait de vérifier ces observations et d'identifier les éventuels effets de congestions dues à de trop nombreux re-routages.

Le mécanisme RAR permet, en fonction du *Recall*, de se rapprocher de la disponibilité totale, tout en générant un nombre d'oscillations du routage acceptable dans la plupart des cas. Il est notamment intéressant de souligner que le mécanisme RAR avec un *Recall* performant de 80% et une *Precision* d'au moins 50% permet un gain important en disponibilité tout en assurant une grande stabilité du routage. Les *Precision* de 20% peuvent générer un nombre trop important d'oscillations qui laisse à penser que certains opérateurs ne sont pas prêts à intégrer une fonctionnalité RAR avec une telle *Precision*.

Les performances observées du dispositif RAR plaident pour l'ajout de ce mécanisme au sein des protocoles IGP, car il est possible, grâce à la prédiction de pannes, d'atteindre une disponibilité beaucoup plus compétitive tout en conservant une stabilité proche de ce qui est en place aujourd'hui. Néanmoins, la faiblesse du mécanisme RAR est le faible contrôle des flux lors du changement des poids des liens. Bien que les capacités des réseaux soient dimensionnées pour supporter les pannes, un nombre important de pannes, de fausses prédictions et/ou un long Δt_p peuvent engendrer des situations de congestion que seul la simulation peut quantifier. Il est donc indispensable d'analyser le comportement du dispositif RAR *via* simulation afin de vérifier la validité de notre modèle de manière pratique, et d'identifier les effets de la congestion sur la disponibilité assurée par le mécanisme.

3.9 Implémentation

L'implémentation du mécanisme RAR dans un simulateur a deux objectifs. Premièrement, celui de valider le modèle analytique proposé à la Sec. 3.6 en reproduisant les conditions étudiées dans la Sec. 3.8. Deuxièmement, vérifier le comportement du mécanisme dans des conditions extrêmes, c'est-à-dire lorsque la probabilité de panne est importante (*i.e.* le MTBF est faible), le Δt_p est important, ou lorsque les fausses prédictions sont nombreuses (*i.e.* avec une *Precision* faible associée à un *Recall* élevé). Car dans ces conditions, le schéma de routage défini par le dispositif RAR peut engendrer de la congestion qui annulerait les bénéfices du mécanisme en terme de disponibilité.

3.9.1 Implémentation du simulateur

La simulation de notre mécanisme a été faite en utilisant la version 3 du simulateur à événements discrets le plus couramment utilisé [nsm11]. En utilisant les implémentations du routage et de gestion des nombres aléatoires, le dispositif RAR de changement dynamique des métriques des liens dont le routeur est risqué, a été implémenté au sein de NS3.

3.9.1.1 Gestion des événements

Les événements qui rythment une simulation sont les pannes, les prédictions de pannes (bonnes ou mauvaises), les actions de modifications des métriques des liens ainsi que les actions de reconvergence du protocole de routage.

Les traces des pannes survenant sur les réseaux d'opérateurs sont des données sensibles, il est donc très difficile de les obtenir. Ne possédant pas de telles données, nous avons choisi d'appliquer la théorie générale de la fiabilité [OL09] afin de générer des pannes aléatoires. Le temps entre deux pannes d'un même nœud est supposé suivre une distribution exponentielle dont la moyenne est égale au MTBF souhaité. Cela revient à générer une distribution exponentielle de paramètre $\lambda = 1/MTBF$. De même, on suppose que la durée d'une panne suit une loi log-normale dont la moyenne par nœud est égale au MTTR et l'écart type égale à $0,6 * MTTR$. La distribution générée pour chaque nœud suit donc une loi $\ln N(\mu, \sigma^2)$ avec $\mu = \log(MTTR) - ((0.5) * \log(1 + ((0.6 * MTTR)^2/MTTR^2)))$ et $\sigma = \sqrt{\log(1 + ((0.6 * MTTR)^2/MTTR^2))}$.

Deux générateurs de distribution (exponentielle et log-normal) sont donc attribués à chaque nœud afin d'être utilisés à chaque panne pour déterminer la durée de celle-ci ainsi que la date de la prochaine. Il n'est donc pas possible de connaître à l'avance le nombre de pannes afin de

respecter le *Recall* et la *Precision* de manière exacte. Nous utilisons donc une variable suivant une distribution uniforme que nous utilisons à chaque panne afin de déterminer selon la valeur du *Recall* si celle-ci est anticipée ou non par le RAM. De plus, une autre variable comprise entre 0 et Δt_p et suivant une distribution uniforme est utilisée afin de déterminer la date d'occurrence de la prédiction.

Un mécanisme différent est utilisé pour générer les fausses prédictions. Dès le début de la simulation, nous estimons en fonction du MTTR et du *Recall*, le nombre de pannes qui devraient être prédite et utilisons ce nombre pour calculer le nombre de fausses prédictions pour chaque routeur. Il ne reste plus alors qu'à utiliser une distribution uniforme pour chaque routeur afin de distribuer les fausses prédictions de manière uniforme tout au long de la simulation.

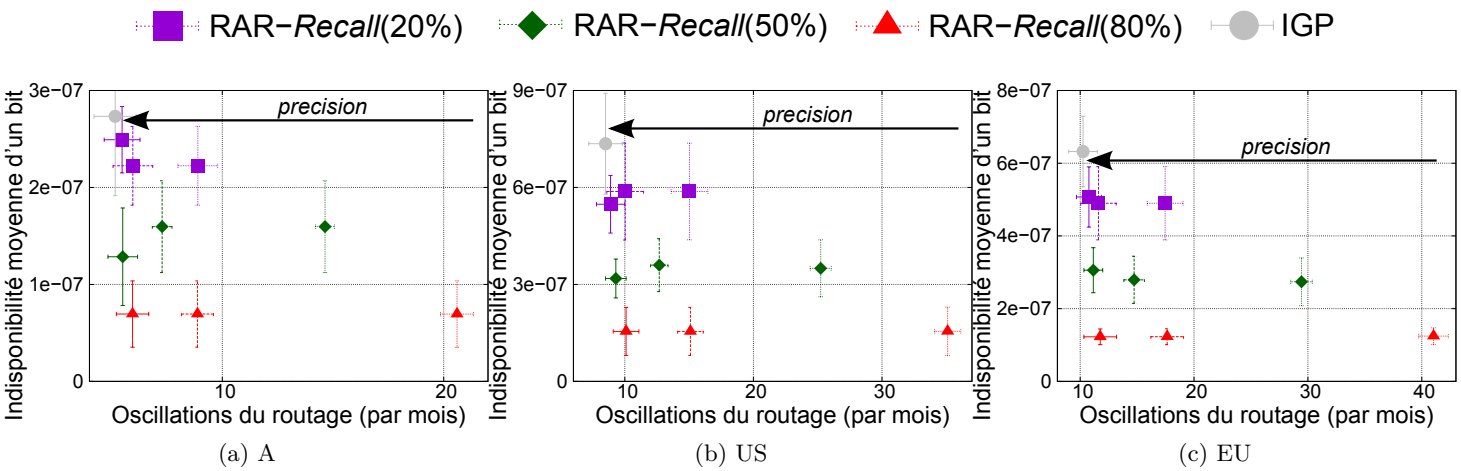
Enfin, afin de se mettre dans les mêmes conditions que notre modèle, le *multi-homing* doit être simulé. Il est donc nécessaire de ne pas inclure l'indisponibilité des flux suite à une panne sur le routeur source $In(f)$ ou le routeur destination $Out(f)$ en arrêtant temporairement les flux concernés pendant la durée de la panne.

3.9.1.2 Scénario des expérimentations

L'utilisation d'un simulateur de niveau paquet tel que NS-3 a pour avantage une précision plus grande au détriment d'une durée de simulation extrêmement longue. Les simulations réalisées utilisent les mêmes configurations et conditions que celles définies pour l'application du modèle analytique à la Sec. 3.8, à ceci près que le débit des flux a dû être réduit afin de pouvoir exécuter les simulations dans un temps raisonnable. Elles utilisent donc les topologies et matrices de trafic définies à la Sec. 3.7 dont les débits des flux ainsi que les capacités des liens sont divisés par 102400. Cette opération n'a pas d'influence sur l'observation des résultats de notre mécanisme, mais permet un gain substantiel de temps, indispensable à notre étude. Néanmoins, chaque simulation reste longue, c'est pourquoi nous avons choisi de limiter le temps simulé à $5*(MTBF+MTTR)$ et d'effectuer seulement sept essais avec une graine différente. Bien qu'un nombre plus important de simulations, ainsi qu'une période de simulation plus longue aurait permis d'obtenir des résultats plus précis, les résultats obtenus affichent des intervalles de confiance satisfaisants (99%). Avec le même réglage du protocole *Hello* que dans l'application numérique précédente (*i.e.* $t_{SHI} = 1s$), nous avons simulé chaque configuration appliquée au modèle analytique, ce qui permet d'étudier l'impact du MTBF, du MTTR, du *Recall*, de la *Precision* et du Δt_p sur la disponibilité et sur le nombre d'oscillations du routage.

3.9.2 Résultats des simulations

3.9.2.1 Analyse conjointe de la disponibilité et de la stabilité du routage



(a) A (b) US (c) EU

FIGURE 3.17: *Ratio* indisponibilité / nombre d'oscillations du routage avec la configuration de référence pour les trois topologies.

La Fig. 3.17 représente l'indisponibilité et le nombre d'oscillations de routage des simulations dans les conditions de référence décrites à la Sec. 3.8 pour les trois topologies. Les combinaisons des trois valeurs de *Recall* et de *Precision* de 20%, 50% et 80% pour le dispositif RAR sont comparées à l'utilisation d'un IGP standard tel que OSPF. Cette figure confirme les résultats obtenus lors de l'application du modèle analytique aux intervalles de confiance près. Les enseignements sont donc les mêmes, à savoir, un gain de disponibilité important, tout en conservant un routage stable excepté pour certain cas lorsque la *Precision* est de 20%.

Comme à la Sec. 2.9.2 des résultats de disponibilité différent en fonction de la *Precision*. Ceci n'est pas dû au mécanisme en lui-même, mais aux effets de bord de la gestion des nombres aléatoires. Pour rappel, cette différence de disponibilité qui reste faible est due à l'utilisation de générateurs de nombres aléatoires supplémentaires afin de répartir correctement les fausses prédictions lorsque celles-ci sont peu nombreuses. L'utilisation de ces variables supplémentaires a pour effet de modifier la distribution du TBF, du TTR et des pannes prédites et aboutissent logiquement à une disponibilité différente.

3.9.2.2 Influence de la probabilité de panne

Pour les mêmes raisons que lors de l'analyse des résultats analytiques, l'analyse est illustrée avec une seule topologie, à savoir la topologie européenne. Néanmoins les résultats de simulations complets pour chaque topologie sont disponibles en Annexe C.2. L'intérêt de cette section est de vérifier que le comportement du mécanisme RAR est conforme à nos résultats analytiques, notamment dans les valeurs extrêmes, c'est-à-dire pour un MTBF faible (par exemple 1000 heures) et un MTTR élevé (par exemple 10 heures). Pour cela, des simulations ont été effectuées avec les conditions étudiées par le modèle analytique, c'est-à-dire un MTBF compris dans une plage de 1000 à 10000 heures, ainsi un MTTR prenant des valeurs allant de 1 à 10 heures.

Tout d'abord le MTTR n'a pas d'influence sur la disponibilité comme l'illustre la Fig. 3.18 qui affiche des résultats similaires au modèle analytique.

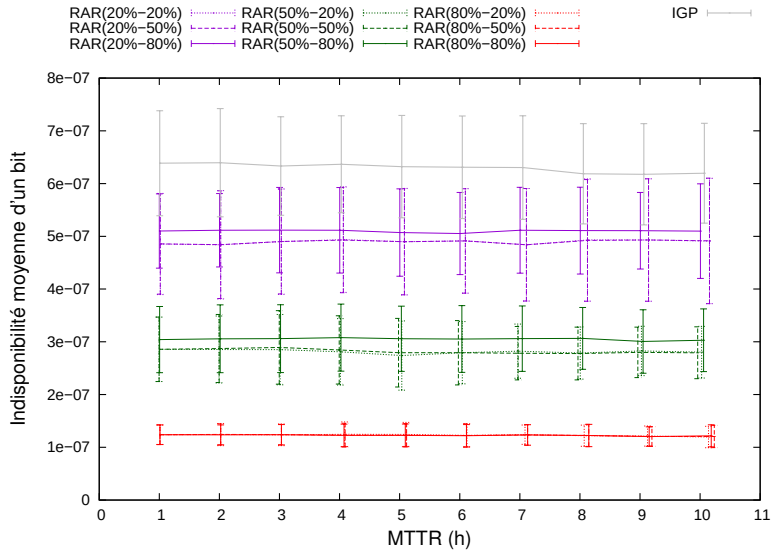


Figure 3.18: Impact du MTTR sur la disponibilité avec la topologie européenne.

Cependant, le MTBF entraîne un comportement bien différent. Comme le montre la Fig. 3.21, à partir d'un MTBF de 2000 heures, des résultats de disponibilité sont bien plus importants que ceux estimés par le modèle analytique. Les intervalles de confiance montrent une grande variation entre les différentes simulations symptomatique d'un phénomène de congestion. Ces cas de congestion apparaissent pour tous les cas de faible *Precision* mais aussi avec une *Precision* de 50% associé à un *Recall* de 80% et, dans une moindre mesure avec un *Recall* de 50%. En effet, lorsque de trop nombreux reroutages sont actif en même temps, cela perturbe le routage de manière trop importante, en envoyant le trafic sur des liens non dimensionnés pour gérer une telle situation. Les pertes dû à la congestion sont alors très couteuse, créant une situation inacceptable

pour les opérateurs de réseaux. Ce phénomène est également visible pour la topologie allemande sur la Fig. 3.19 et pour la topologie US sur la Fig. 3.20. Cependant, la taille de la topologie joue un rôle puisque qu'on peut observer que la topologie allemande, plus petite, est moins exposée à cette congestion.

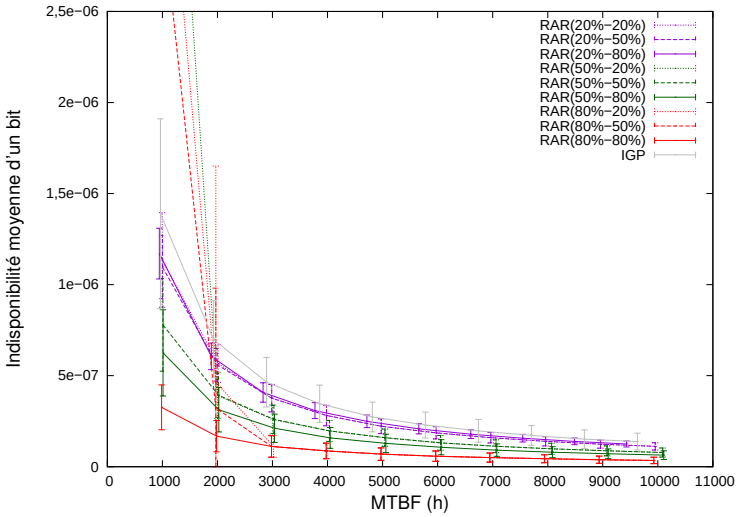


FIGURE 3.19: Impact du MTBF sur la disponibilité avec la topologie allemande.

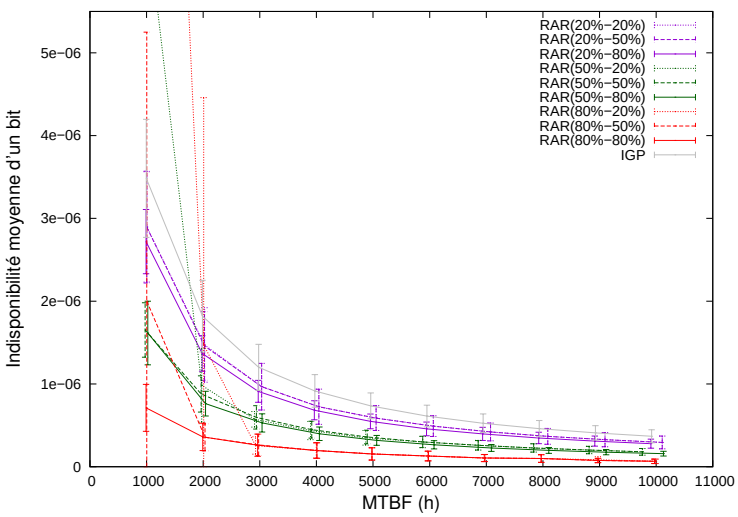


FIGURE 3.20: Impact du MTBF sur la disponibilité avec la topologie US

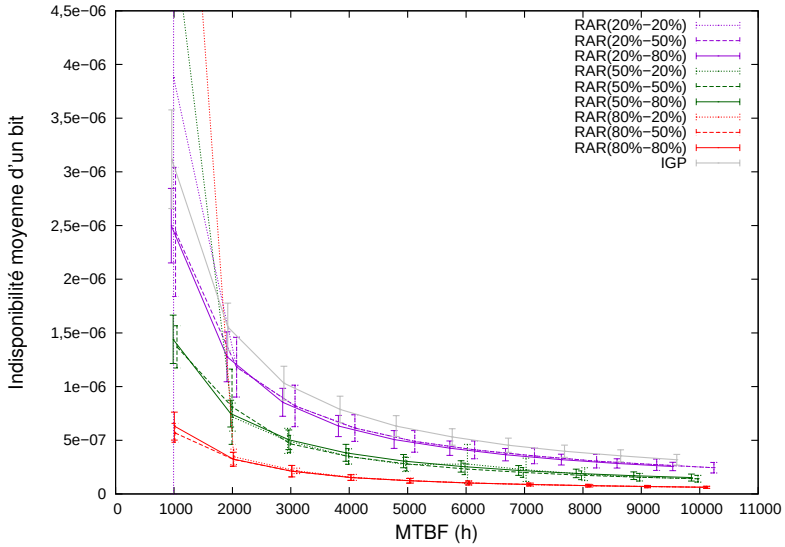


FIGURE 3.21: Impact du MTBF sur la disponibilité avec la topologie européenne.

L'enseignement principal à tirer de ces simulations est la confirmation de la validité de notre modèle, excepté lorsque la congestion apparaît, c'est-à-dire à partir d'un MTBF de 3000 heures. L'indisponibilité assurée par le dispositif RAR avec une *Precision* de 80% voire même de 50%, est plus intéressante que le mécanisme IGP standard pour n'importe quel MTBF entre 1000 et 10000 heures. Pour une *Precision* de 20%, cela est aussi vrai mais uniquement jusqu'à un MTBF de 3000 heures, au-delà duquel de la congestion apparaît et engendre une indisponibilité supérieure à ce que fournit le protocole IGP.

Les résultats de l'incidence du MTBF et du MTTR sur le nombre d'oscillations des simulations sont représentés sur les Fig. 3.22 et 3.23. Les valeurs mesurées sont similaires aux résultats analytiques et confirment la grande stabilité du mécanisme RAR avec une *Precision* supérieure à 50% jusqu'à un MTBF de 3000 heures. Dans les autre cas de figure, la stabilité reste néanmoins correcte sauf avec une *Precision* de 20% et un *Recall* de 50% ou 80% et un MTBF inférieur à 4000 heures. La Fig. 3.23 ne met en lumière aucune incidence du MTTR conformément à notre

modèle analytique.

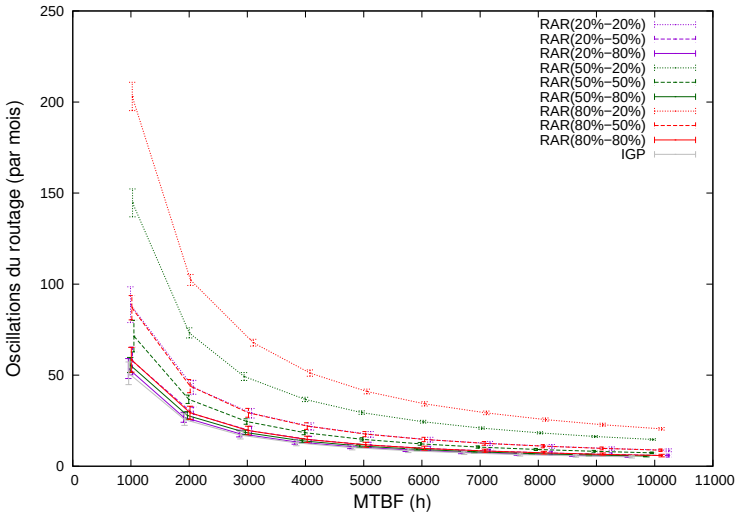


Figure 3.22: Impact du MTBF sur le nombre d'oscillations du routage avec la topologie européenne.

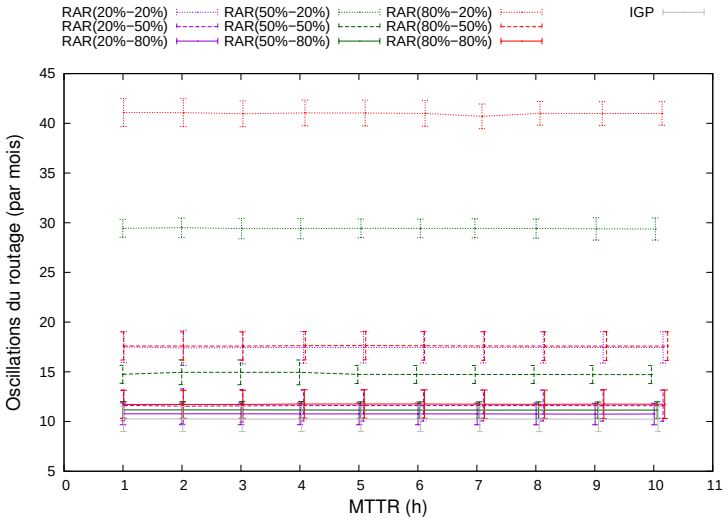


Figure 3.23: Impact du MTTR sur le nombre d'oscillations du routage avec la topologie européenne.

3.9.2.3 Les conséquences de la prédiction de pannes

Après avoir vu les premières limites du dispositif lors de l'utilisation d'une prédiction de pannes avec une faible *Precision* pour des fortes probabilités de panne, il est important d'étudier plus finement l'incidence des caractéristiques de la prédiction de pannes. Pour cela nous simulons la même configuration que celle étudiée avec le modèle analytique.

Le cas $Recall = 0$ n'est pas traité, car dans ce cas de figure, le nombre de fausses prédictions à générer pour respecter la *Precision* définie par l'Eq. (1.2) n'as plus aucun sens. Or, mis à part cette spécificité, la Fig. 3.24 montre des résultats de disponibilité presque identiques à la Fig. 3.14. Une exception notable est la configuration utilisant une *Precision* de 20% pour un *Recall* de 1. Dans cette configuration, le nombre de fausses prédictions entraîne de la congestion annulant les gains en disponibilité.

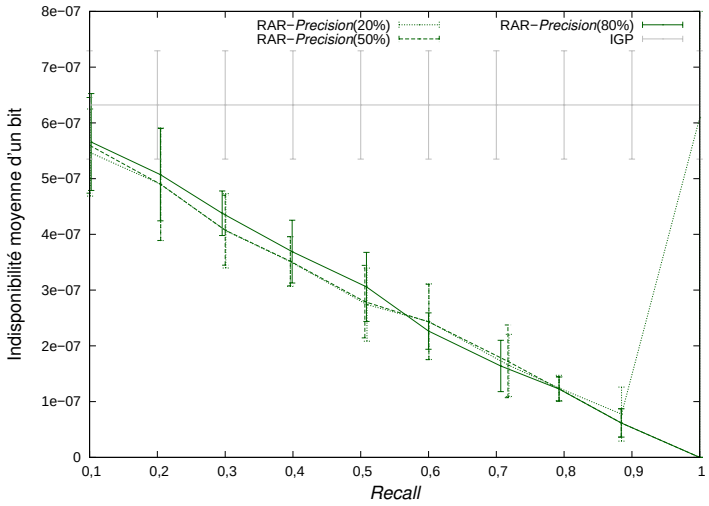
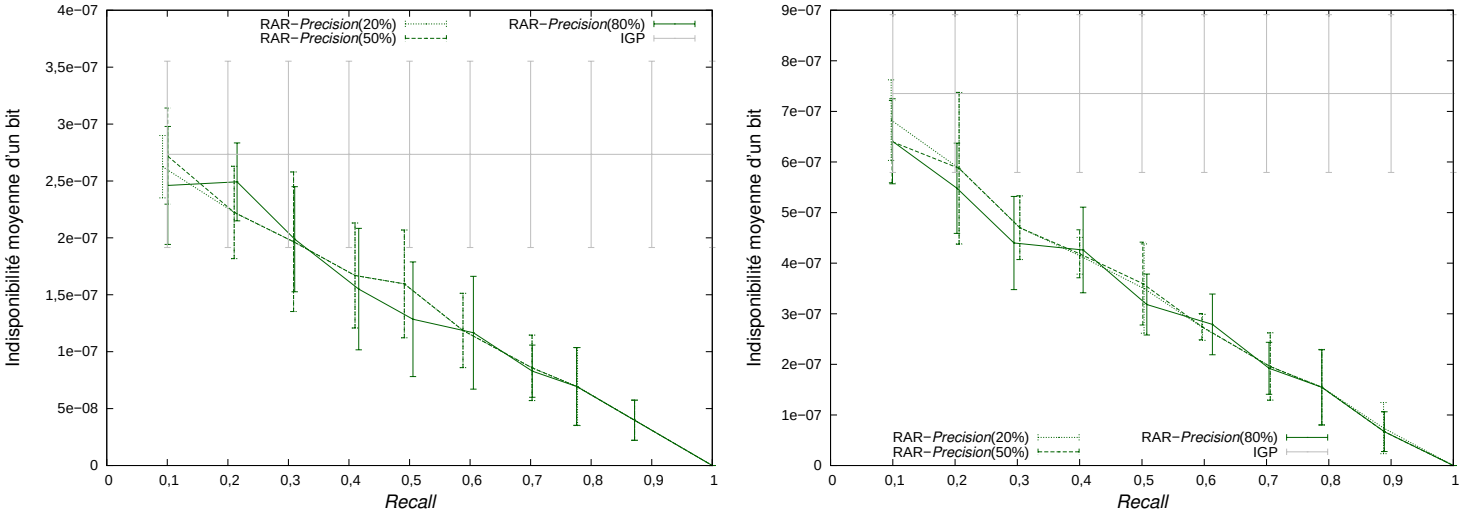


Figure 3.24: Impact du *Recall* sur la disponibilité avec la topologie européenne.

Néanmoins, ce phénomène n'apparaît que sur la configuration la plus extrême en termes de fausses prédictions et disparaît sur les topologies de réseaux allemande et US qui sont plus petites. La Fig. 3.25 montre que ces deux autres topologies suivent en tout point le comportement affiché par le modèle analytique aux intervalles de confiance de 99% près.



(a) A (b) US

FIGURE 3.25: Impact du *Recall* sur la disponibilité avec les topologies allemande et US.

La similitude des résultats de simulation de la Fig. 3.26 avec les résultats analytiques de la Fig. 3.15 ainsi que les intervalles de confiance permette de valider notre modélisation de l'incidence du *Recall* sur les oscillations de routage. L'augmentation du nombre d'oscillations est bien dépendante du *Recall* avec une pente plus accentuée si la *Precision* est faible.

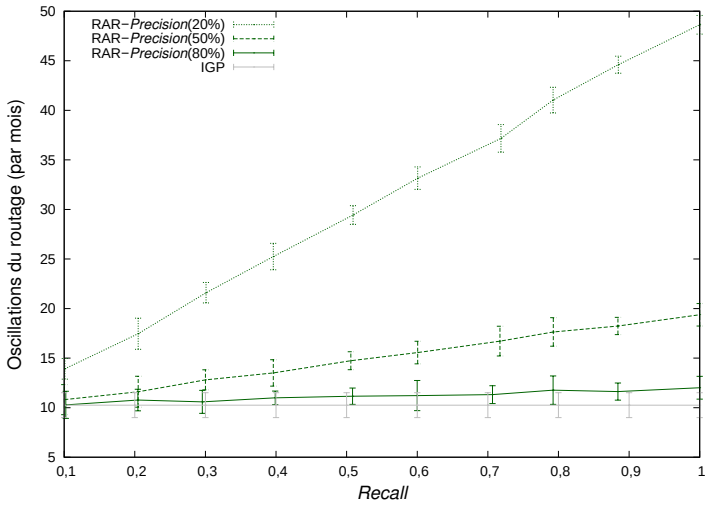


FIGURE 3.26: Impact du *Recall* sur le nombre d'oscillations du routage avec la topologie européenne.

La Fig. 3.27 permet de constater que la *Precision* n'influence pas l'indisponibilité dans les conditions de référence. En effet, dans ces conditions, le nombre de faux positifs n'est pas suffisant pour provoquer de la congestion. Comme au Chap. 2, la Fig. 3.27 met en évidence le comportement relatif aux nombres aléatoires qui créent une instabilité dans les résultats pour certain cas où la *Precision* est importante. Ce phénomène n'apparaît pas aux mêmes valeurs de *Precision* suivant le *Recall*. Il n'est d'ailleurs pas présent dans le cas d'un *Recall* de 80%. Ce phénomène ne se déclenche que lorsque le nombre de fausses prédictions devient très faible. Dans le cas normal, les simulations utilisent un même nombre de variable générant des nombres aléatoires. Ceci implique une probabilité de panne identique et une même graine ainsi que des simulations subissant exactement les mêmes pannes. Il est donc logique d'observer des résultats de disponibilité similaires lorsque le paramètre n'influence pas le mécanisme de convergence comme c'est le cas sur la Fig. 3.30. Cependant, dans le cas où le nombre de fausses prédictions est faible, il est nécessaire de répartir correctement les quelques fausses prédictions de manière

aléatoire et uniforme. Pour cela, notre implémentation crée dynamiquement de nouvelles variables suivant des distributions uniformes le cas échéant. Le résultat de cette opération crée un décalage dans la gestion des nombres aléatoires d'une simulation avec une même graine, responsable d'un déroulement différent de la simulation notamment vis-à-vis des pannes, mais aussi de la prédiction de pannes. Ce comportement n'enlève en rien la validité de notre implémentation, mais il est nécessaire de comprendre son origine, afin de ne pas faire de mauvaise interprétation des résultats obtenus.

Compte tenu de cette information, la Fig. 3.27 confirme l'indépendance entre la disponibilité et la *Precision* utilisée par le mécanisme RAR, mais à condition que le nombre de fausses prédictions ne perturbe pas le routage au point de créer de la congestion. Sur la Fig. 3.28, on observe qu'avec un *Recall* de 80% et une *Precision* de 10%, la quantité importante de reroutage aboutie à des phénomènes de congestion qui ont une incidence importante sur la disponibilité. Une fois encore, cela ne concerne qu'un cas extrême où la prédiction de pannes est effectuée avec une *Precision* médiocre.

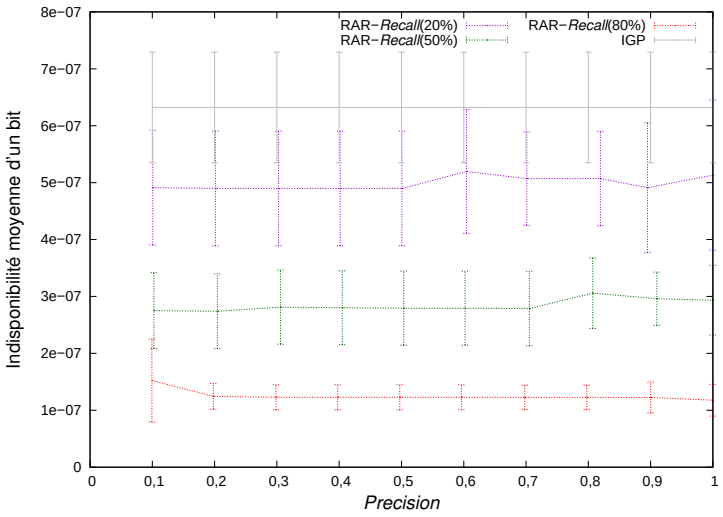


Figure 3.27: Impact de la *Precision* sur la disponibilité avec la topologie européenne.

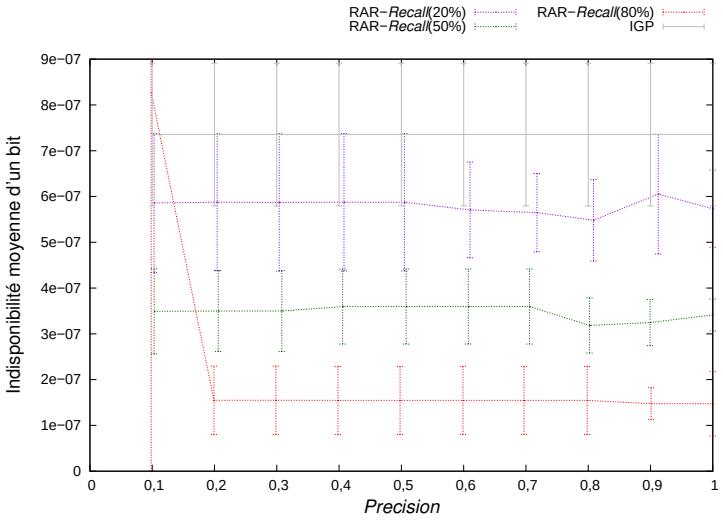


Figure 3.28: Impact de la *Precision* sur la disponibilité avec la topologie US

L'étude de la variation de la *Precision* sur le nombre d'oscillations de routage qui est visible sur la Fig. 3.29 est similaire aux résultats de l'étude numérique, ce qui renforce notre confiance dans le modèle de calcul des oscillations de routage.

Compte tenu de l'exposition du dispositif RAR à la congestion avec une période de validité de prédiction de une heures, il paraît évident que l'analyse de l'impact d'un Δt_p variant entre 5 minutes et 10 heures permettra de mettre en évidence la limite acceptable pour ce paramètre. La Fig. 3.30 montre une disponibilité constante quelle que soit la durée de la prédiction, sauf avec une *Precision* de 20% et un *Recall* de 80%, à partir d'un Δt_p de 5 heures. Ce résultat confirme la validité de notre modèle qui ignore Δt_p dans le calcul de la disponibilité. La Fig. 3.30 met en évidence la possibilité d'utiliser un Δt_p jusqu'à 10 heures sans générer de congestion sauf avec un *Recall* de 80% et une *Precision* de 20%. Dans cette configuration bien particulière un Δt_p de 2 heures constitue une limite sur la topologie européenne.

Sur la Fig. 3.31 avec la topologie allemande, plus petite, le phénomène de congestion n'est visible que pour la configuration engendrant le plus de fausses prédictions (*i.e.* avec un *Recall* de 80% et une *Precision* de 20%), mais seulement lors de l'utilisation d'un Δt_p de 10 heures. Par contre, les résultats avec topologie US (Cf Fig. 3.32) souligne une exposition à la congestion plus importante. Pour cette topologie, tous les mécanismes RAR avec une *Precision* de seulement 20% subisse de la congestion. Avec un *Recall* de 20% seul l'utilisation d'un Δt_p de 10 heures est concerné, mais avec un *Recall* de 50%, la *Precision* de 20% expérimente de la congestion à partir de $\Delta t_p = 5$ *heures* alors que la *Precision* de 50% subit elle aussi de la congestion lorsque $\Delta t_p = 10$ *heures*. Pour terminer le mécanisme RAR avec un *Recall* de 80% est la configuration qui subit le plus rapidement les effets de la congestion. Avec une *Precision* de 20% la congestion

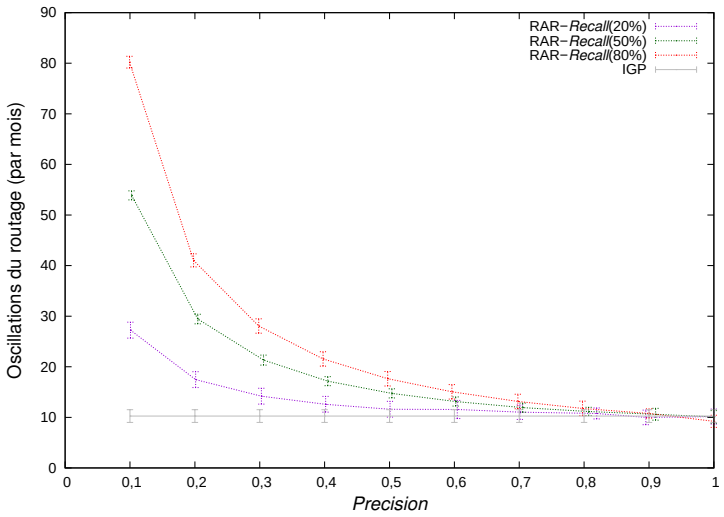


FIGURE 3.29: Impact de la *Precision* sur le nombre d'oscillations du routage avec la topologie européenne.

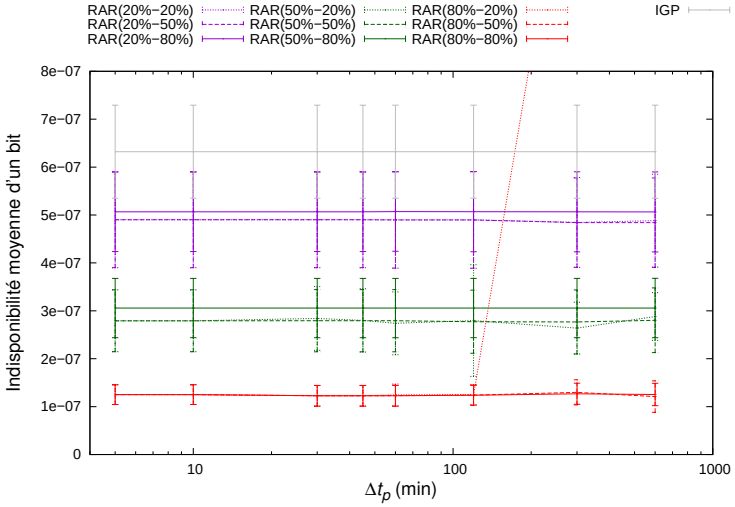


FIGURE 3.30: Impact de Δt_p sur la disponibilité avec la topologie européenne.

se manifeste à partir de $\Delta t_p = 2$ *heures*, mais devient vraiment problématique pour un Δt_p de 5 heures. Même avec une *Precision* de 50% ou de 80%, les Δt_p de 5 et 10 heures en sont pas compatibles avec l'utilisation du dispositif RAR sur la topologie US.

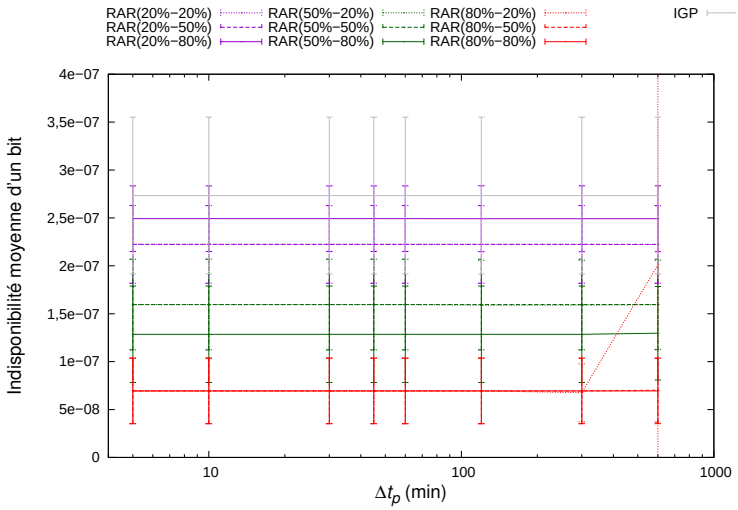


FIGURE 3.31: Impact de Δt_p sur la disponibilité avec la topologie allemande.

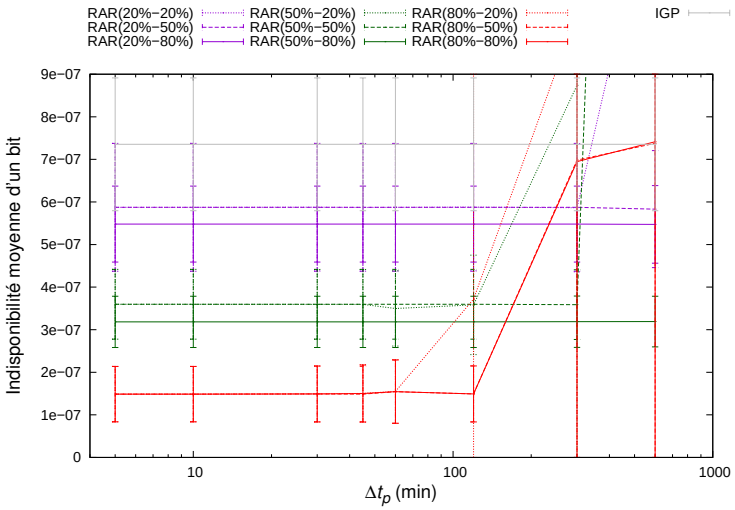


FIGURE 3.32: Impact de Δt_p sur la disponibilité avec la topologie US

Le nombre de re-calculs de routes illustré à la Fig. 3.33 est conforme aux observations de notre application numérique. Elle confirme que la durée de validité d'une prédiction n'intervient pas dans le nombre d'oscillations du routage.

Ces simulations permettent donc de valider les hypothèses de notre modèle analytique, et d'identifier les limites de notre mécanisme RAR par rapport à la probabilité de panne, et notamment du MTBF, mais surtout par rapport aux caractéristiques du module de prédiction de pannes intégré au RAM tels que le *Recall*, la *Precision* et le Δt_p.

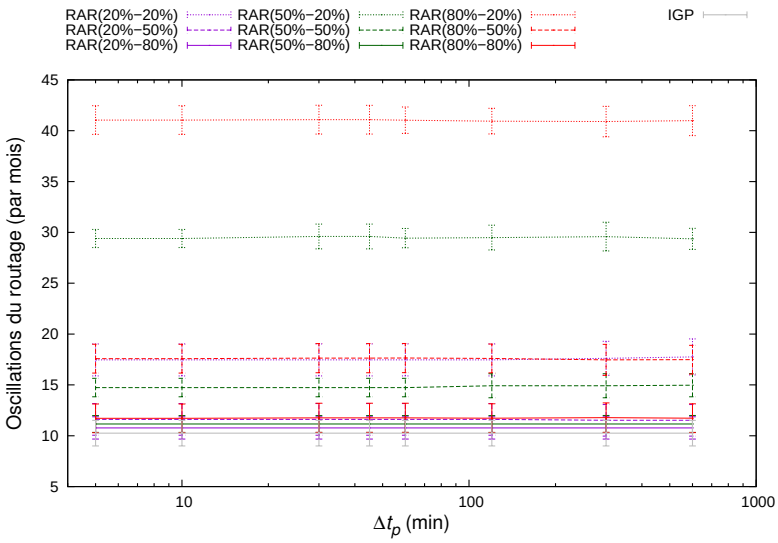


Figure 3.33: Impact de Δt_p sur le nombre d'oscillations du routage avec la topologie européenne.

3.9.2.4 Les enseignements apportés par la simulation

L'utilisation du simulateur NS-3 permet une certaine confiance dans l'implémentation des mécanismes utilisés pendant cette simulation. La simulation a fourni des résultats avec un intervalle de confiance de 99% permettant une analyse précise.

De manière générale, les résultats de simulations attestent de la validité de notre modèle analytique, sauf dans les cas où la congestion apparaît. L'interprétation de ces résultats confirme l'intérêt du dispositif RAR afin d'améliorer la disponibilité au moindre coût tout en conservant un routage stable.

Néanmoins, le faible contrôle possible sur le routage sans connexion ne permet pas de se prémunir contre le phénomène de congestion. Heureusement, les réseaux sont dimensionnés pour absorber les changements de routage consécutifs à une panne, ce qui permet au mécanisme RAR de fonctionner sans congestion dans la majorité des cas. L'étude du comportement des simulations met en évidence le mauvais comportement du mécanisme RAR utilisant une *Precision* de 20% avec des MTBF de 1000 heures ou une utilisation d'un Δt_p supérieur à deux heures. L'utilisation d'une *Precision* faible est possible mais seulement avec un nombre limité de prédictions, c'est-à-dire avec un *Recall* inférieur ou égale à 50%. À condition d'utiliser une *Precision* d'au moins 50% et un Δt_p inférieur à 2 heures, l'opérateur a tout intérêt à utiliser une prédiction de pannes avec un *Recall* important afin d'améliorer la disponibilité des services réseaux, jusqu'à un facteur cinq avec un *Recall* de 80%.

3.10 Expérimentation en environnement réel

Afin de tester la faisabilité du dispositif RAR, la fonctionnalité a été implémentée dans un prototype expérimental [KAB+11]. Le but de ce prototype est de montrer dans un environnement non simulé que l'affectation dynamique des liens permet effectivement de préserver le trafic des pertes qu'engendre une panne, et que les changements de routes sont bien transparents pour l'utilisateur notamment en terme de QoS. Dans un souci de cohérence avec l'étude analytique et les simulations, nous ne considèrerons dans ce prototype que des cas de pannes d'un routeur dans son ensemble.

3.10.1 Implémentation d'un prototype

3.10.1.1 Choix d'implémentation

L'implémentation du prototype utilise deux composants principaux : l'implémentation *open-source* du protocole de routage OSPF Quagga [Qua11] et le *framework* Java de développement

d'agent Ginkgo [Ger10] détaillé à la Sec. 1.2.9.

L'implémentation utilise le langage de programmation Java associé au *framework* Ginkgo. Elle est conçue en suivant l'architecture fonctionnelle définie au Chap. 1, c'est-à-dire avec deux modules comme illustré par la Fig. 3.1. Un premier bloc implémente le module RAM du R&S_DE tandis que le deuxième est en charge des fonctions d'affectation des métriques du module RAR intégré au RM_DE. La plateforme multi-agent Ginkgo est utilisée en tant que *framework* pour la structure de l'implémentation des deux éléments de décisions (le R&S_DE et le RM_DE) et aussi pour les mécanismes de communication inter-agent très utiles pour la diffusion de l'information de risque.

La communication entre le RAM et le module RAR du RM_DE est réalisé au travers de l'implémentation du plan de connaissances de Ginkgo Networks mais pourrait aussi utilisé d'autres procédés tels que les mécanismes de communication interprocessus (IPC [1]), à condition qu'ils minimisent les délais de transmission qui pourraient être préjudiciables à l'achèvement du reroutage des flux de trafic avant l'apparition de la panne.

De même, le moteur d'affféctation des métriques a besoin de communiquer avec le démon OSPF afin de modifier les métriques des liens. Cette communication peut se faire en utilisant un protocole de gestion tel que Netconf [Enn06] ou SNMP [CFSD90] mais peut aussi exploiter des mécanismes propriétaires tels que les interfaces en ligne de commande CLI [2] ou les API [3] spécifiques. Contraint par les fonctionnalités de notre implémentation d'OSPF, nous avons choisi d'utiliser la CLI de Quagga qui permet de déclencher de manière instantanée les LSA de modification des métriques et ainsi de répondre correctement à nos besoins en réactivité.

3.10.1.2 Environnement de la plate-forme d'essai

Pour des raisons pédagogiques, nous avons choisi d'implémenter un cas d'utilisation simple et très parlant, à savoir le risque de panne dû à une surchauffe. L'analyse du risque de cette panne sur un routeur exploite des informations locales, c'est la raison pour laquelle l'implémentation du RAM se fait par l'intermédiaire d'un unique processus accédant aux diverses sondes du système d'exploitation qui sont, dans notre cas, dans le répertoire « */proc/acpi/* » du système Linux. Afin de contrôler ce comportement, nous avons utilisé les outils du *framework* Ginkgo et créé trois agents additionnels sur chaque routeur. Le contrôle s'effectue au travers d'une interface graphique (voir Sec. D) qui contrôle chaque agent avec le protocole de *webservices* SOAP [4]. Le premier agent est responsable de l'observation de la température et de l'émulation d'une panne (*i.e.* la désactivation des interfaces réseaux) lorsque la température atteint les 100˚C. Ce comportement est normalement intégré au niveau du BIOS [5] qui déclenche un redémarrage dès que la température dépasse la limite acceptée, nous avons choisi de reproduire ce comportement au niveau logiciel afin de ne pas endommager notre équipement avec une vraie surchauffe. Pour cela, nous émulons une panne de manière logicielle (puis le retour à la normale) en activant (resp. désactivant) les interfaces réseaux avec *ifconfig*. Cet agent est aussi en charge d'affecter une température spécifique envoyée par l'utilisateur par l'intermédiaire de l'interface de contrôle. Le deuxième agent a pour fonction d'émuler une augmentation linéaire de la température de un 1˚C par seconde jusqu'à 100˚C. À ce moment-là, une panne est déclenchée par l'agent précédemment cité. De manière symétrique, le dernier agent a pour fonction la baisse de la température de 1˚C par seconde jusqu'à la température stationnaire de fonctionnement fixé à 65˚C.

L'implémentation du routage RAR composé de l'implémentation standard du protocole OSPF par Quagga et de notre implémentation Java du dispositif RAR, ainsi que notre implémentation du scénario de panne par surchauffe a été déployée sur un ensemble de huit routeurs Linux. Pour des raisons pratiques, ces routeurs linux utilisent la technologie de virtualisation VMware [Sof11] afin d'être instanciés sur seulement 3 machines physiques. Chaque machine physique est constituée de deux processeurs Intel Xeon double cœur cadencés à 2 GHz, de 8 Go

1. *Inter-Process Communication*
2. *Command Line Interface*
3. *Application Programming Interface*
4. *Simple Object Access Protocol*
5. *Basic Input Output System*

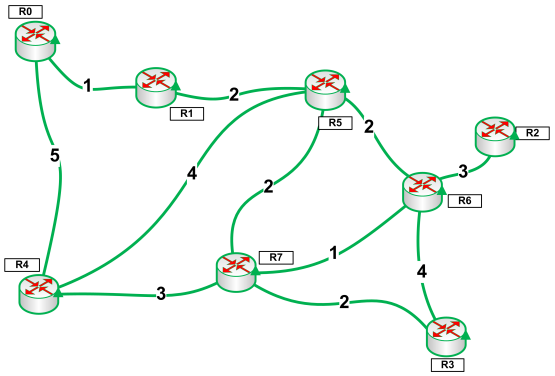


Figure 3.34: Topologie réseau de la plate-forme d'essai.

de mémoire vive et de 4 interfaces réseaux Ethernet à 100 Mbps avec le système d'exploitation *Linux Ubuntu x64 server 9.04* [Lin11c] et le logiciel de virtualisation *VMware Server x64 2.0.2* [Sof11]. Chacune de ces machines physiques a pour fonction l'exécution de plusieurs machines virtuelles. Ces machines virtuelles sont des systèmes Linux *Ubuntu server x32 9.04* avec 512 Mo de mémoire vive sur lesquelles est déployé le logiciel de routage Quagga [Qua11] version 0.99.16 accompagné de notre implémentation du mécanisme RAR. Enfin, la technologie de virtualisation ne permettant pas de définir avec précision les capacités des interfaces virtualisées, nous avons utilisé le programme *TC* [Lin11b] sur chaque routeur Linux afin de définir avec précision la capacité de chaque interface et ainsi réaliser avec précision la topologie affichée sur la Fig. 3.34.

3.10.2 Mesures de performances

La construction d'un prototype de réseaux utilisant le routage sensible aux pannes à pour intérêt de montrer la faisabilité de l'implémentation réelle d'un tel mécanisme, mais permet aussi d'observer son comportement lorsqu'il est associé à une implémentation du protocole OSPF dans un environnement réel. Pour réaliser cette étude, nous avons analysé les conséquences du mécanisme RAR sur le plan de données grâce à l'interface graphique disponible en Annexe D, ce qui permet de déclencher des pannes à la demande afin de visualiser finement les conséquences sur le trafic. Afin de ne pas perturber les résultats, nous avons choisi un Δt_l fixe de dix secondes entre la prédiction de panne et l'occurrence de la panne.

Avec une configuration d'OSPF similaire à celles utilisées dans les sections précédentes, c'est-à-dire avec un *Hello Interval* de une seconde et un *Router Dead Interval* égale à $4 * HelloInterval$, nous avons étudié l'impact d'une panne sur un flux de trafic constant. Pour cela, nous avons utilisé le programme *Iperf* [Lin01] afin d'envoyer un flux de trafic constant de 1 Mbit/s entre R0 et R6 pendant 20s en utilisant le protocole UDP [1]. Si aucune panne ne se produit, ce flux correspond à l'envoi de 1702 paquets IP. Afin d'analyser les conséquences du dispositif RAR sur le plan de données, nous avons déclenché une panne de surchauffe sur le routeur R1 et observé le nombre de paquets reçus avec et sans l'activation du mécanisme RAR. La Fig. 3.35 affiche le nombre de paquets reçus avec chaque mécanisme pour chacun des 100 essais.

Les résultats montrent que le re-routage du mécanisme RAR permet de ne perdre aucun paquet sur les 100 essais alors que le dispositif de restauration IP implique une indisponibilité à peu près égale au temps de détection de panne t_D (*i.e.* $3.5 * t_{HI}$ en moyenne), ce qui correspond à une moyenne de 295 paquets pourl'ensemble des 100 essais.

1. *User Datagram Protocol*

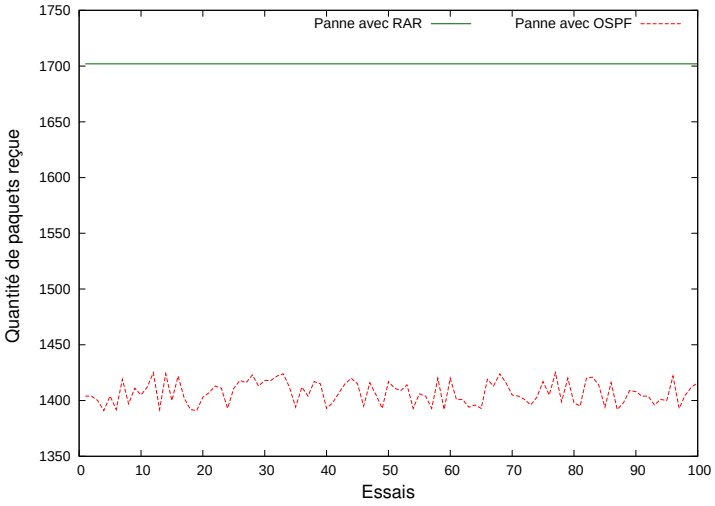


FIGURE 3.35: Comparaison de l'effet d'une panne pendant l'envoi d'un trafic constant de 1 Mbit/s pendant 20s (1702 paquets) sur 100 essais.

La capture de trafic de la Fig. 3.36 montre de plus près l'effet d'une panne sur un flux de trafic constant sans l'utilisation du RAR. A partir du premier carré rouge, qui indique l'occurrence de la panne, on voit distinctement une interruption de trafic jusqu'au carré vert, indiquant le premier message de mise à jour de la topologie (message *LS Update*). Le trafic utilise ensuite le chemin de restauration jusqu'à ce que routeur R1 soit de nouveau opérationnel. À partir de ce moment-là, le carré bleu indique le premier message *LS Update* contenant l'annonce du retour du routeur R1 dans la topologie.

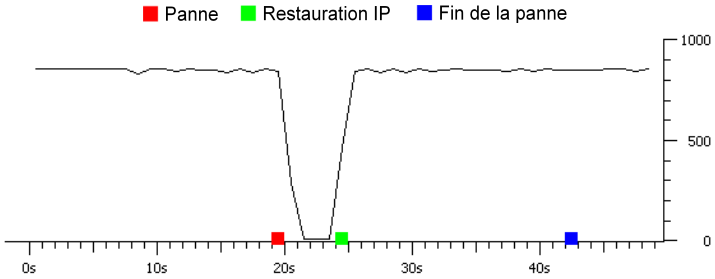


FIGURE 3.36: Effet d'une panne sans le routage anticipant les pannes sur un flux de trafic constant.

Avec le mécanisme RAR, le comportement est bien différent comme en atteste la Fig. 3.37. À partir de la dixième seconde, on remarque la détection d'un risque de panne. C'est à ce moment-là que le module RAR modifie les métriques des liens et que le re-routage préventif se met en place grâce à l'envoi des messages *LS Update*. Comme le souligne la Fig. 3.37, cette opération est transparente au niveau du récepteur, ce qui est conforme aux considérations théoriques et au comportement du mécanisme dans le simulateur. En effet, cela confirme que le nombre de fausses prédictions (et notamment le *Recall*) n'ont pas d'incidence direct sur l'indisponibilité du trafic. On observe ensuite que, ni la panne indiquée par le carré rouge, ni la détection de la panne signalée par le carré vert, ni même la fin de la panne, ne vienne perturber le trafic.

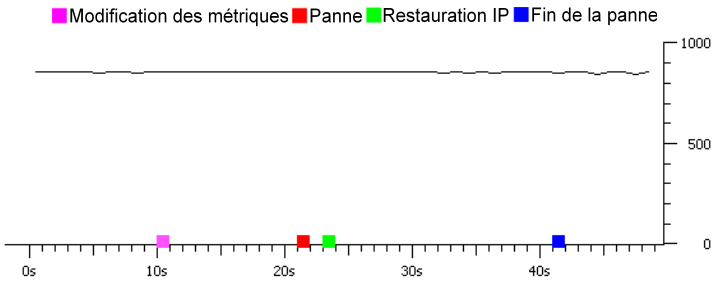


Figure 3.37: Effet d'une panne avec le routage anticipant les pannes sur un flux de trafic constant.

Nous avons également effectué la même analyse sur du trafic vidéo. Pour cela, nous avons mis en place un flux vidéo HD [1] utilisant le protocole UDP entre un serveur et un client utilisant le logiciel Videolan [VLC11]. Cette expérience permet dans un premier temps d'observer le gain en QoE [2] puisque, sans mécanisme RAR, la vidéo se fige quelques secondes pendant la panne, alors qu'avec l'utilisation du dispositif RAR, la panne est totalement transparente pour le téléspectateur. La Fig. 3.38 met en évidence l'interruption de la vidéo pendant toute la durée de la panne avec le fonctionnement standard d'OSPF.

Figure 3.38: Effet d'une panne sans le routage anticipant les pannes sur un flux de trafic vidéo HD.

À l'inverse, lorsqu'une panne est prédite à l'avance, le mécanisme RAR permet au flux vidéo HD d'éviter de ressentir les effets de la panne comme l'illustre la Fig. 3.39.

Figure 3.39: Effet d'une panne avec le routage anticipant les pannes sur un flux de trafic vidéo HD.

Nous avons expérimenté l'incidence d'une panne sur des routeurs différents, avec des flux ayant des chemins différents et observé le même résultat. Au delà de démontrer la faisabilité de l'implémentation du mécanisme RAR dans un environnement réelle, le prototype illustre la capacité du mécanisme RAR pour forcer le trafic à emprunter des chemins dépourvus de routeurs risqués, ce qui permet de protéger le trafic contre les interruptions de service. Ensuite, les tests réalisés sur cette plate-forme confirment que le re-routage n'introduit pas de perte ce qui est de bonne augure par rapport aux fausses prédictions. Enfin, le gain en disponibilité observé est

1. Haute Définition
2. *Quality of Experience*

conforme aux considérations émises à la Sec. 3.6, c'est-à-dire de l'ordre de t_C pour chaque panne prédite à l'avance.

3.11 Conclusion

Le dispositif de routage sensible aux pannes (RAR) développé dans ce chapitre permet, grâce à une information de risque de panne, de dégager une petite fenêtre de temps avant la panne pour forcer les flux de trafic à contourner cette panne. Le routage est préventivement reconfiguré pour atteindre une situation proche de ce qui sera effectif après l'apparition de la panne en affectant aux liens risqués une métrique suffisamment grande pour que ce lien ne soit pas utilisé par le trafic. Cette opération permet de supprimer complètement l'interruption de service induite par la panne lorsque la prédiction de panne se confirme. Le principe de ce mécanisme repose sur le fait que les réseaux étant dimensionnés pour supporter plusieurs pannes, un tel reroutage est facilement supporté par le réseau lorsque le nombre de prédictions de panne reste raisonnable. Par ailleurs les changements de routage résultant de prédiction de pannes peuvent introduire de l'instabilité lorsque ceux-ci sont trop nombreux. Il a donc été nécessaire de prendre en compte cette donnée dans l'évaluation de ce mécanisme. Pour cela, un modèle analytique est proposé pour quantifier les gains théoriques du dispositif RAR en terme d'indisponibilité, ainsi que le nombre d'oscillations de routage engendrées. Une implémentation dans un simulateur à événements discrets a permis de vérifier le bien-fondé de notre modélisation d'une part, et d'observer l'incidence du mécanisme sur la congestion d'autre part. Enfin, l'implémentation d'un prototype expérimental a été réalisée afin de vérifier la faisabilité de la fonction. Les enseignements de cette étude montre un réel gain, proportionnel au nombre de pannes prédites, tout en contenant le nombre d'oscillations du routage dans des proportions assurant une stabilité pour le réseau. Il convient cependant de noter que l'utilisation d'une faible *Precision* (*i.e.* 20%) avec un taux de panne élevé (*i.e.* MTTR $<$ 3000 heures) peuvent générer un nombre d'oscillations qui serait problématique pour des réseaux nécessitant une stabilité très importante. Le mécanisme s'appliquant dans une configuration de routage IP pure, le principal inconvénient vient surtout de l'apparition de congestions qui annihilent tous les bénéfices du mécanisme en générant plus de pertes que le mécanisme de restauration IP. Ce phénomène est présent lorsque le taux de pannes est élevé, mais surtout lorsque le nombre de fausses prédictions est très important et/ou que la durée de prédiction est de plusieurs heures. Ce phénomène plus ou moins accentué suivant la topologie amène à contraindre l'utilisation de ce mécanisme avec une prédiction de pannes ne générant pas trop de fausses prédictions et avec une durée de prédiction de panne inférieure à deux heures. Le dispositif de routage sensible aux pannes permet un gain supérieur au mécanisme AFDT, mais n'est pas utilisable dans autant de cas, avec notamment des contraintes fortes sur les performances de la prédiction de pannes utilisée dans le RAM.

Les problèmes de congestion peuvent être outrepassées en appliquant ce mécanisme au protocole GMPLS, mais ce protocole possède d'autres dispositifs de résilience qui permettent d'envisager un autre type de mécanisme d'autoréparation proactif tel que celui proposé dans le prochain chapitre.

Chapitre 4

Un nouveau mécanisme de résilience adaptatif

Sommaire

4.1 Introduction

Dans la compétition que se livrent les opérateurs de télécommunications la Qualité de service (QoS) et la disponibilité promises à leurs clients dans le SLA sont des préoccupations importantes pour les opérateurs. Il en résulte un investissent important aussi bien en CAPEX qu'en OPEX pour la gestion des pannes [MMJ08]. Bien que la technologie de cœur de réseau la plus aboutie contienne une série complète de mécanismes dédiés aux pannes au travers des dispositifs de protection et de restauration du protocole GMPLS [1] [MP06], leur nature statique et réactive oblige à choisir entre le déploiement de ressources de protection qui seront inutilisées la plupart du temps, ou l'activation de mécanismes de restauration aux performances médiocres. La stratégie de gestion des pannes d'un opérateur est donc un compromis entre la consommation de ressources et disponibilité.

Pour résoudre ce dilemme, l'introduction de fonctions autonomes dans les équipements réseaux permet d'envisager une nouvelle approche combinant les bénéfices des deux grands mécanismes de résilience de GMPLS. En effet, avec la capacité et la rapidité d'analyse des équipements d'aujourd'hui, au service de la prédiction de pannes, comme décrite à la Sec. 1.6, on peut envisager un dispositif qui adapterait, en temps réel, le schéma de résilience, au risque de panne [VNC12, WTVL13, PKM+09, PKA+10, VCL+11]. Une telle fonction d'autoréparation permettrait d'économiser des ressources la majorité du temps, lorsque qu'aucune panne imminente n'est détectée et de mettre en place temporairement un mécanisme de résilience ultra rapide lorsqu'un important risque de panne est annoncé.

Dans ce chapitre nous étudions en détails la faisabilité d'un tel mécanisme au sein du protocole GMPLS, en commençant par étudier le fonctionnement des mécanismes standards de gestion des pannes ainsi que les travaux de recherche s'attachant à l'amélioration de ces mécanismes. Ensuite la logique du mécanisme proactif, son intégration dans le protocole GMPLS et son évaluation *via* une modélisation analytique et des simulations sont abordés.

4.2 (G)MPLS et la gestion des pannes

Le protocole GMPLS est une généralisation des technologies MPLS aux environnements optiques. Ces protocoles permettent d'acheminer des flots de trafic au travers du réseau en créant des tunnels basés sur l'utilisation de *labels*.

4.2.1 MPLS

Le fonctionnement de MPLS [2], défini dans par l'IETF dans de nombreuses RFCs, est synthétisé dans la RFC3031 [RVC01]. Les fondements de ce protocole reposent sur l'établissement de chemins que l'on nomme LSP [3], permettant l'acheminement de paquets IP entre deux routeurs du réseau. Ces chemins utilisent une signalisation de couche 3 (IP) afin de pouvoir relier n'importe quels routeurs de l'Internet. L'architecture MPLS fournit un ensemble d'outils effectuant les opérations de routage et de signalisation pour former un plan de contrôle indépendant du plan de données. Le transfert des paquets au sein de chaque routeur n'est plus géré au niveau IP mais au niveau 2.5 (*i.e.* MPLS) suivant un chemin prédéfini et préconfiguré avec des *labels*. Chaque LSR [4] possède une table de commutations associant un LSP à un *label* d'entrée et de sortie. Ces LSR intermédiaires n'ont plus qu'une fonction de commutation, le choix du routage étant effectué lors de l'étape de signalisation. C'est le routeur d'entrée (*Ingress router*) dans le domaine MPLS qui décide du LSP vers lequel envoyer chaque paquet. Pour ce faire, MPLS repose sur un protocole de routage tel que OSPF [Moy98] ou IS-IS [Ora90] et un protocole de signalisation tel que RSVP [HL97] ou LDP [5] [ADF+01].

1. *Generalized MultiProtocol Label Switching*
2. *MultiProtocol Label Switching*
3. *Label Switched Path*
4. *Label Switched Router*
5. *Label Distribution Protocol*

Les chapitres précédants ont présenté le fonctionnement global de ces protocoles de routage de type IGP, mais l'introduction de l'ingénierie de trafic dans MPLS a abouti à la mise à jour de ces protocoles afin de prendre en compte la charge des liens dans le choix du routage. L'ingénierie de trafic est une des avancées majeures de MPLS. Elle permet d'éviter d'envoyer trop de trafic dans les zones engorgées ce qui a une incidence bénéfique sur la QoE ressentie par les utilisateurs. Pour cela, l'IETF a défini une extension pour chaque protocole avec OSPT-TE [1] [KKY03] et IS-IS-TE [2] [SL04].

Afin de distribuer correctement les *labels* le long d'un chemin, un protocole de signalisation est nécessaire. LDP [ADF+01] définit les mécanismes de signalisation permettant de configurer saut par saut, un chemin, en utilisant le routage IP. C'est le routeur d'entrée qui déclenche la session LDP mais c'est ensuite chaque routeur IP qui décide du prochain saut, et ce, jusqu'à ce que le routeur de destination soit atteint.

MPLS permet aussi de fonctionner avec du routage explicite. Dans ce mode, le LSR de bordure définit tout ou partie du chemin lors du déclenchement de la signalisation. Cela permet de choisir plus finement le chemin emprunté par le LSP afin d'appliquer des mécanismes d'ingénierie de trafic. Il est ainsi possible d'emprunter un autre chemin que le plus court chemin, en utilisant des informations sur la bande passante disponible pour répartir le trafic équitablement sur le réseau. Afin de spécifier le chemin d'un ER-LSP [3], il est nécessaire d'utiliser les protocoles de signalisation CR-LDP [4] pu RSVP :

- CR-LDP [ALAS+02] est une version améliorée de LDP permettant de spécifier la bande passante et la route explicite d'un LSP ;
- le protocole RSVP [HL97] permet de prendre en compte la capacité d'un lien mais ne permet pas de gérer la réservation de la bande passante, ni de spécifier une route explicite. Ces améliorations sont néanmoins disponibles avec l'extension RSVP-TE [5].

Le protocole RSVP-TE [ABG+01] est le protocole de signalisation le plus répandu permettant de faire de l'ingénierie de trafic. Il permet de connaître la bande passante des liens, la bande passante utilisée, ainsi que la bande passante disponible. Il possède aussi la possibilité de spécifier à la source, le chemin explicite du LSP par l'intermédiaire d'un ERO [6].

MPLS a connu un succès important, en apportant au routage IP un plan de contrôle complet, permettant une amélioration des performances et une plus grande maîtrise de l'ingénierie de trafic. Fort de ce succès, les principes de MPLS ont été généralisés aux autres technologies de commutation avec le protocole GMPLS.

4.2.2 GMPLS

Le protocole GMPLS [Man04] applique les principes de MPLS aux technologies de commutation par paquets, tranches de temps, longueurs d'ondes et fibres optiques. Il fournit un plan de contrôle unifié permettant le provisionnement de connexions de bout en bout afin de fournir un haut niveau de QoS. Grâce à GMPLS il est possible de créer un LSP à travers n'importe quelle combinaison de réseaux de paquets, de multiplexages temporels, et de multiplexages optiques. Un aspect important du plan de contrôle GMPLS est sa gestion des pannes. En effet, la coupure d'une seule fibre DWDM [7] peut engendrer des pertes de données importantes. Il est donc indispensable de doter le protocole GMPLS d'outils performants de gestion des pannes. Ces outils sont disponibles sous la forme de deux grands types de dispositifs de résilience, les mécanismes de protection et les mécanismes de restauration.

1. *Open Shortest Path First-Traffic Engineering*
2. *Intermediate System to Intermediate System - Traffic Engineering*
3. *Explicitly Routed Label Switched Path*
4. *Constraint-Based Routing Label Distribution Protocol*
5. *Resource Reservation Protocol - Traffic Engineering*
6. *Explicit Route Object*
7. *Dense Wavelength-Division Multiplexing*

4.2.3 Protection et restauration

Les dispositifs de gestion des pannes de GMPLS ou de MPLS peuvent être classés en deux catégories : la protection (notée P [1]) et la restauration (notée R [2]). Avec la protection, un ensemble de liens et de nœuds de redondance sont présélectionnés et leurs ressources associées sont réservées. En conséquence, lors d'une panne sur le chemin principal, le trafic peut être immédiatement orienté sur le chemin de protection. Le cas de la restauration est différent puisque le chemin de protection n'est établi qu'après l'occurrence de la panne. Les deux mécanismes sont différents en termes d'échelle de temps et d'utilisation de ressources. Les mécanismes de protections et de restaurations sont essentiels afin de délivrer une QoS adéquate aux utilisateurs finaux et pour assurer les besoins de disponibilités obligatoires pour les réseaux d'opérateurs. Le protocole GMPLS a défini l'ensemble de ces dispositifs de résilience dans la RFC4427 [MP06].

La protection (P) nécessite l'allocation de ressources redondantes, généralement une redondance de 100% afin de pouvoir réagir de manière extrêmement rapide. Par exemple, le protocole SDH est conçu pour réorienter un flux de trafic d'un chemin primaire à un chemin secondaire en moins de 50 millisecondes avec une configuration de protection 1+1. Cette approche consomme deux fois plus de ressources qu'une configuration non protégée.

L'alternative de la restauration (R) s'appuie sur une mise en œuvre dynamique des ressources secondaires ce qui implique un délai de restauration d'une connexion d'un ordre de grandeur plus élevé. La restauration peut aussi demander un calcul de route dynamique, coûteux en temps de calcul, si les chemins de redondance ne sont pas calculés à l'avance ou si les chemins pré-calculés ne sont plus disponibles.

La protection et la restauration sont traditionnellement subdivisées en deux techniques : soit appliquées au chemin de la connexion, soit appliquées à un sous-chemin ou à un lien. Dans la résilience appliquée au chemin, une panne est gérée au niveau de la source et de la destination du LSP, alors que la résilience appliquée aux sous-chemins est gérée par les routeurs intermédiaires qui détectent la panne. La Fig. 4.1 montre les différentes configurations de résilience de GMPLS où ces deux techniques sont utilisées aussi bien par la protection que la restauration.

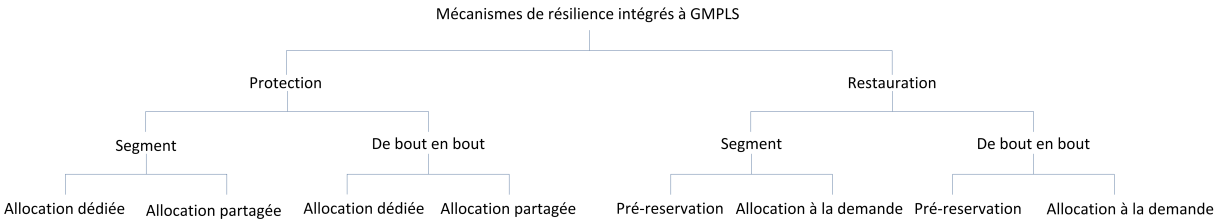


FIGURE 4.1: Modes de protection et de restauration de GMPLS.

Dans tous ces mécanismes de résilience réactifs, plusieurs étapes sont impliquées après la détection d'une panne : le calcul de route, la signalisation, et la *cross-connection*. La différence entre les niveaux de résilience disponibles vient de la réalisation d'une ou plusieurs de ces étapes lors la configuration du service, c'est-à-dire avant l'occurrence de la panne.

4.2.3.1 Protection

La protection possède plusieurs configurations avec la protection dédiée (*1+1*), la protection dédiée avec du trafic externe (*1 :1*), la protection partagée avec du trafic externe (*1 :N*) et la protection multi-chemin partagée avec du trafic externe (*M :N*).

Dans le cas de la protection *1+1*, le trafic est envoyé simultanément sur deux chemins disjoints, un chemin principal (*working*) et un chemin de protection, c'est ensuite la destination qui sélectionne le meilleur signal. Dans le cas d'une panne, le routeur de destination ne reçoit plus de données du LSP principal et utilise donc le LSP de protection pour réceptionner les donnés. Cela nécessite d'utiliser le double de la bande passante du LSP mais permet le rétablissement de la connexion en un temps minimal.

1. Protection (G)MPLS
2. Restauration (G)MPLS

La protection *1 :1* est aussi un mode très coûteux. Un LSP de protection associé au LSP principal est réservé mais le trafic n'est envoyé que sur chemin principal. Les ressources de protection ne peuvent pas être réservées pour un autre LSP de même importance mais elles sont disponibles pour du trafic de moindre priorité tel que le *Best Effort*. Lors d'une panne, le nœud source doit effectuer une étape légère de signalisation afin de basculer le trafic sur le chemin de protection pré-réservé.

La protection *1 :N* diffère de la protection 1 :1 en ce qu'un chemin de protection pré-alloué est associé à plusieurs LSP principaux (N). Il n'en résulte qu'un risque lors d'une panne si le chemin de protection est déjà utilisé par un autre LSP, ceci entraîne une indisponibilité importante. De plus, le basculement du trafic implique des actions supplémentaires par rapport au schéma 1 :1 qui allongent légèrement le délai de rétablissement du LSP.

Enfin le dernier schéma *M :N* est un cas ou N chemins principaux sont protégés par M chemins secondaires. C'est une configuration qui englobe les configurations *1 :1* et *1 :N* et où les M chemins de redondance sont partagés entre N connexions. De manière similaires aux cas *1 :1* et *1 :N* les ressources sont pré-allouées, mais elles sont utilisables par du trafic moins prioritaire. Lors d'une panne, une action de signalisation est nécessaire afin de régler chaque routeur de protection au basculement du trafic du LSP perturbé par la panne.

4.2.3.2 Restauration

La restauration est conçue pour réagir aux pannes et être efficace en terme d'utilisation de bande passante mais avec un calcul de routes et une réservation des ressources dynamique, ce qui est moins rapide que les techniques de protection. Il existe trois schémas de restauration qui sont le reroutage complet (*full-rerouting*), la restauration pré-calculée (*pre-computed restoration*) et la restauration pré-planifiée (*pre-planned restoration*).

Le mécanisme de restauration, aussi appelé reroutage, utilise le concept de *Make-Before-Break* [ABG+01]. Le MBB [1] permet à un ancien chemin de toujours être utilisé pendant la mise en place du nouveau chemin, et ce, sans devoir réserver les ressources en double. Une fois la configuration terminée, le nœud effectuant le reroutage bascule le trafic sur le nouveau chemin puis libère les ressources de l'ancien chemin. Cette fonctionnalité est disponible avec le protocole RSVP-TE grâce à l'option *Shared Explicit* [ABG+01] mais aussi avec le protocole CR-LDP.

Lors d'une restauration, le chemin secondaire est établi après trois étapes : le calcul de routes, la signalisation et la réservation des ressources. Le reroutage complet est le cas le plus dynamique où ces trois étapes sont déclenchées après la détection de la panne. La restauration pré-calculée est plus rapide puisque le calcul de routes a déjà été effectué ; il ne reste plus que la signalisation et la réservation de ressources à compléter au moment de la panne. Enfin, la restauration pré-planifiée signifie que la sélection de la route ainsi que la signalisation ont été effectuées à l'avance, il ne reste donc plus que la réservation de ressources.

En conséquence, les performances en terme d'indisponibilité de chaque mécanisme dépendent des étapes restantes à effectuer pour établir un chemin de secours après l'occurrence de la panne.

4.3 Problématique

Le protocole GMPLS possède deux catégories de dispositifs de résilience dont chacun possède des atouts. La protection permet une interruption de service minimum au détriment de la consommation de ressources, alors que la restauration, grâce à sa nature dynamique, est très efficace dans l'utilisation des ressources mais en assurant une disponibilité moins élevée. En effet, la restauration n'est mise en place dynamiquement que lors d'une panne, ce qui en fait un mécanisme efficace en terme d'utilisation de ressources, mais cette mise en place dynamique met du temps pour établir le nouveau LSP engendrant une interruption de service importante. La protection est bien plus rapide pour rétablir le service puisqu'un LSP de protection est déjà réservé. Mais cette réservation nécessite l'utilisation de deux fois plus de ressources, alors même

1. *Make-Before-Break*

que les ressources de protection ne sont utilisées que dans de rares occasions. Le mécanisme de protection est très coûteux pour l'opérateur.

Le choix de la technique de résilience à utiliser pour un opérateur est difficile. Ce choix est basé sur le type de trafic à protéger, son importance, ses besoins en QoS mais aussi sur la fiabilité des équipements traversés par le flux. Ce paramètre est difficile à prendre en compte puisque un même chemin peu posséder des tronçons d'anciens équipements ayant une fiabilité faible ainsi que des parties composées d'équipements de dernière génération avec une meilleure fiabilité. L'opérateur doit donc considérer le pire cas, ce qui peut l'entraîner à opter pour un schéma de protection qui lui sera très coûteux. De plus, la nature statique de la gestion des pannes accentue le problème. En effet, le type de mécanisme de résilience à appliquer à un LSP est configuré lors de la mise en place du service et rarement remis en cause par la suite. Les statistiques de fiabilité des équipements utilisés pour le choix du type de protection sont des statistiques très long terme, qui représentent plus une tendance qu'une donnée précise et qui peuvent ne plus être valides dans le futur. La complexité du choix de la bonne méthode à utiliser, ainsi que la lourdeur de sa mise en place contraint les opérateurs à ne pas remettre en cause leur configuration de gestion des pannes initiales, même si celle-ci n'est peut-être plus adaptée à la situation.

Enfin, la probabilité de panne des équipements est une donnée qui peut être extrêmement volatile, la mise à jour d'un système d'exploitation de certains routeurs peut engendrer un nombre important de pannes pendant une courte période, le dysfonctionnement d'un système de refroidissement peut augmenter de manière significative la probabilité de panne de certains équipements pendant une période de temps fini. Des événements réseaux comme les LSA *storms*, la propagation d'un virus, ou une attaque de pirates informatiques ont aussi un impact sur la probabilité de panne d'un équipement.

La nature statique de l'utilisation de la résilience GMPLS oblige les opérateurs à utiliser des statistiques longs termes, laissant de coté l'aspect dynamique de l'évolution de la probabilité de panne au court du temps. La conséquence de cela se manifeste par une consommation de ressources très coûteuse de l'utilisation des techniques de protection, alors que cela n'est pas nécessaire la plupart du temps. Cependant, afin de fournir une disponibilité importante, la nature statique de la résilience contraint les opérateurs à surdimensionner la protection afin de minimiser leur risque d'interruption de service. L'autre choix peut-être d'utiliser la restauration qui est beaucoup moins coûteuse, mais qui fournit des performances de disponibilité moindre qui ne sont pas suffisantes pour assurer la QoS nécessaire au trafic dit *premium*.

Une protection trop coûteuse ou une restauration peu performante sont actuellement les deux seules alternatives qui s'offrent aux opérateurs.

4.4 Une protection moins coûteuse et une restauration plus rapide

Malgré l'existence d'un ensemble complet de dispositifs de résilience au sein du protocole GMPLS, l'utilisation à bon escient du mécanisme adapté est un problème en soi. Le choix de la stratégie à adopter est un problème pour de nombreux opérateurs auxquels la communauté scientifique a tenté de répondre [CLSS02]. Ce n'est pas un problème simple puisque chaque technique possède des avantages et des inconvénients. La protection permet une interruption minimum mais est très coûteuse alors que la restauration est peu onéreuse mais engendre une indisponibilité importante.

Des propositions ont été faites afin d'améliorer l'efficacité des mécanismes de résilience, soit en améliorant le coût des dispositifs de protection, soit d'une manière plus commune, en diminuant le temps de rétablissement du service avec le mécanisme peu coûteux de la restauration.

La protection a pour principal inconvénient de nécessiter la réservation des ressources pour les LSP de protection, ce qui est très coûteux en CAPEX comme en OPEX. Pour réduire ce défaut, le concept de préemption permet à plusieurs LSP de réserver une même ressource, un ordre de priorité permettant d'arbitrer l'utilisation de la ressource lorsque plusieurs LSP veulent l'utiliser en même temps. Cette possibilité augmente le temps de rétablissement d'un service

car il est désormais nécessaire de signaler le choix d'utilisation de la ressource partagée, mais cela reste plus performant que la restauration et permet de diminuer les coûts. Néanmoins, l'utilisation de cette option est complexe. Il est difficile pour un opérateur de maîtriser entièrement son comportement et d'en prévoir les conséquences. En réponse, de nombreux travaux [Gro04, LGS02, CTS03, GAVC05, SFT+06] ont proposé des approches pour partager efficacement les ressources de protection et ainsi de faire baisser le coût de cette technique. Mais la restauration reste toujours plus intéressante en terme de coût et possède une nature dynamique qui la rend plus robuste aux multiples pannes.

L'autre alternative est l'amélioration des performances de la restauration. Cette option a été de loin la plus étudiée. Comme vu aux Sec. 2.4 et 3.4, de nombreuses propositions ont été faites pour réduire les différente étapes de la convergence des protocoles IGP [FFEB05] afin de rendre viable l'utilisation de la restauration pour les réseaux à haute disponibilité [SRM02, PIKF04]. De plus, en ce qui concerne l'utilisation de la technologie de commutation de paquets, il existe de manière similaire à la restauration IP, des mécanismes permettant d'outrepasser le délai de convergence par des décisions locales mettant en place des chemins temporaires [TWFV06]. Le *MPLS Fast ReRoute* (MPLS FRR) [RI07, PSA05, VPD04] comme le IP FRR possède les techniques de *Loop Free Alternate* (LFA), de *U turn alternate*, de *Not-via address* et de *tunneling*. De nombreux travaux ont proposé des modalités d'amélioration de la mise en œuvre du MPLS FRR [RI07, WWM+10] mais cette solution ne permet pas d'assurer une disponibilité comparable à la protection et génère des topologies non optimales avec des chemins plus longs, des délais d'acheminement plus longs et de possibles boucles de routage.

En dépit des évolutions améliorant les deux catégories de mécanismes de résilience, et au regard de la nature statique des stratégies de gestion des pannes actuelles, le choix de configuration est toujours un compromis entre vitesse et consommation de ressources. En réponse, nous proposons une approche d'autoréparation complètement différente qui utilise l'estimation en temps réel du risque de pannes pour alterner dynamiquement entre la protection et la restauration afin de minimiser les coûts consacrés à la gestion des pannes tout en maintenant une disponibilité importante.

4.5 Description de la proposition

Le principal défaut des dispositifs de résilience de GMPLS est leur nature statique alors que la probabilité de panne évolue dans le temps. Avec des modules RAM capables de détecter en temps réel lorsque le risque de panne devient important, il devient possible d'envisager d'utiliser cette information pour modifier dynamiquement la gestion des pannes. Une telle intervention nécessite qu'elle soit effectuée en quelques secondes maximum ce que n'est pas capable de réaliser un opérateur humain. C'est pourquoi le principe de réseau autonome est le candidat idéal pour implémenter une telle fonctionnalité au sein des équipements réseaux. L'utilisation d'une information de risques fourni par le RAM permet au réseau d'adapter de manière extrêmement rapide son mécanisme de gestion des pannes au risque de pannes en temps réel et ainsi d'obtenir une meilleure efficacité. Notre proposition a pour but de fournir une protection minimum lorsque le risque de panne est faible afin de limiter le coût de la gestion des pannes, et une protection maximum lorsqu'un risque de panne est détecté, afin de minimiser le délai de rétablissement du LSP. Le mécanisme ALR [1] repose sur l'utilisation de la restauration en temps normal, et de l'établissement de LSP de protection temporaire lorsqu'un élément du LSP est concerné par une prédiction de panne [VNC12, WTVL13, PKM+09, PKA+10, VCL+11]. On obtient donc pour un coût proche de la restauration, de meilleures performances en termes de disponibilité.

4.5.1 Aperçu général du principe de résilience dynamique

L'objectif du dispositif ALR est d'améliorer les performances de la résilience de GMPLS en introduisant la prédiction de pannes au sein des équipements de réseau. Le mécanisme joue sur l'augmentation temporaire du niveau de protection d'un LSP possédant un routeur risqué afin

1. *Adaptive Level of Recovery*

d'être prêt si une panne devait effectivement apparaître [VNC12, WTVL13, PKM+09, PKA+10, VCL+11]. Le reste du temps, quand le risque de panne est faible, le LSP peut rester moins protégé ce qui permet à l'opérateur d'avoir plus de ressources disponibles.

L'objectif de cette proposition n'est pas d'assurer une disponibilité semblable à la protection, mais d'exploiter au mieux les informations sur les pannes possibles dans le réseau afin d'améliorer la disponibilité fournit par la restauration. Ainsi, lors d'une prédiction de panne, la rapidité d'exécution des machines permet au couple RAM et au module FMF du R&S_DE de déclencher la modification du mécanisme de résilience des LSP concernés par la panne, afin, le cas échéant, d'être capable de rétablir le LSP dans un délai minimum. De plus, cette méthode permet les performances de disponibilité de la protection avec l'avantage de la restauration puisque l'on connaît à l'avance les éléments concernés par la future panne. Ainsi il est possible d'adapter le chemin du LSP de protection de façon optimale pour n'exclure que les éléments risqués.

Si par contre la prédiction devait s'avérer être fausse, le trafic ne serait pas impacté, mais les ressources réservées pendant le temps de la prédiction Δt_p auraient été inutilement consommées, créant un coût supplémentaire. Néanmoins, cette surconsommation limitée au temps Δt_p reste négligeable comme en atteste les résultats obtenus. (Cf les sections suivantes). Enfin, si une panne n'est pas précédée par une prédiction, c'est le dispositif de restauration qui est utilisé pour rétablir les services perturbés. Cela permet d'obtenir un meilleur *ratio* coût/disponibilité que les mécanismes de protection et de restauration du protocole GMPLS en exploitant les indicateurs de l'état de santé du réseau, ainsi que la rapidité d'exécution (supérieure à celle d'un être humain) que permet l'autoréparation.

Afin de réaliser ce changement dynamique, le module FMF du R&S_DE (voir Fig. 4.2), suite à la réception d'une information de risques de panne en provenance du RAM, doit prendre le contrôle des éléments en charge du protocole GMPLS (*i.e.* OSPT-TE et RSVP-TE) sur le *Ingress router* de chaque LSP concerné par la panne afin de modifier le mécanisme de résilience associé à chaque LSP. Pour ce faire, il est nécessaire de prendre en compte les spécificités et outils du protocole GMPLS tel que le MBB et le XRO [1].

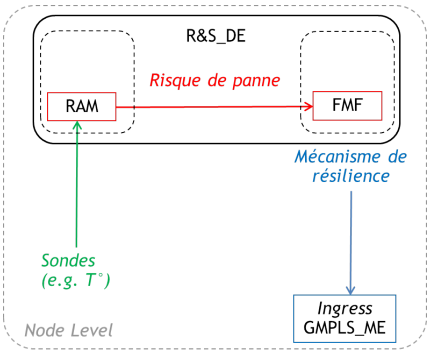


FIGURE 4.2: Architecture fonctionnelle du mécanisme ALR.

4.5.2 Considérations protocolaires

Le dispositif ALR doit composer avec les spécificités du protocole GMPLS, pour en décrire le fonctionnement. Ceci est illustré au travers de l'exemple des Fig. 4.4, 4.5 et 4.6 de la section suivante.

Afin de changer le niveau de protection, deux procédures distinctes sont nécessaires. La première a pour rôle de mettre à jour le chemin principal (*working*) avec les nouvelles caractéristiques du mécanisme de résilience. La seconde a pour rôle la configuration et la mise en place éventuelle des ressources dédiées au chemin de sauvegarde.

En effet, la protection et la restauration GMPLS concernent trois objets décrits dans la RFC4872 [LRP07] : les objets *Session*, *Association* et *Protection*. L'objet « session »contient des paramètres pour identifier un LSP tel qu'un *Tunnel ID*, un *LSP ID* ainsi que la source

1. *Exclude Route Object*

et la destination du tunnel. Un tunnel protégé est composé de deux LSP possédant le même *Tunnel ID*. Le *LSP ID* permet de différencier les deux LSP du tunnel que sont le LSP *working* et le LSP protection. L'objet « association »contient un paramètre important, l'*Association ID* qui référence le LSP associé, c'est-à-dire le LSP *working* lors du basculement sur le LSP de protection et *vice-versa*. Enfin, l'objet « protection »représenté à la Fig. 4.3 contient les détails du schéma de résilience :

- 0x00 pour non protégé ;
- 0x01 pour reroutage complet ;
- 0x02 pour reroutage avec trafic externe ;
- 0x04 pour la protection 1 :N (N>=1) ;
- 0x08 pour la protection 1+1 unidirectionnelle ;
- 0x10 pour la protection 1+1 bidirectionnelle.

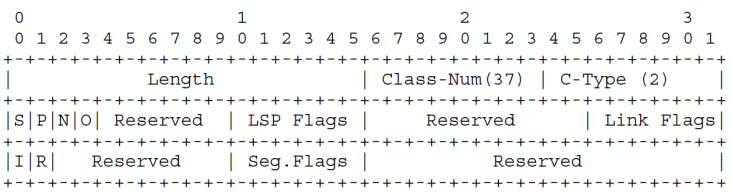


Figure 4.3: Objet protection de RSVP.

Les autres paramètres importants comprennent un bit (P) qui précise si le LSP est un LSP de protection, un bit (O) qui renseigne si le LSP est actuellement en train de transporter du trafic et un bit (S) qui indique si la protection est cross-connectée.

La mise à jour dynamique du mécanisme de résilience utilisé nécessite de modifier ces trois objets sur le LSP principal et dans le même temps de créer le nouveau LSP de sauvegarde et les ressources qui lui sont associées.

Mais, la mise à jour dynamique de la protection d'un LSP n'est pas nativement supportée par le protocole GMPLS. Il est donc nécessaire de combiner cela avec un second mécanisme nommé *Make-Before-Break* (MBB). Le MBB défini dans la RFC3209 [ABG+01] permet de cloner un LSP avant de supprimer les ressources de celui-ci. Ce dispositif utilise l'option *Shared Explicit* qui rend possible l'utilisation par le nouveau LSP des mêmes ressources que le précédant LSP sur les segments qui sont partagés. En effet, le nouveau LSP principal doit avoir un chemin identique à l'ancien LSP. Sans l'option *Shared Explicit*, il est fort possible que l'établissement du nouveau LSP principal ne soit pas possible par manque de bande passante. Avec cette option, le nouveau LSP peut utiliser les ressources de l'ancien LSP. Une fois le nouveau LSP établi avec les bons paramètres de gestion des pannes, le trafic peut être basculé d'un LSP à l'autre, sans perturbation du service.

De plus, un objet particulier appelé *Exclude Route Object* (XRO), définit dans la RFC4874 [LFC07], permet de spécifier des routeurs à exclure de la route d'un LSP. Cet objet est utilisé pour la création du LSP de protection afin de ne pas utiliser les routeurs avec un risque de panne. En combinant les trois composants que sont les objets du mécanisme de résilience, le MBB avec l'option *Shared Explicit* et le XRO, il est possible d'implémenter un mécanisme de résilience adaptatif entièrement compatible avec les spécifications de GMPLS.

4.5.2.1 Extension protocolaire pour une meilleure intégration au protocole de gestion des pannes de GMPLS

Même si le changement du dispositif de résilience est possible par l'intermédiaire du MBB, cela pourrait être fait de manière beaucoup plus efficace. En effet, cette modification ne change pas fondamentalement les caractéristiques du LSP principale. Une simple mise à jour de ce LSP avec les nouvelles caractéristiques de la résilience serait préférable à l'établissement complet d'un nouveau LSP principal comme c'est le cas avec le MBB. Les étapes de signalisation du nouveau LSP de protection restent néanmoins obligatoires.

Une autre amélioration possible est l'intégration du mécanisme de changement de niveau de protection au sein du contrôleur GMPLS. En effet, il serait possible d'utiliser le mécanisme

de notification (message *notify*) pour transporter l'information de risque de panne jusqu'au routeur d'entrée du LSP. Pour cela, il serait nécessaire d'ajouter des nouveaux *Error_Code* dans l'objet *ERROR_SPEC* du message *notify* afin de signifier lorsqu'un risque de panne est détecté, ou lorsqu'un élément redevient non risqué. Suite à la réception du message *notify*, le routeur d'entrée du LSP pourrait alors automatiquement déclencher le changement du dispositif de résilience. Mais ce type de fonctionnement peut ne pas être désiré par tous les opérateurs et ce sur tous les LSP, il est donc nécessaire de rajouter une option pour activer ou non ce type de fonctionnement sur un LSP. Une option possible serait l'utilisation des bits de poids fort du champ *LSP Protection Type* de l'objet protection [LRP07] qui sont actuellement non utilisés et qui permettrait de spécifier si un changement de mécanisme de résilience doit être initié lorsqu'une information de risque de panne est reçu par l'intermédiaire d'un message *notify*.

Néanmoins, les mécanismes de MBB permettent une première implémentation du dispositif ALR sans extension protocolaire, ce qui est un avantage pour la promotion de ce nouveau mécanisme d'autoréparation proactif.

4.5.3 Illustration par l'exemple

L'exemple décrit ci-dessous permet de visualiser le comportement du mécanisme de manière pratique. La Fig. 4.4 illustre la situation normale où un LSP est établi pour transporter du trafic entre deux routeurs. Aucun des routeurs composant le chemin du LSP principale étant risqué, aucun LSP de protection n'est configuré ; c'est donc la restauration qui fait office de mécanisme de résilience associé.

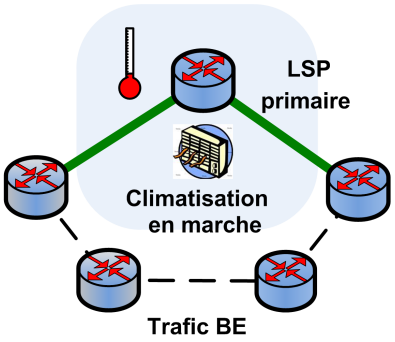


FIGURE 4.4: Configuration initiale du mécanisme ALR.

Dans un deuxième temps (voir Fig. 4.5), le système de refroidissement du routeur situé au milieu du LSP tombe en panne. Cela se traduit rapidement par une augmentation de la température qui est observée par le module de prédiction de pannes du RAM. Lorsque la température devient trop importante, le RAM de l'équipement concerné diffuse une information de risque de panne aux éléments concernés et notamment au module FMF du R&S_DE en charge du contrôle de GMPLS. Une fois l'information de risque de panne reçue, le nœud source du LSP est piloté afin de déclencher le changement du dispositif de résilience pour utiliser la protection. Afin d'établir un LSP de protection, le routeur de bordure calcule le nouveau chemin pour la protection avec un XRO contenant le routeur risqué. Il doit aussi re-signaler un nouveau LSP principal qui utilise les même ressources que le précédant LSP en faisant usage de l'option *Shared Explicit* du MBB. Enfin, une fois le nouveau LSP principal associé au LSP de protection, le trafic peut alors utiliser le nouveau tunnel de manière transparente pour l'utilisateur. L'ancien LSP principal peut alors être supprimé pour aboutir à la configuration de la Fig. 4.5 où le trafic est maintenant protégé par un LSP dédié.

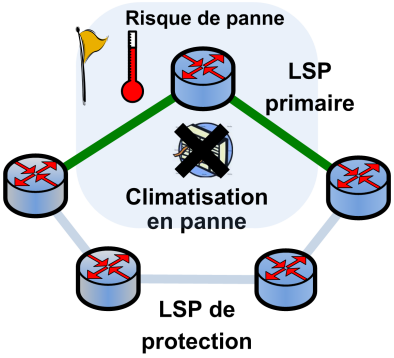


Figure 4.5: Modification du mécanisme de résilience de la restauration vers la protection.

Ainsi, lorsque la panne apparaît, comme l'illustre la Fig. 4.6, le trafic est orienté de manière extrêmement rapide sur le LSP de protection afin de minimiser l'interruption du service. L'utilisation de la protection aura permis une interruption de service beaucoup plus faible que

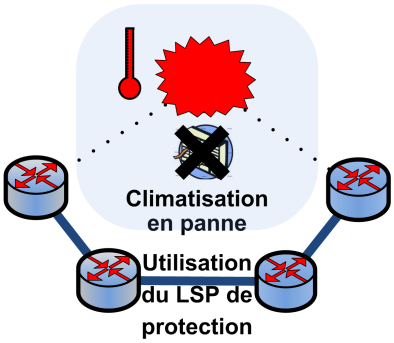


Figure 4.6: Incidence d'une panne sur le mécanisme ALR.

si la restauration avait dû s'appliquer. Mais la majorité du temps, lorsqu'aucun risque de panne n'est détecté, les LSP sont configurés pour utiliser la restauration comme l'illustre la Fig. 4.4, ce qui permet une consommation des ressources beaucoup moins importante qu'avec l'utilisation de la protection de manière systématique.

4.5.4 Algorithme

L'Algorithme 3 résume le déroulement du dispositif ALR suite à l'envoi d'une information de risque de panne par le RAM au module FMF. Cette information de risque de panne peut concerner trois types d'équipements, le routeur (noté $n \in N$ où N est l'ensemble des routeurs du réseau), un lien du routeur (noté *link*) ou l'interface à laquelle est rattachée ce lien sur le routeur (noté IF).

algorithme 3 Adaptation du mécanisme de résilience de GMPLS.

```
1: for each router n_i in N do
2:     if n_i.new_risk > n_i.old_risk then
3:         for All lsp_j in n_i.lsp do
4:             if not lsp_j.isprotected() OR ( lsp_j.isprotected() AND n_i in lsp_j.Protection_Path ) then
5:                 create new Working LSP with SharedExplicit option (Make-BeFore-Break)
6:                 create new Protection Path by adding n_i to XRO
7:                 execute Associate (new Working LSP, new Protection Path)
8:                 execute Switch traffic From lsp_j.working to new Working LSP
9:                 delete lsp_j.old_working_LSP
10:                if lsp_j.old_Protection_Path.exist() then
11:                    delete lsp_j.old_Protection_Path
12:                end if
13:            end if
14:        end for
15:    else if n_i.new_risk <n_i.old_risk then
16:        for All lsp_j in n_i.lsp do
17:            if not lsp_j.working_LSP.contain_risky_element() then
18:                create new Working LSP with SharedExplicit option (Make-BeFore-Break)
19:                execute ConfigureRecoveryLevel(new Working LSP, Restoration)
20:                execute Switch traffic From lsp_j.working to new Working LSP
21:                delete lsp_j.Protection_Path
22:                delete lsp_j.old_working_LSP
23:            else
24:                create new Working LSP with SharedExplicit option (Make-BeFore-Break)
25:                create new Protection Path by removing n_i to XRO
26:                execute Associate (new Working LSP, new Protection Path)
27:                execute Switch traffic From lsp_j.working to new Working LSP
28:                delete lsp_j.old_working_LSP
29:                delete lsp_j.old_Protection_Path
30:            end if
31:        end for
32:    else if n_i.new_risk = normal then
33:        for each link_k in n_i.links do
34:            if (link_k.new_risk > link_k.old_risk) OR (link_k.If.new_risk > link_k.If.old_risk) then
35:                for All lsp_j in link_k.lsp do
36:                    if not lsp_j.isprotected() OR ( lsp_j.isprotected() AND link_k in lsp_j.Protection_Path ) then
37:                        create new Working LSP with SharedExplicit option (Make-BeFore-Break)
38:                        create new Protection Path by adding link_k to XRO
39:                        execute Associate (new Working LSP, new Protection Path)
40:                        execute Switch traffic From lsp_j.working to new Working LSP
41:                        delete lsp_j.old_working_LSP
42:                        if lsp_j.old_Protection_Path.exist() then
43:                            delete lsp_j.old_Protection_Path
44:                        end if
45:                    end if
46:                end for
47:            else if (link_k.new_risk < link_k.old_risk) OR (link_k.If.new_risk < link_k.If.old_risk) then
48:                for All lsp_j in link_k.lsp do
49:                    if not lsp_j.working_LSP.contain_risky_element() then
50:                        create new Working LSP with SharedExplicit option (Make-BeFore-Break)
51:                        execute ConfigureRecoveryLevel(new Working LSP, Restoration)
52:                        execute Switch traffic From lsp_j.working to new Working LSP
53:                        delete lsp_j.Protection_Path
54:                        delete lsp_j.old_working_LSP
55:                    else
56:                        create new Working LSP with SharedExplicit option (Make-BeFore-Break)
57:                        create new Protection Path by removing link_k from XRO
58:                        execute Associate (new Working LSP, new Protection Path)
59:                        execute Switch traffic From lsp_j.working to new Working LSP
60:                        delete lsp_j.old_working_LSP
61:                        delete lsp_j.old_Protection_Path
62:                    end if
63:                end for
64:            end if
65:        end for
66:    end if
67: end for
```

4.6 Modélisation analytique

Cette section définit le modèle analytique permettant de comparer les performances de notre mécanisme adaptatif de résilience avec les mécanismes classiques de restauration (R) et de protection 1 :1 (P). Les deux critères principaux de notre comparaison sont l'indisponibilité de service et la consommation de ressources réseaux. Pour cela, notre modèle analytique prend en compte la probabilité de panne d'un élément de réseau (MTBF, MTTR), mais pour des raisons de simplicité, nous ne considérons dans cette thèse, que les pannes de routeurs. Le modèle reste néanmoins facilement transposable à l'ajout des pannes de lien ou de carte de ligne. De plus, les performances de la prédiction de pannes sont aussi pris en compte *via* les paramètres *Recall*, *Precision* et Δt_p.

4.6.1 Définitions et notations

Le réseau est représenté par un graphe orienté $G=(N,E)$ où N est l'ensemble des nœuds (routeurs) du réseau et E l'ensemble des liens orientés. Le trafic est représenté par l'ensemble F des flux de trafic transportés par le réseau G où chaque flux de trafic $f \in F$ est défini par son routeur source $In(f) \in N$, son routeur destination $Out(f) \in N$ et son débit $\mu(f)$ en bits/s. Pour chaque flux de trafic $f \in F$, le protocole de routage (OSPT-TE) définit le chemin principal (*working path*) composé des routeurs de transit par le sous-ensemble $w(f) \subseteq N$. Pour le dispositif de protection, il définit deux chemins disjoints que sont le chemin principal $w(f) \subseteq N$ et le chemin de protection $p(f) \subseteq N$ tel que $w(f) \cap p(f) = \emptyset$. Pour la restauration, lorsque un nœud $n \in N$ tombe en panne, un chemin de restauration $r(f,n)$ est défini en supprimant le nœud n du graphe G tel que $r(f,n) \subseteq N \backslash \{n\}$.

Dans ce modèle, les conséquences d'une panne sur les nœuds source et destination ne sont pas prises en compte puisque les mécanismes de résilience intra-domaine n'ont aucun moyen de rétablir une telle situation. Néanmoins, dans les réseaux d'opérateurs, de telles circonstances sont gérées par des techniques de *multi-homing* impliquant une redondance des nœuds d'extrémité et l'intervention de plusieurs réseaux.

La comparaison entre le comportement des trois dispositifs de résilience (P,R et ALR) varie suivant deux critères, la probabilité de panne d'un nœud et la prédiction de pannes.

Pour commencer, la probabilité de panne est caractérisée par $MTBF(n)$, le temps moyen entre deux pannes du nœud $n \in N$, et $MTTR(n)$ le temps moyen de réparation du nœud $n \in N$. Le temps de réparation ést très inférieur au temps entre deux pannes $MTTR(n) << MTBF(n)$. Pour un processus ergodique stationnaire, la probabilité qu'un nœud $n \in N$ soit en panne $P_{node}(n)$ est :

$$P_{node}(n) = \frac{MTTR(n)}{MTBF(n) + MTTR(n)} << 1 \tag{4.1}$$

Lorsque les routeurs sont considérés comme identiques, le paramètre n est omis dans les notations de tous les paramètres.

Lors d'une panne, la mise en œuvre des mécanismes de protection (P) et de restauration (R) entraîne une indisponibilité du service. Pour chaque flux de trafic $f \in F$, lorsqu'un routeur de transit $n \in w(f)$ tombe en panne, le mécanisme de protection nécessite un délai $T_P(f,n)$ pour rediriger le trafic sur le chemin de protection. Ce cas est identique au cas d'une prédiction de panne réussi avec le dispositif ALR. Dans le cas d'une fausse prédiction, la mise en œuvre d'un chemin alternatif est similaire au mécanisme de restauration et occasionne un délai $T_R(f,n)$. Enfin les valeurs de ces délais sont telles que $0 \leq T_P(f,n) << T_R(f,n) << MTTR(n)$. Dans cette modélisation, T_P et T_R sont identiques pour tout le réseau et définis par les moyennes pondérées suivantes :

$$T_P = \frac{\sum_{f \in F} \sum_{n \in w(f)} \mu(f).P_{node}(n).T_P(f,n)}{\sum_{f \in F} \sum_{n \in w(f)} \mu(f).P_{node}(n)} \tag{4.2}$$

$$T_R = \frac{\sum_{f \in F} \sum_{n \in w(f)} \mu(f).P_{node}(n).T_R(f,n)}{\sum_{f \in F} \sum_{n \in w(f)} \mu(f).P_{node}(n)} \tag{4.3}$$

Concernant le mécanisme ALR, celui-ci est paramétré par les trois variables de la prédiction de pannes (Voir Sec. 1.6.3) que sont le *Recall*, la *Precision* et Δt_p. Alors que Δt_p est une constante du dispositif de prédiction, $Recall(n)$et $Precision(n)$ sont les variables associées à chaque routeur $n \in N$.

Enfin tous ces paramètres sont considérés comme non modifiables dans le temps.

4.6.2 Données de la comparaison

4.6.2.1 Estimation de l'indisponibilité du réseau

Pour le mécanisme ALR, la probabilité conditionnelle peut être divisée en une somme de deux probabilités concernant une panne prédite ($Recall$) et une panne non prédite ($1 - Recall$) ce qui donne :

$$\begin{aligned} U_{ALR}(f,n) = P_{node}.Recall(n).T_P(f,n)/MTTR(n) \\ +P_{node}(n).(1 - Recall(n)).T_R(f,n)/MTTR(n) \end{aligned} \tag{4.4}$$

Il est utile de noter que la formule pour le cas de la protection (respectivement le cas de la restauration) est donnée par l'Eq. (4.4) avec $Recall(n) = 1$ (respectivement $Recall(n) = 0$). Il en résulte donc la relation suivante :

$$U_{ALR}(f,n) = Recall(n).U_P(f,n) + (1 - Recall(n)).U_R(f,n) \tag{4.5}$$

Ce qui donne $U_P(f,n) \leq U_{ALR}(f,n) \leq U_R(f,n)$. En considérant que les pannes des routeurs sont des événements indépendants, pour $X = ALR$, P et R, l'indisponibilité du flux $f \in F$ est :

$$U_X(f) = 1 - \left(\prod_{n \in w(f)} (1 - U_X(f,n)) \right) \approx \sum_{n \in w(f)} U_X(f,n) \tag{4.6}$$

Cette approximation est valide puisque suivant l'Eq. (4.1), le cas de pannes simultanées d'au moins deux routeurs dans un même réseau est suffisamment petit pour être négligé.

Afin de calculer l'indisponibilité du réseau, il est nécessaire de définir une moyenne de l'indisponibilité de chaque flux f pondérée par le débit de chaque flux $\mu(f)$:

$$U_X(G,F) = \frac{\sum_{f \in F} \mu(f).U_X(f)}{\sum_{f \in F} \mu(f)} \tag{4.7}$$

À partir de l'Eq. (4.5), $U_P(G,F) \leq U_{ALR}(G,F) \leq U_R(G,F)$. Le facteur de gain du mécanisme ALR par rapport à la restauration classique peut être défini par le *ratio* suivant :

$$\gamma_{ALR/R} = U_R(G,F)/U_{ALR}(G,F) \geq 1 \tag{4.8}$$

Dans le cas de routeurs identiques, les Eq. (4.5), (4.6) et(4.7) donnent :

$$U_{ALR}(G,F) \approx Recall.U_P(G,F) + (1 - Recall).U_R(G,F) \tag{4.9}$$

Enfin, suivant les Eq. (4.8), (4.2) et (4.3), le gain du mécanisme ALR par rapport à la restauration donne finalement :

$$\gamma_{ALR/R} = \frac{1}{1 - Recall.(1 - T_P/T_R)} \tag{4.10}$$

On peut notamment noter que le gain de la protection par rapport à la restauration est donné par l'Eq. (4.10) avec $Recall = 1$, i.e., $\gamma_{P/R} \approx T_R/T_P$.

4.6.2.2 Estimation de la consommation de ressource

Bien que l'avantage du dispositif ALR sur la protection soit déjà tout d'abord un dimensionnement plus faible (CAPEX), il est préférable de comparer les trois mécanismes sur la consommation réelle des ressources du réseau. Il s'agit de la bande passante réservée sur chaque routeur par chacun des mécanismes et qui ne peut être utilisée pour d'autres besoins. Cette utilisation des ressources dépend tout d'abord de la topologie du réseau (G), mais aussi du trafic que transporte ce réseau, exprimé sous forme de matrices de trafic (F). Afin de calculer les ressources utilisées par chacun des mécanismes, nous avons utilisé le logiciel de dimensionnement FAB [1] développé aux Bell Labs [PHNP08].

En ce qui concerne la protection 1 :1, la consommation de ressources $R_P(G,F)$ est obtenue en sommant la bande passante réservée sur tous les routeurs des chemins principaux et des chemins de protection. Les ressources de protection sont incluses dans les ressources consommées puisque celles-ci sont réservées et donc ne sont plus disponibles pour d'autres services. Pour la restauration, il est nécessaire de différencier deux cas de figures :

1. lorsqu'aucune panne ne survient où seules les ressources utilisées par les plus courts chemins sont intégrées dans R_0.
2. lorsque un nœud $n \in N$ tombe en panne où R_n additionne à R_0, les ressources empruntées par les chemins de restauration re-routant le trafic qui passait par n. En effet, Il est nécessaire de ne pas libérer les ressources des chemins initiaux passant par n afin de préparer au mieux l'étape de retour à la normale.

En négligeant les cas très peu probables des pannes simultanées, les ressources consommées en moyenne par le dispositif de restauration intègrent les périodes sans panne et les périodes de pannes de chaque routeur n en fonction de leur probabilité $P_{node}(n)$ pour donner l'Eq. (4.11) :

$$R_R(G,F) = \left(1 - \sum_{n \in N} P_{node}(n)\right).R_0 + \sum_{n \in N} P_{node}(n).R_n \tag{4.11}$$

Le calcul de la consommation de ressources du mécanisme ALR est plus complexe puisqu'il doit considérer le cas des TP, des FP et des FN en faisant intervenir le $Recall$, la $Precision$ et Δt_p. Avec le $Recall$ et la $Precision$ définis par les Eq. (1.1) et (1.2), les probabilités de TP, de FP et de FN pour un nœud n sont :

$$\begin{aligned} P_{TP}(n) &= P_{node}(n).Recall(n).\left(1 + \frac{\Delta t_p/2}{MTTR(n)}\right) \\ P_{FP}(n) &= P_{node}(n).Recall(n).\left(\frac{1}{Precision} - 1\right).\left(\frac{\Delta t_p}{MTTR(n)}\right) \\ P_{FN}(n) &= P_{node}(n).(1 - Recall(n)) \end{aligned} \tag{4.12}$$

Les formules de l'Eq. (4.12) permettent de prendre en compte que :
- lors d'une mauvaise prédiction (FP), le réseau utilise des ressources supplémentaires pour protéger une futur panne inexistante pendant toute la durée de la prédiction définie par Δt_p ;
- lorsqu'une panne est prédite à l'avance (TP), le réseau consomme des ressources supplémentaires uniquement avant l'apparition de la panne. Cette période commence à la date du début de la prédiction (début de Δt_p) et se termine à l'apparition de la panne. En considérant que l'occurrence des pannes est uniformément distribuée pendant Δt_p, *i.e.* entre 0 et Δt_p, la durée moyenne d'utilisation inutile des ressources de protection est $\Delta t_p/2$.

En remplaçant $P_{node}(n)$ by $P_{ALR}(n) = P_{TP}(n) + P_{FP}(n) + P_{FN}(n)$ dans l'Eq. (4.11), on obtient la consommation moyenne de ressources dans tout le réseau pour le mécanisme ALR :

$$R_{ALR}(G,F) = \left(1 - \sum_{n \in N} P_{ALR}(n)\right).R_0 + \sum_{n \in N} P_{ALR}(n).R_n \tag{4.13}$$

1. *Forwarding Adjacency Builder*

Fort des formules permettant d'estimer à la fois l'indisponibilité mais aussi la consommation des ressources, il est maintenant possible de comparer les trois mécanismes sur des configurations réseaux représentatives de la réalité des opérateurs.

4.7 Étude de cas : trois réseaux de classe opérateur

Afin d'analyser le comportement du dispositif ALR proposé vis-à-vis des mécanismes standards de résilience GMPLS grâce au modèle analytique, les trois configurations de réseaux cœur (voir Fig. 2.7) précédemment utilisées permettent de mesurer l'impact des variables du modèle. Bien que ces topologies aient été présentées à la Sec. 2.7, il peut être utile de rappeler que la première topologie est un réseau national allemand (Fig. 2.7a) de 17 nœuds similaire au réseau utilisé dans [MMJ08], que le deuxième réseau est la topologie NSF-Net à 29 nœuds (Fig. 2.7b) et enfin que la dernière topologie est un réseau européen de 34 nœuds (Fig. 2.7c).

Mais contrairement aux deux chapitres précédents, deux dimensionnements différents sont utilisés pour la restauration et la protection. Le Tab. 4.1 permet de synthétiser les caractéristiques de chaque topologie et notamment le nombre d'interfaces nécessaires pour chaque dimensionnement. Basé sur la topologie, les métriques et la matrice de trafic, le logiciel de dimensionnement FAB [PHNP08] permet d'obtenir le dimensionnent nécessaire au mécanisme de protection d'une part, et aux dispositifs de restauration et d'ALR d'une autre part. Ce dimensionnement réalisé à partir d'interfaces de 10 Gbits/s est exprimé en nombre d'interface totale dans le Tab. 4.1.

Réseau	$\|N\|$	$\|E\|$	$\bar{d}$	Densité	$d(G)$	$\|F\|$	Trafic (Gbit/s)	P # IF	ALR et R # IF
A	17	26	3,06	0,19	8	242	1363	816	626
US	29	44	3,03	0,11	9	812	485	492	440
EU	34	49	2,88	0,09	14	1122	1554	1510	1244

TABLE 4.1: Caractéristiques des réseaux étudiés.

Au travers de ce dimensionnement, le premier avantage du mécanisme ALR sur le mécanisme de protection 1 :1 est identifié, puisqu'on peut voir sur la Fig. 4.7 que le mécanisme de protection entraîne un coût en CAPEX de 12 à 30% supérieur au coût du dimensionnement nécessaire à l'exécution des dispositifs de restauration et ALR. Pour plus de détails, l'annexe A contient les dimensionnements nécessaires au mécanisme de protection (Cf. les Tab. A.4, A.9 et A.14) ainsi que les dimensionnements utilisés pour les mécanismes ALR et restauration (Cf les Tab. A.3, A.8 et A.13).

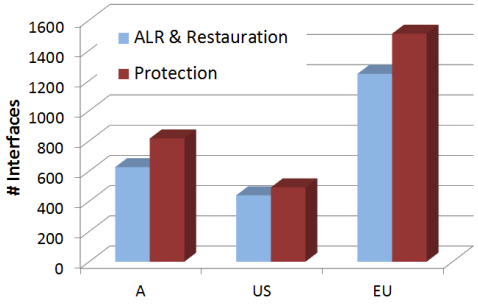


FIGURE 4.7: Comparaison des dimensionnements des réseaux.

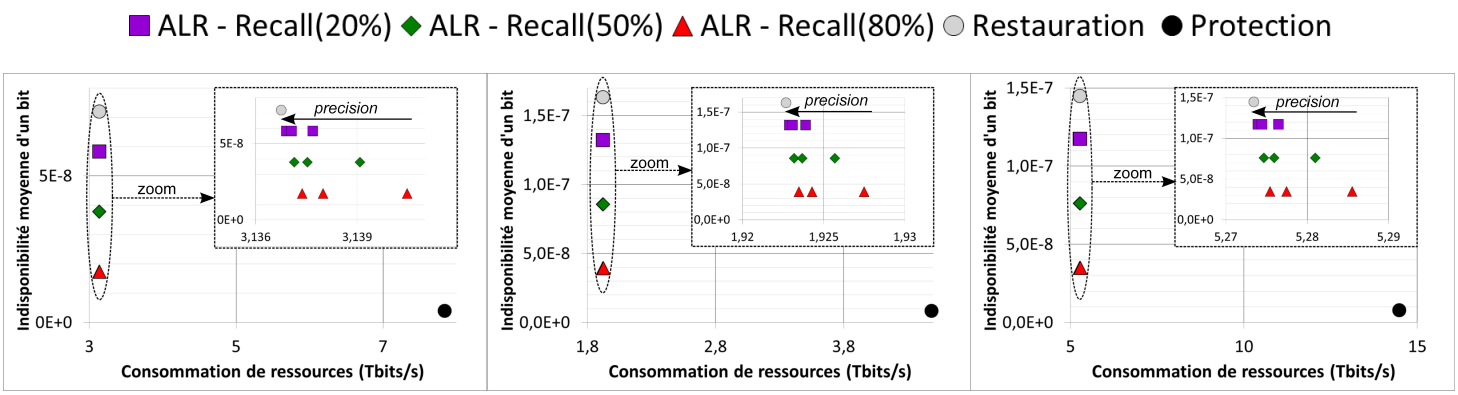
Les autres paramètres de ces trois topologies (Fig. A.1, A.2 et A.3) tels que les métriques de routage (Tab. A.2, A.7 et A.12) et la matrice de trafic (Tab. A.5, A.10 et A.15) sont aussi disponible à l'annexe A.

4.8 Application numérique du modèle analytique

Le modèle de calcul de l'indisponibilité et de la consommation de ressources a été appliqué aux trois topologies susmentionnées, afin d'étudier l'impact des deux paramètres caractérisant les pannes que sont le MTBF et le MTTR sur le comportement du dispositif ALR. De même, les répercussions du *Recall*, de la *Precision* et de Δt_p ont été analysées.

Pour cela, des conditions de références ont été utilisées avec un MTBF de 5000 heures ainsi qu'un MTTR de 5 heures identique pour chaque routeur. De même, la prédiction de pannes est considérée comme identique sur chaque routeur et avec une durée de validité de référence d'une prédiction Δt_p d'une heure. Bien que l'état de l'art propose des valeurs de Δt_p de quelques minutes [ST08], nous avons constaté qu'une période d'une heure n'engendrait pas de différence significative dans le comportement de notre mécanisme. Étant donné que plus Δt_p est grand, plus la prédiction de pannes est performante, cette valeur d'une heure permet une compatibilité avec la quasi-totalité des mécanismes de prédiction de pannes, et laisse une marge suffisamment importante pour espérer rester compatible avec les futures fonctionnalités de prédiction que proposeront les équipements de réseau. Ce Δt_p d'une heure est donc utilisé pour les neuf configurations de prédiction de pannes de références formées par la combinaison d'un *Recall* et d'une *Precision* de 20%, 50% et 80%. Enfin, cette analyse numérique considère une interruption de service de 50 millisecondes dans le cas de la protection (T_P) et de 1 seconde pour la restauration (T_R).

4.8.1 Analyse conjointe de la disponibilité et des ressources utilisées



(a) A (b) US (c) EU

Figure 4.8: *Ratio* indisponibilité / utilisation des ressources avec les conditions de référence pour les trois topologies.

La Fig. 4.8 représente l'indisponibilité et la consommation de ressources théoriques dans les conditions de référence décrites à la Sec. 4.8 pour les trois topologies. Les combinaisons des trois valeurs de *Recall* et de *Precision* de 20%, 50% et 80% pour le dispositif ALR sont comparées à l'utilisation de la protection et de la restauration. Cette figure illustre l'intérêt du mécanisme ALR par rapport au mécanisme de protection et de restauration en termes de *ratio* ressources utilisées et indisponibilité. Le *Recall* a un impact significatif sur la disponibilité, au contraire de la *Precision*. Concernant la consommation de ressources, celle-ci augmente lorsque la *Precision* diminue (voir la Fig. 4.8), mais cette augmentation reste négligeable au regard de la consommation de ressources du mécanisme de prédiction. Cette illustration des conditions de référence pose les bases de l'avantage du dispositif ALR sur le mécanisme de restauration, avec un gain d'indisponibilité $\gamma_{ALR/R}$ supérieur à 4 pour le mécanisme ALR avec un *Recall* de 80%, de quasiment 2 pour un *Recall* de 50% et de 1,4 pour un *Recall* de 20%. Le mécanisme ALR ne permet pas d'assurer une disponibilité égale à la protection mais permet des performances intermédiaires permettant de se rapprocher de la protection jusqu'à un facteur 4.

En contrepartie, l'observation de la consommation de ressources moyenne de chaque dispositif met en valeur l'excellente performance du mécanisme ALR qui reste très proche du mécanisme

de restauration, permettant un gain supérieur à 3 par rapport à la protection. Le faible impact sur la consommation de ressources de la *Precision* n'est visible que lorsque nous zoomons sur la Fig. 4.8 car l'effet de la consommation de ressources due aux faux positifs est largement contenue dans le temps et dans l'espace. À la vue de ces premiers résultats, l'utilisation d'un RAM avec un *Recall* le plus important possible au détriment de la *Precision* semble le plus adapté au dispositif ALR.

4.8.2 Influence de la probabilité de panne

Cette section est dédié à l'étude de l'impact de certains paramètres sur le comportement de notre mécanisme. Bien que les trois topologies aient été étudiées, les différents résultats sont similaires dans leur comportement. Dans un souci de clarté, nous illustrons notre analyse avec une seule topologie, à savoir la topologie européenne. Néanmoins, les résultats complets pour chaque topologie sont disponibles en annexe à la Sec. B.3. Il est intéressant d'évaluer l'impact théorique de tous les paramètres du modèle sur le comportement du mécanisme ALR, à commencer par les paramètres caractérisant les pannes, à savoir le MTBF et le MTTR. Pour cela, l'influence du MTBF dans une plage comprise entre 1000 et 10000 heures et celle du MTTR pour des valeurs allant de 1 à 10 heures sont analysées en termes de disponibilité et de consommation de ressource.

La Fig. 4.9 montre clairement l'impact de la fréquence des pannes sur la disponibilité. Une augmentation qui s'accélère en dessous d'un MTBF de 3000 heures est distinctement visible. Mais même si l'écart avec la protection s'agrandie, l'ordre et le *ratio* entre chaque mécanisme restent les mêmes que ceux constatés sur la configuration de référence (voir la Fig. 4.8c). Alors que le nombre de pannes a un impact clair, la durée de celles-ci ne change en rien l'indisponibilité lors d'une panne. La Fig. 4.10 confirme que le MTTR n'influence pas la disponibilité, car dans cette thèse, nous ne nous intéressons qu'à l'indisponibilité due aux délais d'intervention des mécanismes de résilience.

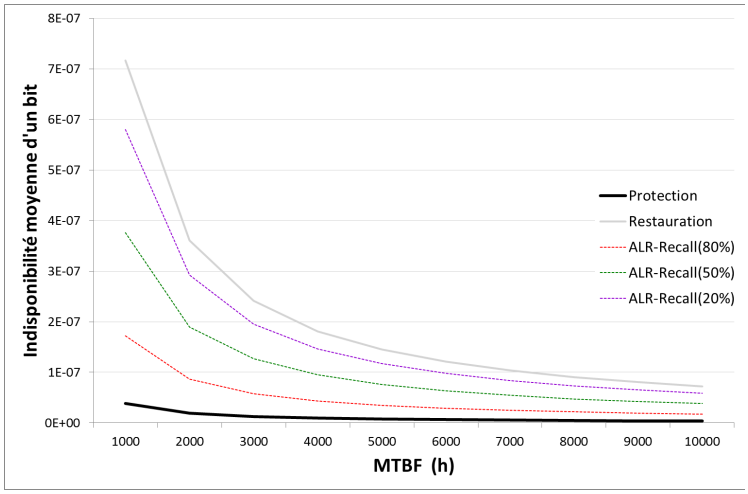


FIGURE 4.9: Impact du MTBF sur la disponibilité avec la topologie européenne.

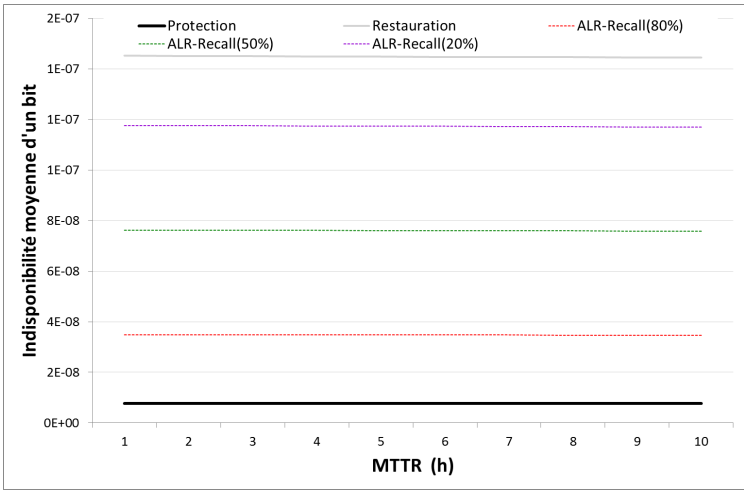


FIGURE 4.10: Impact du MTTR sur la disponibilité avec la topologie européenne.

Mais les choses sont différentes vis-à-vis de la consommation de ressources. Tout d'abord, la consommation de ressources de la protection étant stable quelle que soit la probabilité de panne et beaucoup plus importante que pour les autres mécanismes, elle ne figure pas dans les figures illustrant la consommation de ressources. Le lecteur peut se référer à la Fig. 4.8c où il peut noter une utilisation de 14,5 Tbits/s pour la topologie européenne. Le nombre de panne implique naturellement une consommation de ressources plus importante pour le mécanisme de restauration mais dans des proportions moins importantes que le dispositif ALR comme le montre la Fig. 4.11. Le surplus de consommation dû aux fausses prédictions est visible sur les faibles *Precision* (*i.e.* 20%) mais reste modéré en comparaison avec la consommation de la protection de 14,5 Tbits/s .

La durée des pannes a aussi une influence puisque, lors d'une panne, des ressources supplémentaires sont consommées. Néanmoins, cela engendre un comportement similaire quel que soit le *Recall* ou la *Precision*. Dans la plage de 1 à 10 heures, la Fig. 4.12 affiche une variation de seulement 0.5 % des ressources utilisées. On reste donc encore bien loin des valeurs du schéma de protection.

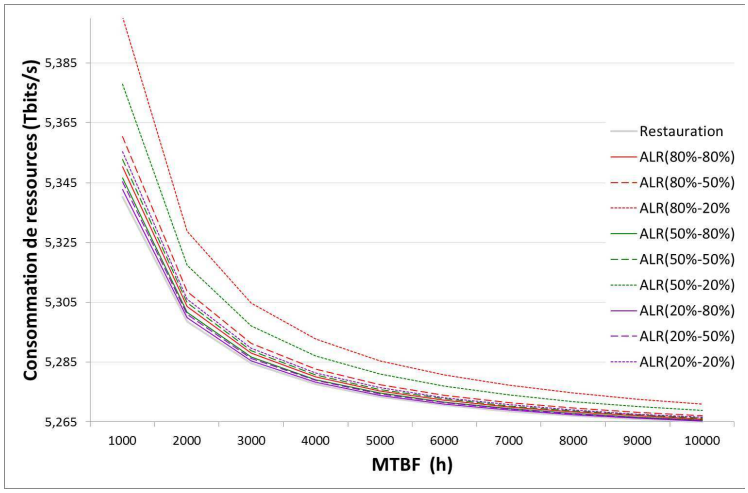


Figure 4.11: Impact du MTBF sur l'utilisation des ressources avec la topologie européenne.

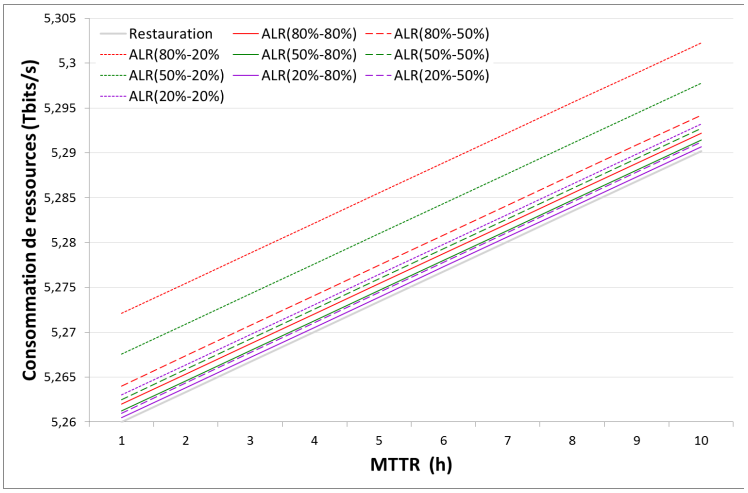


Figure 4.12: Impact du MTTR sur l'utilisation des ressources avec la topologie européenne.

4.8.3 Les conséquences de la prédiction de pannes

L'effet des paramètres de prédiction de pannes est l'une des questions importantes de cette thèse. Étant donné que nous ne proposons pas de méthode de prédiction de pannes associée à notre mécanisme, il est indispensable d'identifier les contraintes que doit posséder le module RAM, ainsi que le comportement qui en découle en fonction de ses performances. Pour cela, il est nécessaire d'analyser l'incidence de toutes les valeurs du *Recall* et de la *Precision* associées aux conditions de références, ainsi que le comportement du mécanisme ALR avec un Δt_p variant de 5 minutes à 10 heures.

Le paramètre qui, de prime abord, semble le plus important vis-à-vis du comportement du dispositif ALR en terme de disponibilité est le *Recall*. Celui-ci caractérise la proportion de pannes anticipées par le module RAM. La Fig. 4.13 montre fort logiquement une augmentation linéaire des performances en fonction du *Recall*. En commençant avec des performances similaires à la restauration pour le *Recall* le plus faible, le mécanisme ALR atteint les performances de la protection lorsque toutes les pannes sont anticipées (*i.e.* $Recall = 1$). Le modèle ne considérant aucune incidence de la *Precision* sur la disponibilité, la Fig. 4.13 est donc valable quelle que soit la *Precision* du mécanisme ALR.

En revanche, la situation est différente pour la consommation de ressources puisque les ressources de protection ne sont utilisées que lors d'une prédiction. La consommation additionnelle de ressources se fait en moyenne pendant une durée $\Delta t_p/2$ lors d'une prédiction correcte ce qui est négligeable car les ressources sont ensuite réservées pendant toute la durée de la panne (en moyenne égale au MTTR), aussi bien pour le dispositif ALR que pour la restauration. La différence la plus importante se situe au niveau des fausses prédictions. Le nombre de fausses prédictions est caractérisé par la *Precision*, elle-même dépendante du *Recall* (voir Eq. (1.2)). Cette relation est illustrée sur la Fig. 4.14 qui montre une augmentation de la consommation de ressources avec le *Recall*. Néanmoins, l'augmentation est plus forte pour les *Precision* faibles et notamment pour le mécanisme ALR avec une *Precision* de 20%. Mais cette augmentation ne représente qu'une augmentation de 0,3% par rapport à la restauration dans le pire cas, soit une valeur presque trois fois moins importante qu'avec le mécanisme de protection.

La Fig. 4.15 souligne l'incidence direct de la *Precision* sur la consommation de ressources avec une accélération en dessous de 30%. Le paramètre clé étant la quantité de fausses prédic-

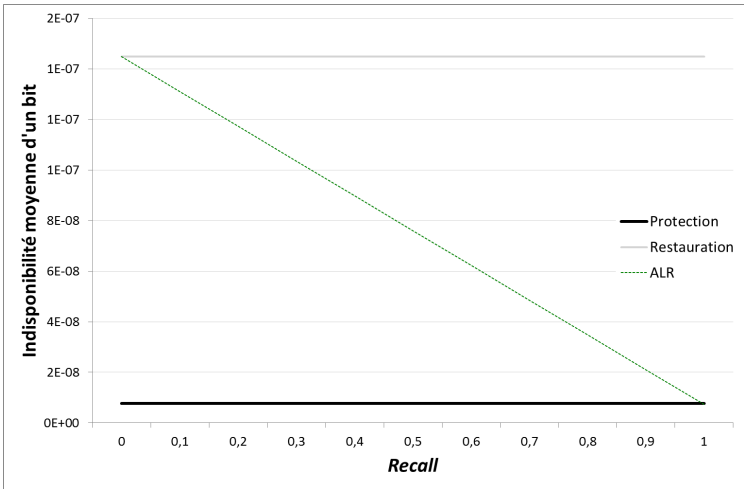


FIGURE 4.13: Impact du *Recall* sur la disponibilité avec la topologie européenne.

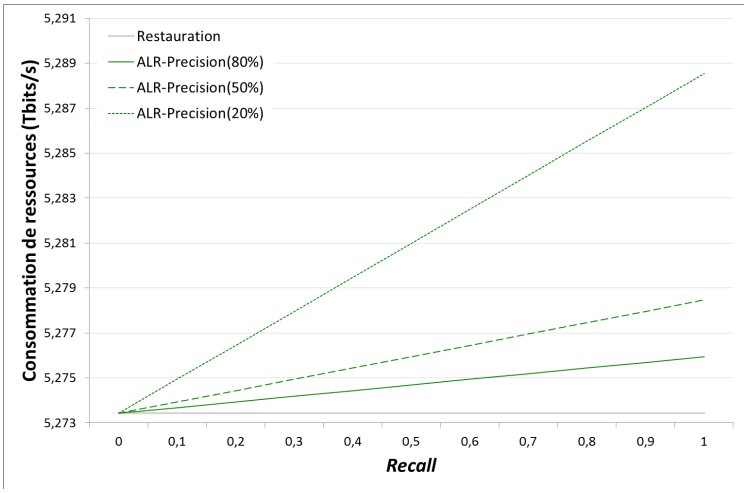


FIGURE 4.14: Impact du *Recall* sur l'utilisation des ressources avec la topologie européenne.

tions, le *Recall* influence indirectement les trois courbes. À *Precision* égale, le mécanisme avec le *Recall* le plus important provoque plus de fausses prédictions et donc une consommation plus importante. Mais conformément aux autres observations, la consommation reste de 5,3 Tbits/s dans le pire des cas, encore bien loin de la protection qui nécessite la réservation de 14,5 Tbits/s.

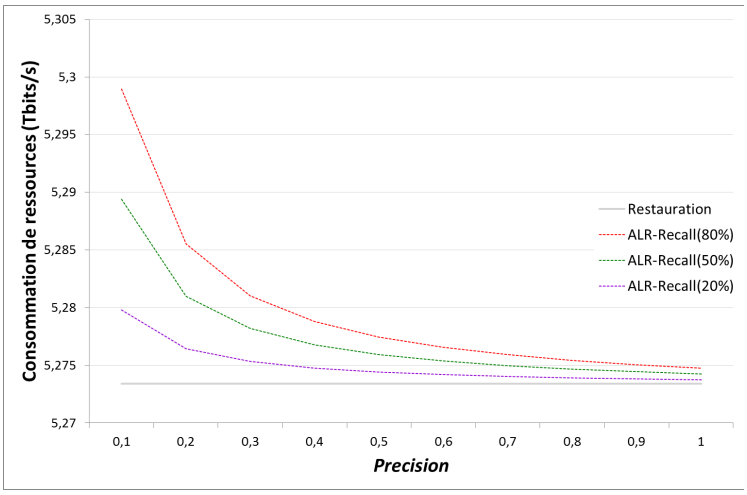


FIGURE 4.15: Impact de la *Precision* sur l'utilisation des ressources avec la topologie européenne.

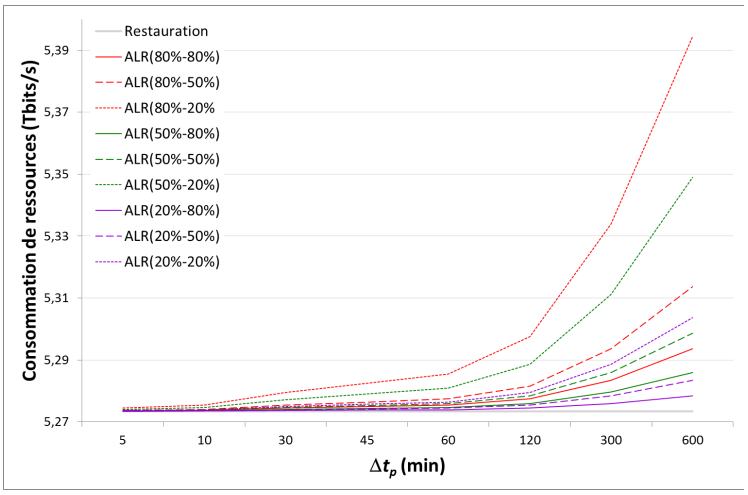


FIGURE 4.16: Impact de Δt_p sur l'utilisation des ressources avec la topologie européenne.

Dans notre configuration de référence, le Δt_p a délibérément été choisi avec une marge afin d'éviter de contraindre notre dispositif à un sous-ensemble trop restreint de mécanismes de prédiction. D'après notre modèle, le Δt_p ne modifie pas la disponibilité mais implique une surconsommation de ressources lors d'une prédiction, qu'elle soit bonne ou mauvaise. La Fig. 4.16 illustre ce comportement. Une variation de 2% est observée entre un Δt_p de 5 minutes et un autre de 10 heures pour le pire des cas. Cette surconsommation reste encore plus contenue jusqu'à deux heures, pour ensuite subir une augmentation plus rapide, notamment dans le cas avec de nombreuses fausses prédictions, *i.e.* avec une *Precision* faible associée à un *Recall* pas trop faible. Ce paramètre Δt_p peut aussi introduire d'autres inconvénients qui sont traités au travers des simulations à la Sec. 4.9.2.

4.8.4 Les enseignements de l'étude théorique

Les enseignements de l'application du modèle analytique aux trois configurations réseaux illustratives soulignent la prépondérance du *Recall* dans les bénéfices qu'apporte le mécanisme

ALR. Bien que la probabilité de panne ait une influence sur la disponibilité et la consommation de ressources, cette influence est comparable aux comportements observés avec les mécanismes standards de protection et de restauration. Aucune limite n'a pour l'instant été identifiée quant à l'application du dispositif ALR par rapport à certaines probabilités de panne, mais la simulation des mêmes conditions permettrait de vérifier cette observation. Le mécanisme ALR, avec différentes valeurs de *Recall*, de *Precision* et de Δt_p permet, pour une consommation de ressources qui reste similaire à la restauration, de fournir une disponibilité intermédiaire aux mécanismes de protection et de restauration GMPLS. Le dispositif ALR permet, en fonction du *Recall*, de se rapprocher de la disponibilité assurée par la protection, tout en consommant presque trois fois moins de ressources. En particulier, le mécanisme ALR avec un *Recall* performant de 80%, est un compromis très intéressant entre coût de dimensionnement du réseau (CAPEX), utilisation des ressources (OPEX) et disponibilité.

Les performances observées du mécanisme ALR ne plaident pas pour un remplacement du dispositif de protection mais plutôt pour une alternative au mécanisme de restauration. Avec le même dimensionnement de réseau et pratiquement la même consommation de ressources que la restauration, il est possible, grâce à la prédiction de pannes, d'atteindre une disponibilité beaucoup plus compétitive que la restauration, voire même de s'approcher des performances assurées par la protection. Le nombre de fausses prédictions et la durée de validité des prédictions Δt_p ayant une incidence négligeable, notre faveur va vers l'utilisation de méthodes de prédiction permettant de garantir un *Recall* maximum, quels que soient la *Precision* et le Δt_p engendrés.

Néanmoins, lors d'une quantité très importante de fausses prédictions, il est envisageable que des prédictions simultanées viennent perturber le comportement du mécanisme ALR. Ce phénomène n'est pas considéré par ce modèle car nous considérons qu'il n'apparaît que dans de trop rares occasions. Cependant, dans un souci de rigueur, il est préférable d'implémenter le dispositif ALR dans un simulateur afin de valider notre modèle analytique et de déceler d'éventuelles limites dans le comportement de notre dispositif dans des conditions extrêmes.

4.9 Implémentation

L'implémentation du mécanisme ALR dans un simulateur a deux objectifs. Premièrement, de valider le modèle analytique proposé à la Sec. 4.6 en reproduisant les conditions étudiées à la Sec. 4.8. Deuxièmement, de vérifier le comportement du mécanisme dans des conditions extrêmes, c'est-à-dire lorsque le Δt_p est important, ou lorsque les fausses prédictions sont nombreuses (*i.e.* avec une *Precision* faible associée à un *Recall* élevé). Car dans ces conditions, lorsque plusieurs nœuds sont déclarés risqués par le RAM, le module ALR du R&S_DE peut ne pas être en mesure de trouver un chemin de protection non risqué. Dans ce cas, il peut soit choisir un chemin risqué au hasard pour établir un chemin de protection, soit ne pas établir de chemin de protection préventif et se reposer sur le mécanisme de restauration. Dans les deux cas, cela abouti à une indisponibilité plus grande. Nous avons choisi d'implémenter la deuxième stratégie qui ne fait aucun arbitrage entre les nœuds risqués, afin de ne pas perturber les expérimentations avec des décisions aléatoires et ainsi évaluer au mieux les conséquences du pire cas.

4.9.1 Implémentation du simulateur

Contrairement aux Chap. 2 et 3, nous avons choisi d'implémenter notre propre simulateur à événements discrets. Les raisons sont multiples. L'implémentation partielle du protocole GMPLS et l'absence d'implémentation des fonctionnalités de résilience dans les simulateurs couramment utilisés (NS2, NS3, Omnet++, SSFnet++) sont les raisons à l'origine de ce choix. Le problème de passage à l'échelle de ces simulateurs de niveau paquet est une autre raison de ce choix. En effet, en nous intéressant aux pannes qui sont des événements relativement rares, la granularité paquet et le nombre d'événements qui en découle impliquent des temps de simulations très longs, contraignant la durée du temps simulé et le nombre de simulations possibles. Cette granularité paquet n'étant pas critique pour notre étude, nous avons choisi de développer notre propre simulateur avec une granularité au niveau du flux. Les événements sont donc les actions de

modifications des flux telles que les pannes et les modifications apportées lors des prédictions de panne.

Notre simulateur est implémenté en Java. Il ne part pas d'une base vierge puisqu'il se base sur le cœur de l'outil de dimensionnement FAB [PHNP08] pour le routage des flux et la réservation des ressources. Il utilise également une bibliothèque de gestion des nombres aléatoires développée par l'université de Montréal [SSJ11]. Cette bibliothèque nommé SSJ est notamment utilisée pour générer des événements suivant différentes lois de distribution.

4.9.1.1 Gestion des événements

Les événements qui rythment une simulation sont les pannes, les prédictions de pannes (bonnes ou mauvaises), ainsi que les actions de résilience du protocole GMPLS.

Les traces des pannes survenant sur les réseaux d'opérateurs sont des données sensibles qu'il est très difficile d'obtenir. En l'absence de telles données, la théorie générale de la fiabilité [OL09] a été appliquée afin de générer des pannes aléatoires. Le temps entre deux pannes d'un même nœud est supposé suivre une distribution exponentielle dont la moyenne est égale au MTBF souhaité. Cela revient à générer une distribution exponentielle de paramètre $\lambda = 1/MTBF$. De même, on suppose que la durée d'une panne suit une loi log-normale dont la moyenne par nœud est égale au MTTR et l'écart type égale à $0.6{*}MTTR$. La distribution générée pour chaque nœud suit donc une loi $\ln N(\mu, \sigma^2)$ avec $\mu = \log(MTTR) - ((0.5) * \log(1 + ((0.6 * MTTR)^2/MTTR^2)))$ et $\sigma = \sqrt{\log(1 + ((0.6 * MTTR)^2/MTTR^2))}$.

L'autre type d'événement généré au début de la simulation est la prédiction d'une panne. Cela ne concerne que le dispositif ALR. Pour les pannes anticipées, c'est-à-dire les bonnes prédictions, lors de la génération des pannes, nous utilisons une distribution uniforme pour chaque nœud afin de respecter le *ratio* de pannes détectées défini par le *Recall* en fonction du nombre de panne précédemment générées. Une fois cette étape terminée, nous utilisons le nombre de pannes anticipées pour calculer le nombre de fausses prédictions à générer afin de respecter le *ratio* définit par la *Precision* pour chaque nœud. Une distribution uniforme est ensuite utilisée pour répartir ces fausses prédictions uniformément sur toute la période simulée.

À partir de tous ces événements, la simulation traite ensuite chaque événement un par un, afin d'effectuer les actions adéquates. Notre simulateur étant de niveau flux, l'activation des mécanismes de résilience du protocole GMPLS n'est pas une exception. Il est donc nécessaire de générer l'indisponibilité liée à l'activation de ces mécanismes lors d'une panne. Ce temps d'activation est variable car pour chaque panne les temps de re-calcul de route, de signalisation et surtout de détection de la panne puis d'information des nœuds de bordure du LSP sont variables. Pour ces raisons, nous assumons que la protection entraîne une indisponibilité uniformément distribuée entre 40 et 60 millisecondes et que la restauration entraîne une indisponibilité uniformément distribué entre 0,8 et 1,2 seconde. Cette durée d'indisponibilité est générée pendant la simulation, lors de chaque panne en utilisant une distribution uniforme.

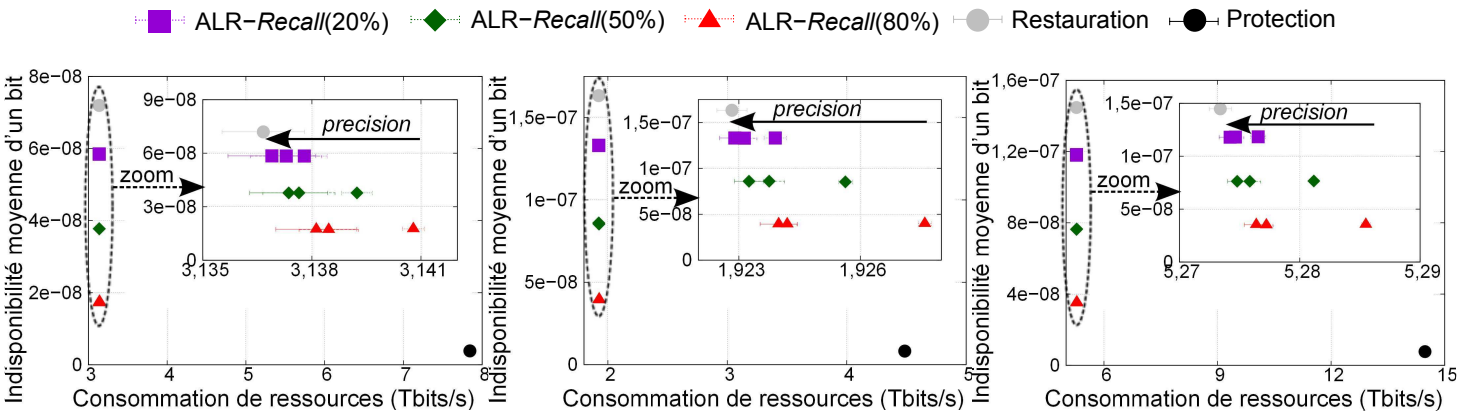
4.9.1.2 Scénario des expérimentations

L'utilisation de notre simulateur de niveau flux a pour avantage une rapidité de simulation bien meilleure que NS3, permettant un nombre et une durée de simulation bien plus importante. Similairement à ce qui a été s fait aux chapitres 2 et 3, les simulations réalisées utilisent les mêmes configurations et conditions que celles définies pour l'application du modèle analytique à la Sec. 4.8. Elles utilisent donc les topologies et matrices de trafic définies à la Sec. 4.7, et chaque configuration appliquée au modèle analytique, ce qui permet d'étudier l'impact du MTBF, du MTTR, du *Recall*, de la *Precision* et du Δt_p sur la disponibilité et sur la consommation de ressources. Contrairement aux deux autres chapitres, chaque simulation est effectuée 100 fois avec une graine différente pour un temps simulé égale à $50 * (MTBF + MTTR)$. L'intervalle de confiance de 99% est affiché sur chaque graphique et atteste d'une précision plus importante des résultats de simulation.

4.9.2 Résultats des simulations

4.9.2.1 Analyse conjointe de la disponibilité et des ressources utilisées

La Fig. 4.17 représente l'indisponibilité et la consommation de ressources des simulations des conditions de référence décrites à la Sec. 4.8 pour les trois topologies. Les combinaisons des trois valeurs de *Recall* et de *Precision* de 20%, 50% et 80% pour le dispositif ALR sont comparées à l'utilisation de la protection et de la restauration. Cette figure confirme les résultats obtenus lors de l'application du modèle analytique aux intervalles de confiance prés. Les enseignements sont donc les mêmes, à savoir, un *ratio* utilisation de ressources / disponibilité meilleur que les mécanismes existants. Pour une consommation de ressources similaire à la restauration, le mécanisme ALR permet d'assurer une disponibilité intermédiaire entre la protection et la restauration, voire même de se rapprocher grandement des performances de la protection avec un *Recall* de 80%. De plus, la simulation de cette première configuration confirme que la *Precision* ne dégrade pas les performances du dispositif ALR en termes d'indisponibilité, pour une valeur de MTBF et de MTTR de respectivement 5000 heures et 5 heures.



(a) A (b) US (c) EU

Figure 4.17: *Ratio* indisponibilité / utilisation des ressources avec les conditions de référence pour les trois topologies.

4.9.2.2 Influence de la probabilité de panne

Pour les mêmes raisons que lors de l'analyse des résultats analytiques, notre analyse n'est illustrée qu'avec une seule topologie, à savoir la topologie européenne. Néanmoins les résultats de simulations complets pour chaque topologie sont disponibles en annexe à la Sec. C.3.

L'intérêt de cette section est de vérifier que le comportement du dispositif ALR est conforme à nos résultats analytiques, notamment dans les valeurs extrêmes, c'est-à-dire pour un MTBF faible (exemple : 1000 heures) et un MTTR élevé (exemple : 10 heures). Pour cela, des simulations sont effectuées avec les conditions étudiées par le modèle analytique, c'est-à-dire un MTBF compris dans une plage de 1000 à 10000 heures, et un MTTR prenant des valeurs allant de 1 à 10 heures.

La Fig. 4.18 confirme le comportement affiché par le modèle analytique, même avec une probabilité de panne importante due à un MTBF de 1000 heures. Il en va de même pour le MTTR qui, comme supposé, n'a pas d'influence sur la disponibilité comme l'illustre la Fig. 4.19.

Les résultats de l'incidence du MTBF et du MTTR sur la consommation de ressources des simulations sont représentés sur la Fig. 4.20 et la Fig. 4.21. Les valeurs mesurées sont similaires aux résultats analytiques et confirment l'intérêt du dispositif ALR dans une plage de probabilité de panne importante.

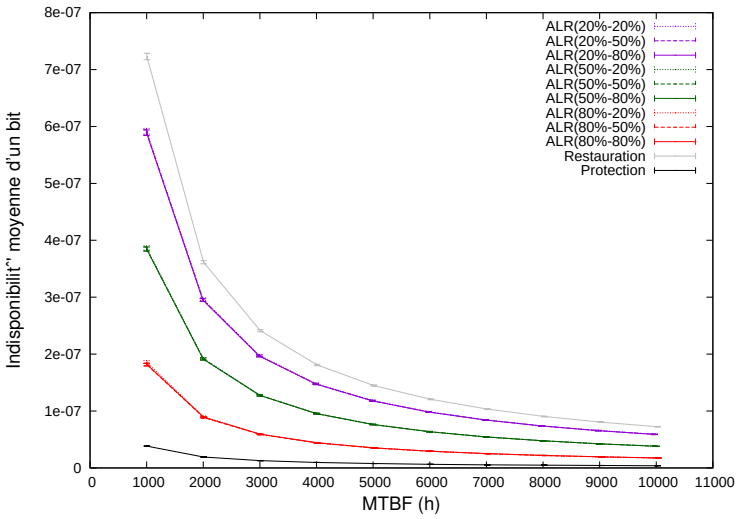


FIGURE 4.18: Impact du MTBF sur la disponibilité avec la topologie européenne.

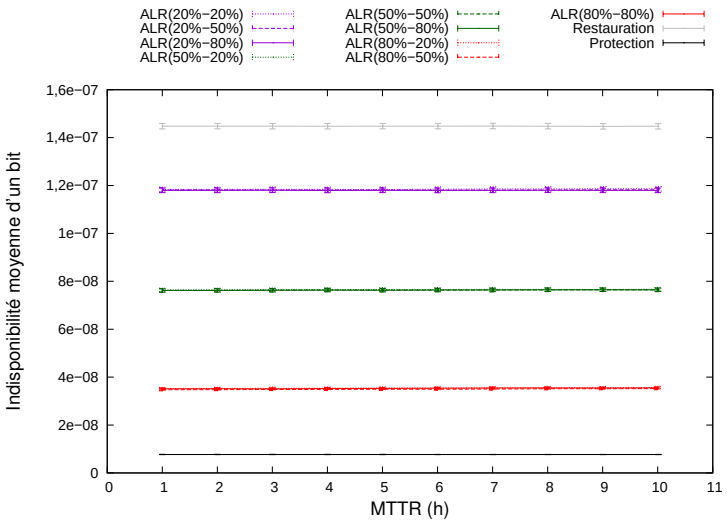


FIGURE 4.19: Impact du MTTR sur la disponibilité avec la topologie européenne.

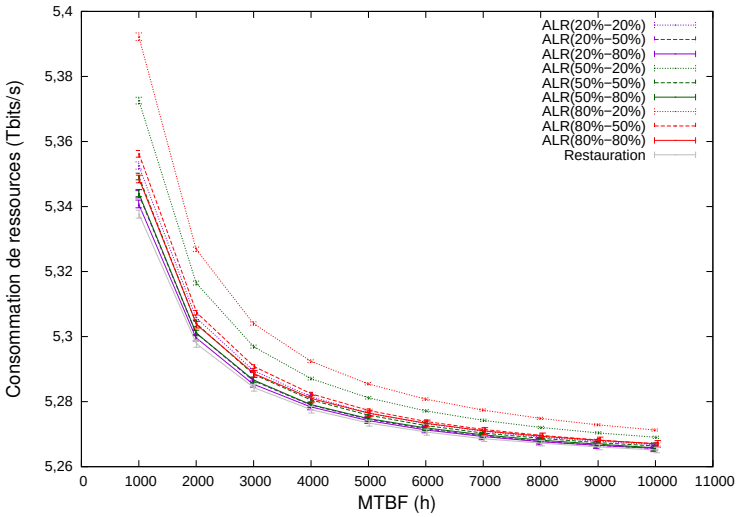


FIGURE 4.20: Impact du MTBF sur l'utilisation des ressources avec la topologie européenne.

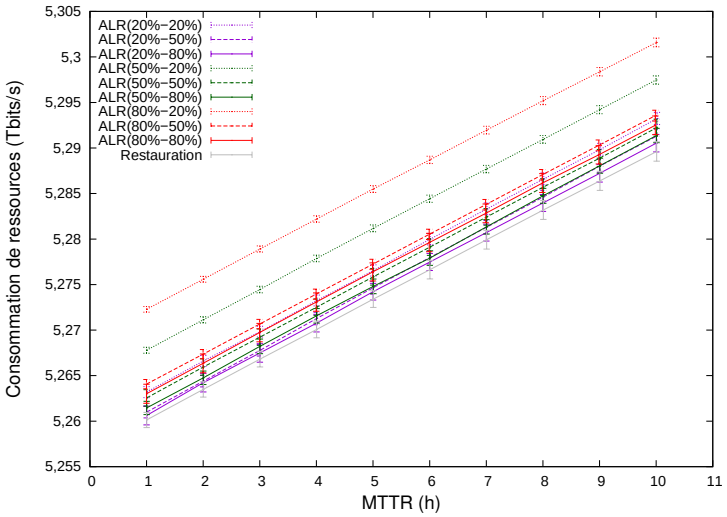


FIGURE 4.21: Impact du MTTR sur l'utilisation des ressources avec la topologie européenne.

4.9.2.3 Les conséquences de la prédiction de pannes

Après avoir eu la confirmation que la probabilité de panne n'influençe pas la stabilité du mécanisme ALR, il est important de valider les caractéristiques de la prédiction de pannes compatibles avec le mécanisme ALR. Pour cela nous simulons la même configuration que celle étudiée avec le modèle analytique.

Avec le *Recall*, on peut noter que le cas $Recall = 0$ n'est pas traité, car dans ce cas de figure, le nombre de fausses prédictions à générer pour respecter la *Precision* définie par l'Eq. (1.2) n'as plus aucun sens. Mis à part cette spécificité, la Fig. 4.22 montre des résultats de disponibilité identiques à la Fig. 4.13 avec des intervalles de confiance de 99% qui viennent renforcer le bien-fondé de notre modèle. En revanche, la situation est légèrement différente pour la consommation de ressources puisque, bien que les résultats soient conformes aux résultats analytiques, une variation plus importante est observée et visible au travers d'intervalles de confiance légèrement plus grand.

La Fig. 4.24 permet de constater que la *Precision* ne joue aucun rôle dans l'indisponibilité. Les fausses prédictions ne viennent pas perturber le mécanisme de création proactive de chemins de protection avant une panne et ce, même dans les cas où la quantité de fausses prédictions est importante, comme c'est le cas avec une *Precision* de 10% et un *Recall* de 80%. L'évolution

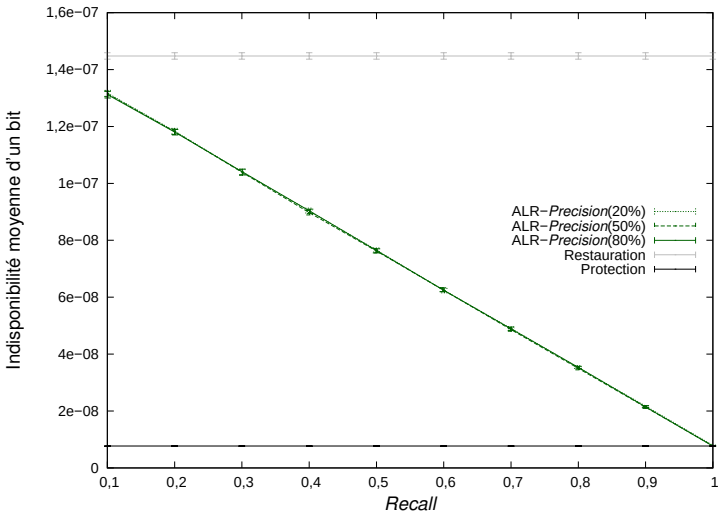


Figure 4.22: Impact du *Recall* sur la disponibilité avec la topologie européenne.

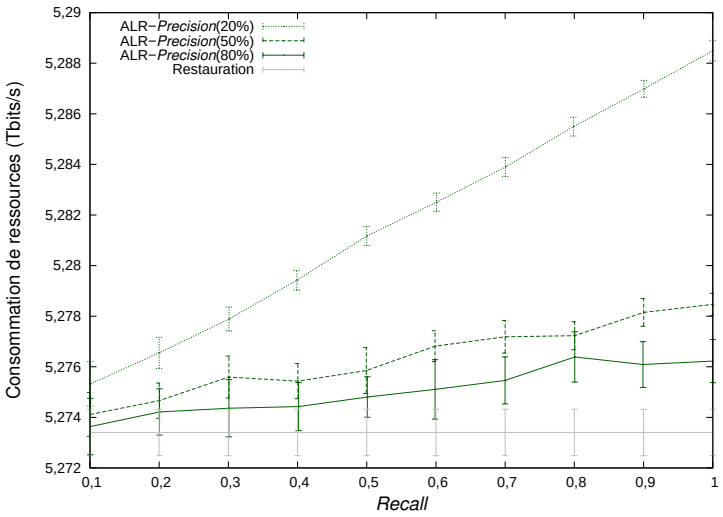


Figure 4.23: Impact du *Recall* sur l'utilisation des ressources avec la topologie européenne.

maîtrisée de la consommation de ressources pour des valeurs de configurations générant de nombreuses prédictions mis en valeur par la Fig. 4.25 suit en tout point les caractéristiques mises en lumière par notre modèle analytique.

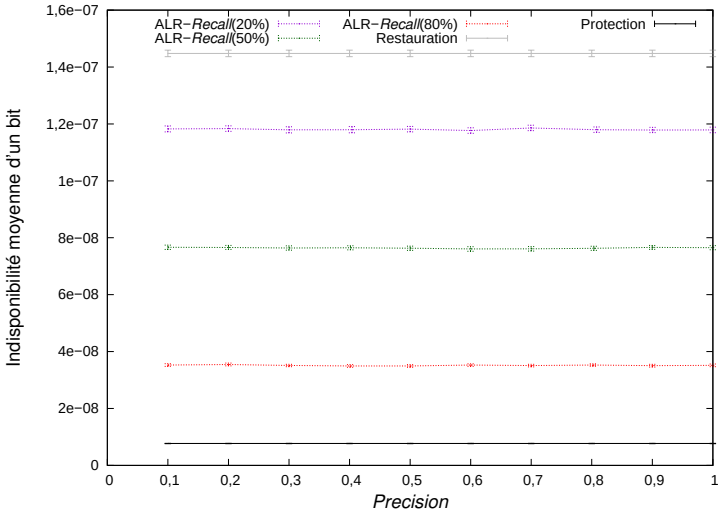


Figure 4.24: Impact de la *Precision* sur la disponibilité avec la topologie européenne.

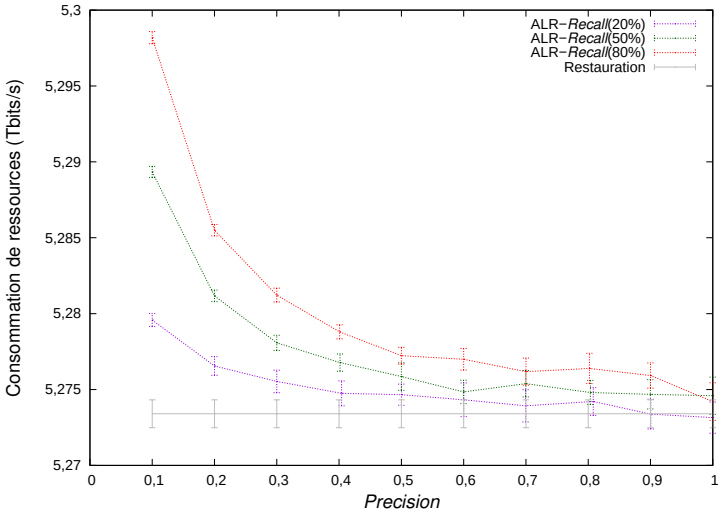


Figure 4.25: Impact de la *Precision* sur l'utilisation des ressources avec la topologie européenne.

Il ne reste plus qu'à s'assurer que la durée de prédiction reste un paramètre sans grande incidence sur le fonctionnement du mécanisme proposé. La consommation de ressources illustrée par la Fig. 4.26 reste logiquement conforme aux observations de notre application numérique.

La Fig. 4.27 diffère légèrement des hypothèses du modèle analytique. Alors que le modèle considère que la durée de validité d'une prédiction ne modifie en rien la disponibilité assuré, on observe une très légère augmentation dans le cas du dispositif ALR possédant un *Recall* de 80%, une *Precision* de 20% et un Δt_p de 10 heures. Cette augmentation d'indisponibilité se justifie par l'impossibilité de choisir un chemin de protection lorsqu'il y a trop de prédictions (notamment de fausses prédictions) et que celles-ci durent trop longtemps. Il est intéressant de noter que cette augmentation est à peine perceptible, même pour le cas d'un Δt_p extrême de 10 heures et d'une *Precision* de seulement 20%. L'impact est si faible que les performances de disponibilité restent meilleures que celles assurées par le mécanisme de restauration, ainsi que celles assurées par les mécanismes ALR avec des *Recall* inférieurs à 80%. Cette simulation permet donc de valider les hypothèses du modèle analytique, et de s'assurer que le dispositif

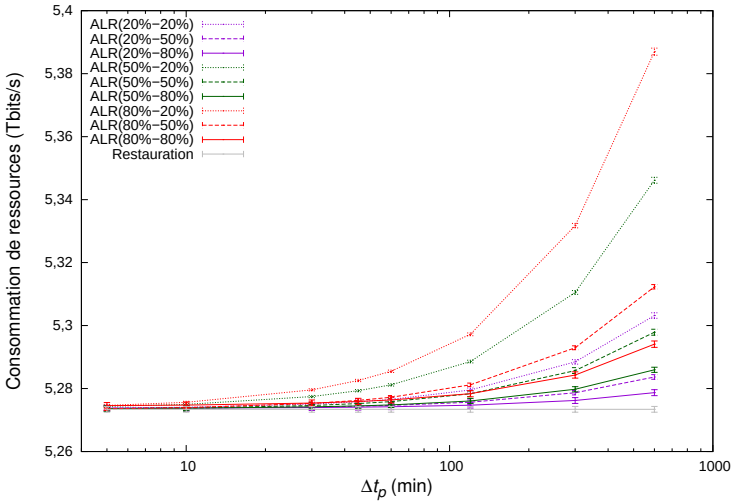


FIGURE 4.26: Impact de Δt_p sur l'utilisation des ressources avec la topologie européenne.

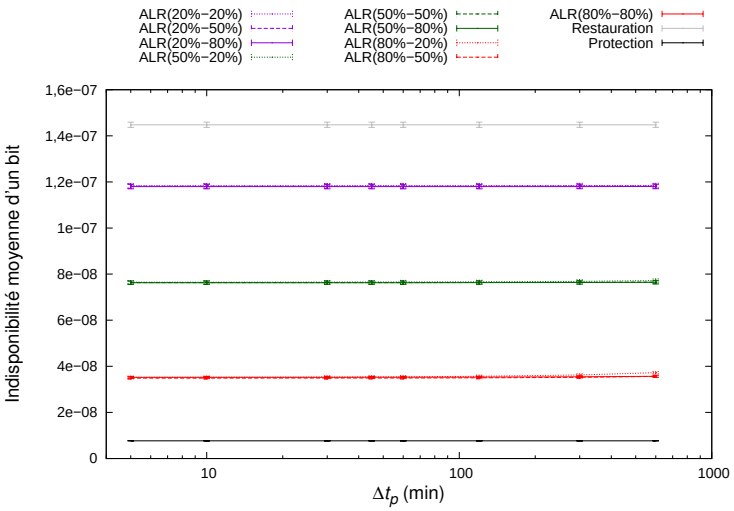


FIGURE 4.27: Impact de Δt_p sur la disponibilité avec la topologie européenne.

ALR reste stable pour les plages de paramètres testées.

4.9.2.4 Les enseignements apportés par la simulation

Contrairement aux simulations des chapitres précédents, l'utilisation d'un simulateur de niveau flux permet de simuler un temps beaucoup plus long et d'effectuer un nombre important d'essais. Les résultats obtenus bénéficient donc d'une précision accrue, directement visible par l'intermédiaire des intervalles de confiance de 99%. De manière générale, les résultats de simulations attestent de l'exactitude de notre modèle analytique. L'interprétation de ces résultats confirme l'intérêt du mécanisme ALR pour fournir des performances bien supérieures à la restauration pour un coup presque identique, mettant en évidence un *ratio* disponibilité/consommation de ressources supérieur aux solutions existantes.

De plus, la simulation permet de vérifier le bon fonctionnement et la stabilité du mécanisme ALR couplé avec une prédiction de pannes médiocre, notamment lorsque plusieurs routeurs sont déclarés risqués au même moment, suite à des fausses prédictions. Fort de cette dernière vérification, l'enseignement à retenir est le lien fort entre performance de disponibilité et *Recall*, validant l'importance du *Recall* comme critère numéro un dans le choix d'implémentation de la fonction de prédiction de pannes.

4.10 Conclusion

Le dispositif d'autoréparation proactif développé dans ce chapitre propose de modifier dynamiquement le mécanisme de résilience d'un LSP en alternant entre la protection et la restauration en fonction du risque de panne. Une telle opération rompt avec le mode de fonctionnement actuel complètement statique, en exploitant la rapidité d'exécution des équipements pour modifier le mécanisme de résilience avant qu'une panne n'apparaisse, en réponse à une prédiction de panne. Une telle fonctionnalité permet d'améliorer grandement la disponibilité des LSP pour un coûts très en deçà de ce que nécessite la protection. Sans être aussi perturbant qu'une modification du routage tel que le dispositif proposé au Chap. 3, le mécanisme ALR profite de la haute disponibilité assurée par les mécanismes de protection pour améliorer la disponibilité sans perturber le trafic. L'inconvénient de ce mécanisme est la consommation de ressources supplémentaires dédiées à la protection qu'engendre une prédiction et notamment une fausse prédiction. Néanmoins les ressources de protections allouées dynamiquement par le dispositif d'autoréparation ALR ne sont allouées que pour une période limitée au maximum égale à Δt_p. De plus, la prédiction de panne permettant de connaître à l'avance la panne, le chemin de protection n'as pas besoin d'être complètement disjoint du LSP principal mais n'a comme seule contrainte que l'exclusion des ressources risquées. Ce détail permet une efficacité en gestion des ressources, similaire à la restauration, et donc bien meilleure que la protection. Une modélisation analytique ainsi qu'une implémentation dans un simulateur à événements discrets ont permis de vérifier l'avantage de ce mécanisme par rapport aux deux types de mécanismes de résilience standards de GMPLS que sont la protection et la restauration. Les résultats de l'étude mettent en évidence la très faible incidence des fausses prédictions sur la consommation de ressources ou sur la disponibilité. La prédiction de pannes optimale compatible avec le dispositif de résilience adaptative est une prédiction possédant le plus haut *Recall*, quelles que soit la *Precision* ou la durée de validité des prédictions Δt_p. Ce dispositif ne permet pas d'assurer une disponibilité comparable à la protection, mais est une amélioration importante au système de restauration, en apportant de manière autonome une gestion plus efficace des pannes, lorsque les symptômes de la panne permettent de l'anticiper de quelques secondes.

Conclusion générale et perspectives

Les contributions

L'objectif de cette thèse, qui s'est déroulée en milieu industriel chez le constructeur majeur d'équipements de réseaux qu'est Alcatel-Lucent, ést de proposer des mécanismes de gestion des pannes exploitant les principes de la gestion autonome des réseaux, tout en étant facilement intégrable et déployable dans les réseaux d'opérateurs. En effet, même si une redéfinition complète des réseaux pourrait paraître la meilleure solution, les contraintes économiques interdisent un tel scénario. L'amélioration de la gestion des réseaux par l'autonomie nécessite donc de prendre en compte les technologies actuellement déployées afin de convaincre les opérateurs de sa pertinence. La gestion de panne n'échappe pas à cette règle, devenue de plus en plus complexe, elle nécessite des délais de réaction extrêmement rapides dont ne sont pas capables les opérateurs humains, ce qui en fait un candidat de choix pour l'introduction de fonctions d'autoréparation autonomes. En effet, lors d'une panne, chaque seconde est comptée, la rapidité d'analyse et d'exécution que permet un équipement réseau rendrait possible une meilleure gestion des pannes et une meilleure qualité de service pour les utilisateurs.

La première tâche a donc été d'étudier la gestion actuelle des réseaux, ainsi que les architectures de gestion autonome afin de proposer les blocs architecturaux permettant l'exploitation des possibilités offertes avec la gestion autonome pour la gestion des pannes. Compte tenu de l'importance de la gestion des pannes dans la préoccupation des opérateurs et des contraintes fortes qu'elle implique, la définition des blocs architecturaux spécifiques à la gestion des pannes est indispensable pour répondre correctement aux enjeux. L'architecture UAFAReS ainsi proposée dans cette thèse est intégrée au sein de l'architecture complète de gestion autonome GANA définie par le projet EFIPSANS. Cette architecture construite autour de la notion d'éléments de décision, d'entités gérées et de boucles de rétroaction hiérarchiques, de par ses principes volontairement simples et modulables, permet d'instancier toutes les fonctions nécessaires à la gestion de réseau et notamment la gestion des pannes. L'architecture UAFAReS de gestion des pannes qui est principalement constitué des deux éléments de décision que sont le FM_DE [1] et le R&S_DE [2] se démarque des autres propositions d'architecture de gestion autonome des pannes en attachant une importance toute particulière à l'autoréparation proactive. En effet, un module RAM [3] intégré au sein des R&S_DE a pour fonction d'évaluer le risque de panne des équipements réseaux en utilisant des dispositifs de prédiction de pannes pour en informer le réseau. Cela permet un nouveau type de mécanisme d'autoréparation, qui agit de manière préventive, quelques secondes avant la panne, pour préparer au mieux le réseau à son apparition, afin de diminuer son impact sur le trafic et ainsi améliorer l'expérience des utilisateurs. En s'appuyant sur cette nouvelle fonctionnalité, les éléments de décision en charge du routage tels que le RM_DE ou des dispositifs de résilience tel que le R&S_DE sont alors capables d'exploiter au mieux une telle information pour assurer les fonctions d'autoréparation proactives proposées dans cette thèse.

Le premier mécanisme d'autoréparation instanciant cette architecture s'attèle à améliorer la disponibilité dans les réseaux IP. Le mécanisme de restauration IP opérant lors de l'occurrence d'une panne engendre une interruption de service néfaste pour le trafic sensible à la QoS. Le temps mis par le processus de convergence du protocole de routage à état de liens est respon-

1. *Fault-Management Decision Element*
2. *Resilience and Survivability Decision Element*
3. *Risk Assessment Module*

sable de cette indisponibilité, et l'étape de détection de la panne en constitue une part très importante. Le mécanisme de détection dynamique des pannes (*i.e.* AFDT [1]) utilise le risque de panne en temps réel fourni par l'architecture proposée au premier chapitre afin d'accélérer temporairement l'envoi des messages *Hello* de détection de pannes en provenance des équipements risqués afin d'abaisser l'interruption en dessous d'une seconde tout en garantissant la stabilité du réseau. En fonction de la probabilité de panne et des caractéristiques de la prédiction de pannes, les gains en disponibilité, ainsi que la quantité de messages *Hello* responsables de l'instabilité ont été quantifiés de manière théorique, puis de manière pratique *via* l'implémentation du mécanisme dans un simulateur. Le dispositif permet de diminuer l'indisponibilité lorsqu'une panne est anticipé, mais ne permet pas de la supprimer entièrement. En revanche, il présente le grand intérêt d'être très peu sensible aux performances de la prédiction de pannes. En effet, le mécanisme a pour unique cible la fréquence d'envoi des messages de détection de pannes *Hello*, ce qui est relativement peu perturbateur pour le réseau. L'effet des fausses prédictions ou d'une durée de validité de prédiction Δt_p longue reste maîtrisée ce qui fait du mécanisme AFDT le candidat idéal pour réduire l'indisponibilité lorsque la prédiction de pannes n'est pas précise.

Pour éviter l'indisponibilité lors d'une panne, il est nécessaire de modifier directement le routage du trafic de manière proactive et non réactive. Le mécanisme de routage sensible aux pannes (*i.e.* RAR [2]) proposé au Chap. 3 se sert de l'information de risque de panne pour modifier dynamiquement les métriques de routage afin de repousser le trafic hors des éléments susceptibles de tomber en panne. L'information de risque de panne permet de dégager une fenêtre de temps pour préventivement reconfigurer le routage vers une situation similaire à ce qu'il sera après la panne. Cela a pour avantage d'assurer la continuité de service pour les pannes anticipées, mais les fausses prédictions augmentent le nombre d'oscillations de routage et peuvent même créer de la congestion. Le modèle analytique proposé permet de quantifier les gains attendus pour la disponibilité et d'évaluer les conséquences des fausses prédictions sur la stabilité du routage. De plus, la simulation du dispositif permet d'évaluer les conséquences sur la congestion. Le reroutage proactif permet de diminuer le risque globale de panne du trafic transitant sur le réseau et ainsi de réduire l'indisponibilité occasionnée par les pannes de manière plus importante que le précédent mécanisme. Outre la stabilité qui peut être dégradée lors de trop nombreuses fausses prédictions, la principale limitation vient de la congestion. Avec de trop nombreuses fausses prédictions et/ou une période de validité Δt_p trop longue, le reroutage engendre de la congestion qui vient ruiner tous les efforts faits pour réduire l'indisponibilité et amène des pertes de paquets dans des proportions inacceptables. L'utilisation du dispositif est donc restreinte à l'utilisation d'une prédiction de pannes précise afin de profiter pleinement des avantages de cette reconfiguration proactive. Enfin, l'implémentation d'un prototype a permis de vérifier la faisabilité du concept dans un environnement expérimental représentatif des réseaux réels.

Le problème de congestion est propre aux réseaux IP. Cependant l'utilisation d'un environnement connecté, tel que GMPLS, permet de mieux gérer ce problème. En effet, le protocole GMPLS fournit un ensemble de dispositifs de résilience qui permet d'envisager une approche plus fine. Le mécanisme de résilience adaptative (*i.e.* ALR [3]) proposé à la fin de cette thèse exploite l'information de risque pour alterner entre le dispositif de restauration peu coûteux en ressources lorsqu'aucun risque n'est détecté, et le dispositif de protection pour restaurer le trafic en un temps minimum lorsque le risque de panne est élevé. Le flux de trafic principal n'est pas altéré, seul le mécanisme de protection est modifié. Le dispositif permet d'utiliser peu de ressources lorsque cela n'est pas nécessaire, et lorsqu'une panne est anticipée, la protection permet un rétablissement assurant une disponibilité forte. La disponibilité et la consommation de ressources sont les critères qui ont été étudiés dans un premier temps de manière théorique avec un modèle analytique, puis de manière expérimentale avec l'implémentation du dispositif ALR dans un simulateur. Compte tenu de l'incertitude de la prédiction de pannes, le mécanisme ALR ne permet pas d'assurer sur le long terme une disponibilité comparable à la protection. Néanmoins, il fournit une alternative intéressante à la restauration. En effet, pour un coût si-

1. *Adaptive Failure Detection Timers*
2. *Risk-Aware Routing*
3. *Adaptive Level of Recovery*

milaire à la restauration, il permet de rétablir les pannes anticipées de manière beaucoup plus efficace et donc de fournir une meilleure disponibilité. Enfin, il est compatible avec la plupart des dispositifs de prédiction de pannes car peu sensible à la prédiction de pannes faiblement précise.

La complexité de la gestion de réseau, et encore plus de la gestion des pannes, rend indispensable l'utilisation de plus d'autonomie dans les réseaux. Les solutions proposées répondent à ce problème en s'appuyant sur les points forts des systèmes autonomes, en particulier sur la vitesse de réaction qui est fondamentale dans la gestion des pannes.

Il est alors possible de mieux gérer les pannes en agissant de manière proactive en fonction de l'état du réseau. Les trois mécanismes d'autoréparation proactive proposent une meilleure gestion des pannes pour les protocoles IP et GMPLS, les technologies actuellement les plus utilisées par les opérateurs, avec des actions proactives. L'utilité d'introduire de l'autonomie dans la gestion des pannes, notamment par l'intermédiaire de mécanismes proactifs, reste valide quel que soit l'environnement technologique, à la fois pour un grand nombre d'anciennes technologies de réseau, mais sans aucun doute aussi pour les technologies futures.

À plus long terme

Le travail de cette thèse a contribué à l'élaboration de trois dispositifs autonomes pour la gestion des pannes. Néanmoins, des étapes sont nécessaires avant un déploiement à grande échelle par les opérateurs.

Premièrement, un processus de standardisation est nécessaire dans certain cas. Il a été souligné le fait que le mécanisme AFDT requière une étape de standardisation auprès de l'IETF afin de permettre au protocole *Hello* l'utilisation d'une période inférieure à une seconde, ainsi que l'utilisation d'une fréquence spécifique pour chaque voisin. Le dispositif ALR ne requière pas obligatoirement d'extension du protocole GMPLS, mais cette option est possible et mériterait d'être considérée comme il l'a été brièvement évoqué. Plus généralement, le déploiement à large échelle des trois mécanismes d'autoréparation nécessite l'utilisation d'un module d'évaluation du risque, ainsi qu'un module de contrôle du mécanisme, tels ceux proposés par l'architecture UAFAReS. Or leur utilisation ne peut se faire qu'après leur adoption par les opérateurs et les constructeurs grâce la standardisation de l'architecture. Dans tous les cas, la définition précise de propositions d'extensions protocolaires a peu été traitée dans cette thèse et nécessiterait d'être traitée de façon exhaustive afin de rendre le déploiement de nos contributions envisageable dans les réseaux opérationnels des opérateurs.

Le deuxième point concerne la prédiction de pannes. Nous avons vu que de nombreux travaux se sont intéressés à la prédiction de pannes. Or celle-ci est aujourd'hui présente dans peu de produits industriels. Il est donc nécessaire de passer à l'étape suivante afin de concevoir et d'intégrer des mécanismes de prédiction de pannes au sein des équipements réseaux et d'en étudier les performances sur des réseaux opérationnels représentatifs. Ce dernier point est essentiel, puisque les opérateurs étant très frileux dans la divulgation des informations relatives aux pannes survenant sur leurs réseaux, une phase de test de la fonction de prédiction de pannes sur leurs réseaux opérationnels est nécessaire, d'une part pour obtenir des résultats, et d'autre part pour donner à ces résultats le crédit nécessaire pour convaincre les opérateurs de leur validité. À partir de ces résultats, les constructeurs et les opérateurs seront alors en mesure de décider ou non d'utiliser les mécanismes d'autoréparation proposés en fonctions des paramètres de la prédiction de pannes observées.

Enfin, la mise en œuvre de nos dispositifs dans des réseaux opérationnels requière une dernière étape d'expérimentation en environnement réel de la combinaison entre un module d'évaluation des risques temps réel exhaustif intégré au sein des équipements et nos dispositifs d'autoréparations proactifs, afin de vérifier en environnement réel, que le comportement du système global est conforme aux enseignements apportés par cette thèse. En effet, nous avons pris soin d'être le plus précis et méticuleux possible afin d'évaluer les coûts et les gains des trois mécanismes proposés, mais les mécanismes s'appliquant aux systèmes critiques que sont les réseaux de cœurs, des tests du système global sur des réseaux réels de classes opérateurs sont indispensables pour

obtenir la confiance des opérateurs dans des dispositifs ne nécessitant plus leurs contrôles.

L'exploitation d'une information de risque de panne en temps réel ne se restreint pas aux trois mécanismes proposés et a vocation à être utilisée de manière plus large. Cela constitue de nouveaux axes de recherche qui méritent d'être explorés. Par exemple, toujours dans le domaine du routage IP, certains routeurs possèdent un rôle plus important que d'autres. Il s'agit notamment des routeurs de bordure d'aire de routage (*i.e.* ABR), des routeurs de bordure de système autonome (*i.e.* ASBR), ou des routeurs de bordure BGP. Il est alors possible d'utiliser l'information de risque de panne pour proactivement changer les rôles des routeurs [RNP+11] pour que ceux-ci s'exécutent sur des éléments de réseaux non risqués. Cette stratégie peut être facilement transposée dans d'autres d'environnements réseaux. L'IMS est un bon exemple puisque son architecture est constituée d'éléments ayant chacun un rôle bien particulier (P-CSF [1], I-CSF [2], HSS [3], S-CSF [4], MGW [5], MGCF [6], etc.), tout comme la mobilité IP où le *home agent* et le *foreign agent* sont deux éléments ayant chacun un rôle bien spécifique.

De plus, l'évaluation continue du risque de pannes peut constituer une information précieuse dans le processus de recherche de la cause originale d'une panne. En effet, le processus de localisation de la source d'une panne est un processus complexe. Qu'il soit effectué par des opérateurs humains ou de manière autonome par le réseau, il s'effectue après l'occurrence de la panne, dans un environnement dégradé, ce qui rend la tâche plus difficile. L'évaluation du risque de pannes par son action proactive permet d'obtenir une information qui est précieuse et permettrait d'améliorer le processus de recherche de l'origine de la panne.

Enfin, dans le cas d'une réparation de la panne de manière autonome, il est utile de s'assurer que l'intervention a bien permis de régler le problème et ne risque pas d'en créer de nouveaux. L'information de risque de panne peut alors être utilisé afin de s'assurer qu'aucune panne n'est sur le point d'apparaître et donc que la réparation a bien été effectuée et ce sans engendrer de vice caché.

Il ne fait aucun doute que l'observation continue du réseau afin de fournir une information de risque de panne en continue constituerait un catalyseur pour le développement de nouveaux dispositifs autonomes permettant une bien meilleure gestion des pannes. Il nous semble donc maintenant qu'il est temps de doter les équipements réseaux de cette fonctionnalité.

1. *Proxy-Call Session Control Function*
2. *Interrogating-Call Session Control Function*
3. *Home Subscriber Server*
4. *Serving-Call Session Control Function*
5. *Media Gateway*
6. *Media Gateway Controller Function*

Annexe A

Topologies et matrices de trafic

Sommaire

A.1 Topologie allemande

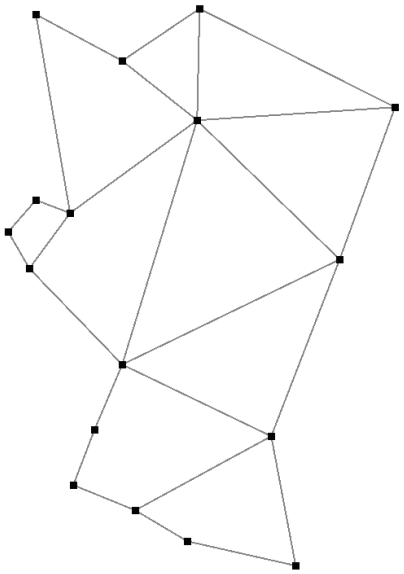


FIGURE A.1: Topologie Allemande.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1							x	x			x						
2							x	x						x			
3					x			x		x				x			
4					x					x							
5			x	x													
6								x		x	x	x			x		
7	x	x						x									
8	x	x	x			x	x				x						
9												x				x	
10			x	x		x											
11	x					x		x							x		
12						x			x								
13															x		x
14		x	x														
15						x					x		x			x	
16									x						x		x
17													x			x	

TABLE A.1: Topologie allemande.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	0	0	0	0	0	306	298	0	0	174	0	0	0	0	0	0
2	0	0	0	0	0	0	114	120	0	0	0	0	0	144	0	0	0
3	0	0	0	0	37	0	0	208	0	88	0	0	0	278	0	0	0
4	0	0	0	0	36	0	0	0	0	41	0	0	0	0	0	0	0
5	0	0	37	36	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	316	0	182	353	85	0	0	224	0	0
7	306	114	0	0	0	0	0	157	0	0	0	0	0	0	0	0	0
8	298	120	208	0	0	316	157	0	0	0	258	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	64	0	0	0	54	0
10	0	0	88	41	0	182	0	0	0	0	0	0	0	0	0	0	0
11	174	0	0	0	0	353	0	258	0	0	0	0	0	0	275	0	0
12	0	0	0	0	0	85	0	0	64	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	179	0	143
14	0	144	278	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	224	0	0	0	0	275	0	179	0	0	187	0
16	0	0	0	0	0	0	0	0	54	0	0	0	0	0	187	0	86
17	0	0	0	0	0	0	0	0	0	0	0	0	143	0	0	86	0

TABLE A.2: Métriques avec la topologie allemande.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	0	0	0	0	0	48,83	68,36	0	0	97,66	0	0	0	0	0	0
2	0	0	0	0	0	0	48,83	97,66	0	0	0	0	0	68,36	0	0	0
3	0	0	0	0	107,42	0	0	156,25	0	146,48	0	0	0	117,19	0	0	0
4	0	0	0	0	78,12	0	0	0	0	107,42	0	0	0	0	0	0	0
5	0	0	107,42	78,12	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	185,55	0	234,38	87,89	185,55	0	0	107,42	0	0
7	48,83	48,83	0	0	0	0	0	68,36	0	0	0	0	0	0	0	0	0
8	68,36	97,66	156,25	0	0	185,55	68,36	0	0	0	156,25	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	166,02	0	0	0	156,25	0
10	0	0	146,48	107,42	0	234,38	0	0	0	0	0	0	0	0	0	0	0
11	97,66	0	0	0	0	87,89	0	156,25	0	0	0	0	0	0	166,02	0	0
12	0	0	0	0	0	185,55	0	0	166,02	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97,66	0	68,36
14	0	68,36	117,19	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	107,42	0	0	0	0	166,02	0	97,66	0	0	126,95	0
16	0	0	0	0	0	0	0	0	156,25	0	0	0	0	0	126,95	0	107,42
17	0	0	0	0	0	0	0	0	0	0	0	0	68,36	0	0	107,42	0

TABLE A.3: Capacités avec la topologie allemande (Gbits/s).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	0	0	0	0	0	58,59	97,66	0	0	126,95	0	0	0	0	0	0
2	0	0	0	0	0	0	68,36	126,95	0	0	0	0	0	117,19	0	0	0
3	0	0	0	0	146,48	0	0	185,55	0	244,14	0	0	0	117,19	0	0	0
4	0	0	0	0	146,48	0	0	0	0	146,48	0	0	0	0	0	0	0
5	0	0	146,48	146,48	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	263,67	0	273,44	107,42	214,84	0	0	136,72	0	0
7	58,59	68,36	0	0	0	0	0	87,89	0	0	0	0	0	0	0	0	0
8	97,66	126,95	185,55	0	0	263,67	87,89	0	0	0	156,25	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	214,84	0	0	0	214,84	0
10	0	0	244,14	146,48	0	273,44	0	0	0	0	0	0	0	0	0	0	0
11	126,95	0	0	0	0	107,42	0	156,25	0	0	0	0	0	0	185,55	0	0
12	0	0	0	0	0	214,84	0	0	214,84	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	126,95	0	126,95
14	0	117,19	117,19	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	136,72	0	0	0	0	185,55	0	126,95	0	0	166,02	0
16	0	0	0	0	0	0	0	0	214,84	0	0	0	0	0	166,02	0	126,95
17	0	0	0	0	0	0	0	0	0	0	0	0	126,95	0	0	126,95	0

TABLE A.4: Capacités avec la topologie allemande avec la protection GMPLS (Gbits/s).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	3,04	3,64	4,03	2,83	12,69	7,39	7,78	1,21	4,30	14,89	2,14	4,58	0	3,83	5,05	3,04
2	3,04	0	2,16	2,22	1,64	5,87	5,27	5,33	0,52	2,28	4,05	0,95	1,75	0	1,46	2,12	1,24
3	3,64	2,16	0	7,51	8,24	10,37	3,88	5,28	0,89	6,76	5,33	1,68	2,61	0	2,24	3,44	1,93
4	4,03	2,22	7,51	0	9,68	12,01	4,18	5,27	1,06	13,50	5,84	1,99	3,03	0	2,54	4,05	2,25
5	2,83	1,64	8,24	9,68	0	8,29	3,00	3,77	0,71	6,43	4,11	1,33	2,07	0	1,75	2,74	1,53
6	12,69	5,87	10,37	12,01	8,29	0	11,70	14,22	4,33	13,60	18,98	9,80	10,71	57,44	9,75	15,94	8,50
7	7,39	5,27	3,88	4,18	3,00	11,70	0	9,87	1,08	4,35	9,08	1,93	3,76	0	3,11	4,41	2,60
8	7,78	5,33	5,28	5,27	3,77	14,22	9,87	0	1,28	5,45	11,14	2,34	4,29	0	3,72	5,22	3,02
9	1,21	0,52	0,89	1,06	0,71	4,33	1,08	1,28	0	1,21	1,84	1,54	1,28	0	1,07	3,32	1,22
10	4,30	2,28	6,76	13,50	6,43	13,60	4,35	5,45	1,21	0	6,30	2,31	3,32	0	2,81	4,55	2,50
11	14,89	4,05	5,33	5,84	4,11	18,98	9,08	11,14	1,84	6,30	0	3,31	7,03	0	6,50	7,78	4,70
12	2,14	0,95	1,68	1,99	1,33	9,80	1,93	2,34	1,54	2,31	3,31	0	2,12	0	1,90	4,35	1,88
13	4,58	1,75	2,61	3,03	2,07	10,71	3,76	4,29	1,28	3,32	7,03	2,12	0	0	4,68	6,17	4,88
14	0	0	0	0	0	57,44	0	0	0	0	0	0	0	0	0	0	0
15	3,83	1,46	2,24	2,54	1,75	9,75	3,11	3,72	1,07	2,81	6,50	1,90	4,68	0	0	5,08	3,28
16	5,05	2,12	3,44	4,05	2,74	15,94	4,41	5,22	3,32	4,55	7,78	4,35	6,17	0	5,08	0	7,58
17	3,04	1,24	1,93	2,25	1,53	8,50	2,60	3,02	1,22	2,50	4,70	1,88	4,88	0	3,28	7,58	0

TABLE A.5: Matrice de trafic avec la topologie allemande (Gbits/s).

A.2 Topologie américaine (US)

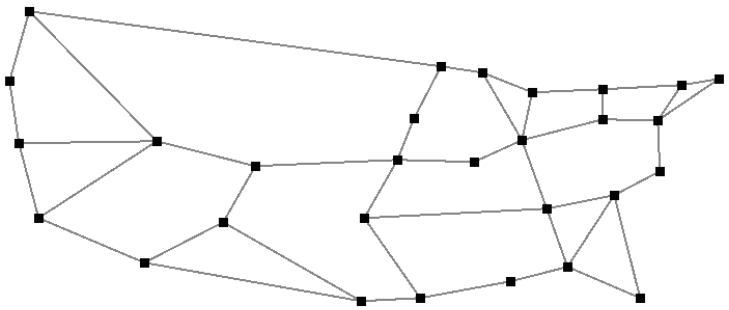


Figure A.2: Topologie américaine (US).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
1		x	x										x																
2	x		x		x																								
3	x	x		x																									
4			x		x		x																						
5		x		x		x																							
6					x		x	x				x					x												
7				x		x						x																	
8						x			x																				
9								x		x										x				x					
10									x		x																		
11										x		x																	x
12						x	x				x																		
13		x												x															
14													x		x	x	x												
15														x		x													
16														x	x		x	x											
17						x								x		x				x									
18																x			x										
19																		x		x	x								
20									x								x		x										
21																			x			x	x						
22																					x		x			x			
23																					x	x		x					
24									x														x		x				
25																								x		x	x		x
26																						x			x		x		
27																									x	x		x	
28																											x		x
29											x														x			x	

Table A.6: Topologie américaine (US).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
1	0	464	307	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	464	0	315	0	491	0	0	0	0	0	0	0	514	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	307	315	0	600	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	600	0	378	0	570	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	491	0	378	0	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	585	0	475	311	0	0	0	486	0	0	0	0	523	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	570	0	475	0	0	0	0	0	497	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	311	0	0	462	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	462	0	491	0	0	0	0	0	0	0	0	0	546	0	0	0	941	0	0	0	0	0
10	0	0	0	0	0	0	0	0	491	0	487	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	487	0	247	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2539
12	0	0	0	0	0	486	497	0	0	0	247	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	514	0	0	0	0	0	0	0	0	0	0	0	311	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	311	0	715	614	487	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	715	0	527	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	614	527	0	447	352	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	523	0	0	0	0	0	0	0	487	0	447	0	0	0	1119	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	352	0	0	565	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	565	0	791	387	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	546	0	0	0	0	0	0	0	1119	0	791	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	387	0	0	1539	1371	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1539	0	736	0	0	747	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1371	736	0	418	0	0	0	0	0
24	0	0	0	0	0	0	0	0	941	0	0	0	0	0	0	0	0	0	0	0	0	0	418	0	687	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	687	0	1046	915	0	1138
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	747	0	0	1046	0	564	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	915	564	0	380	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	380	0	474
29	0	0	0	0	0	0	0	0	0	0	2539	0	0	0	0	0	0	0	0	0	0	0	0	0	1138	0	0	474	0

TABLE A.7: Métriques avec la topologie américaine (US).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
1	0	19,53	19,53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	19,53	0	48,83	0	87,89	0	0	0	0	0	0	0	68,36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	19,53	48,83	0	58,59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	58,59	0	58,59	0	48,83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	87,89	0	58,59	0	97,66	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	97,66	0	48,83	97,66	0	0	0	39,06	0	0	0	0	58,59	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	48,83	0	48,83	0	0	0	0	0	39,06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	97,66	0	0	97,66	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	97,66	0	39,06	0	0	0	0	0	0	0	0	0	0	29,30	0	0	0	87,89	0	0	0	0
10	0	0	0	0	0	0	0	0	39,06	0	39,06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	39,06	0	58,59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39,06
12	0	0	0	0	0	39,06	39,06	0	0	0	58,59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	68,36	0	0	0	0	0	0	0	0	0	0	0	68,36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	68,36	0	29,30	39,06	39,06	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	29,30	0	29,30	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	39,06	29,30	0	39,06	78,12	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	58,59	0	0	0	0	0	0	0	39,06	0	39,06	0	0	0	29,30	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	78,12	0	0	68,36	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	68,36	0	29,30	68,36	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	29,30	0	0	0	0	0	0	0	29,30	0	29,30	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	68,36	0	0	48,83	29,30	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48,83	0	39,06	0	0	48,83	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	29,30	39,06	0	48,83	0	0	0	0	0
24	0	0	0	0	0	0	0	0	87,89	0	0	0	0	0	0	0	0	0	0	0	0	0	48,83	0	68,36	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	68,36	0	39,06	29,30	0	19,53
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48,83	0	0	39,06	0	19,53	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	29,30	19,53	0	29,30	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	29,30	0	29,30
29	0	0	0	0	0	0	0	0	0	0	39,06	0	0	0	0	0	0	0	0	0	0	0	0	0	19,53	0	0	29,30	0

TABLE A.8: Capacités avec la topologie US (Gbits/s).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
1	0	29,30	29,30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	29,30	0	78,12	0	87,89	0	0	0	0	0	0	0	87,89	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	29,30	78,12	0	78,12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	78,12	0	58,59	0	48,83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	87,89	0	58,59	0	117,19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	117,19	0	29,30	117,19	0	0	0	29,30	0	0	0	0	87,89	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	48,83	0	29,30	0	0	0	0	0	39,06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	117,19	0	0	117,19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	117,19	0	29,30	0	0	0	0	0	0	0	0	0	39,06	0	0	0	107,42	0	0	0	0	0
10	0	0	0	0	0	0	0	0	29,30	0	29,30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	29,30	0	48,83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39,06
12	0	0	0	0	0	29,30	39,06	0	0	0	48,83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	87,89	0	0	0	0	0	0	0	0	0	0	0	87,89	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	87,89	0	29,30	48,83	48,83	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	29,30	0	29,30	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	48,83	29,30	0	48,83	87,89	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	87,89	0	0	0	0	0	0	0	48,83	0	48,83	0	0	0	29,30	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	87,89	0	0	87,89	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	87,89	0	29,30	78,12	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	39,06	0	0	0	0	0	0	0	29,30	0	29,30	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	78,12	0	0	48,83	29,30	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48,83	0	19,53	0	0	58,59	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	29,30	19,53	0	48,83	0	0	0	0	0
24	0	0	0	0	0	0	0	0	107,42	0	0	0	0	0	0	0	0	0	0	0	0	0	48,83	0	78,12	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	78,12	0	48,83	29,30	0	19,53
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	58,59	0	0	48,83	0	29,30	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	29,30	29,30	0	29,30	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	29,30	0	29,30
29	0	0	0	0	0	0	0	0	0	0	39,06	0	0	0	0	0	0	0	0	0	0	0	0	0	19,53	0	0	29,30	0

TABLE A.9: Capacités avec la topologie US avec la protection GMPLS (Gbits/s).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
1	0	1,68	1,58	0,47	0,28	1,02	0,47	0,19	0,28	0,09	0,28	0,28	0,19	0,37	0,75	0,09	0,37	0,19	0,37	0,56	0,19	0,28	0,47	0,19	0,19	1,21	0,65	0,09	0,37
2	1,68	0	7,54	2,24	1,02	4,75	2,33	0,84	1,12	0,19	1,21	0,93	0,56	1,77	3,63	0,19	2,05	0,84	1,49	2,05	1,30	0,93	1,68	0,93	0,84	5,77	3,07	0,19	1,49
3	1,58	7,54	0	2,05	1,02	4,38	2,14	0,84	1,02	0,19	1,21	0,93	0,56	1,68	3,45	0,19	1,86	0,84	1,40	1,96	0,65	0,93	2,05	0,84	0,75	5,31	2,89	0,19	1,40
4	0,47	2,24	2,05	0	0,37	1,12	0,75	0,19	0,28	0,09	0,37	0,28	0,19	0,47	0,84	0,09	0,47	0,19	0,37	0,56	0,19	0,28	0,56	0,28	0,19	1,40	0,75	0,09	0,37
5	0,28	1,02	1,02	0,37	0	0,65	0,37	0,19	0,19	0,09	0,19	0,19	0,09	0,28	0,56	0,09	0,28	0,19	0,19	0,37	0,09	0,19	0,28	0,19	0,19	0,75	0,47	0,09	0,19
6	1,02	4,75	4,38	1,12	0,65	0	1,40	0,56	0,75	0,19	0,75	0,65	0,37	1,21	2,42	0,19	1,12	0,56	0,93	1,58	0,47	0,65	1,49	0,65	0,56	3,82	2,05	0,19	1,02
7	0,47	2,33	2,14	0,75	0,37	1,40	0	0,28	0,37	0,09	0,37	0,28	0,19	0,56	1,02	0,09	0,47	0,28	0,47	0,75	0,28	0,28	0,65	0,28	0,28	1,68	0,93	0,09	0,47
8	0,19	0,84	0,84	0,19	0,19	0,56	0,28	0	0,19	0,09	0,19	0,09	0,09	0,19	0,37	0,09	0,19	0,09	0,19	0,28	0,09	0,09	0,28	0,19	0,09	0,65	0,37	0,09	0,19
9	0,28	1,12	1,02	0,28	0,19	0,75	0,37	0,19	0	0,09	0,19	0,19	0,09	0,28	0,47	0,09	0,28	0,19	0,19	0,37	0,09	0,19	0,37	0,19	0,19	0,75	0,47	0,09	0,19
10	0,09	0,19	0,19	0,09	0,09	0,19	0,09	0,09	0,09	0	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,19	0,09	0,09	0,09
11	0,28	1,21	1,21	0,37	0,19	0,75	0,37	0,19	0,19	0,09	0	0,19	0,09	0,28	0,47	0,09	0,28	0,19	0,19	0,37	0,09	0,19	0,28	0,19	0,09	0,75	0,37	0,09	0,28
12	0,28	0,93	0,93	0,28	0,19	0,65	0,28	0,09	0,19	0,09	0,19	0	0,09	0,28	0,47	0,09	0,28	0,19	0,19	0,28	0,09	0,19	0,28	0,19	0,09	0,75	0,37	0,09	0,19
13	0,19	0,56	0,56	0,19	0,09	0,37	0,19	0,09	0,09	0,09	0,09	0,09	0	0,19	0,28	0,09	0,19	0,09	0,09	0,19	0,19	0,09	0,09	0,09	0,09	0,37	0,28	0,09	0,09
14	0,37	1,77	1,68	0,47	0,28	1,21	0,56	0,19	0,28	0,09	0,28	0,28	0,19	0	0,75	0,09	0,47	0,19	0,37	0,47	0,56	0,19	0,09	0,28	0,19	1,30	0,65	0,09	0,37
15	0,75	3,63	3,45	0,84	0,56	2,42	1,02	0,37	0,47	0,09	0,47	0,47	0,28	0,75	0	0,19	0,93	0,47	0,75	0,84	1,12	0,47	0,28	0,56	0,37	2,61	1,40	0,09	0,75
16	0,09	0,19	0,19	0,09	0,09	0,19	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,19	0	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,19	0,19	0,09	0,09
17	0,37	2,05	1,86	0,47	0,28	1,12	0,47	0,19	0,28	0,09	0,28	0,28	0,19	0,47	0,93	0,09	0	0,28	0,37	0,47	0,56	0,19	0,19	0,28	0,19	1,21	0,65	0,09	0,37
18	0,19	0,84	0,84	0,19	0,19	0,56	0,28	0,09	0,19	0,09	0,19	0,19	0,09	0,19	0,47	0,09	0,28	0	0,19	0,19	0,28	0,09	0,09	0,19	0,09	0,65	0,37	0,09	0,19
19	0,37	1,49	1,40	0,37	0,19	0,93	0,47	0,19	0,19	0,09	0,19	0,19	0,09	0,37	0,75	0,09	0,37	0,19	0	0,47	0,56	0,19	0,09	0,28	0,19	1,21	0,65	0,09	0,28
20	0,56	2,05	1,96	0,56	0,37	1,58	0,75	0,28	0,37	0,09	0,37	0,28	0,19	0,47	0,84	0,09	0,47	0,19	0,47	0	0,65	0,28	0,19	0,37	0,28	1,58	0,93	0,09	0,47
21	0,19	1,30	0,65	0,19	0,09	0,47	0,28	0,09	0,09	0,09	0,09	0,09	0,19	0,56	1,12	0,09	0,56	0,28	0,56	0,65	0	0,09	0,09	0,09	0,09	0,56	0,28	0,09	0,19
22	0,28	0,93	0,93	0,28	0,19	0,65	0,28	0,09	0,19	0,09	0,19	0,19	0,09	0,19	0,47	0,09	0,19	0,09	0,19	0,28	0,09	0	0,28	0,19	0,09	0,75	0,37	0,09	0,19
23	0,47	1,68	2,05	0,56	0,28	1,49	0,65	0,28	0,37	0,09	0,28	0,28	0,09	0,09	0,28	0,09	0,19	0,09	0,09	0,19	0,09	0,28	0	0,37	0,19	1,68	0,84	0,09	0,47
24	0,19	0,93	0,84	0,28	0,19	0,65	0,28	0,19	0,19	0,09	0,19	0,19	0,09	0,28	0,56	0,09	0,28	0,19	0,28	0,37	0,09	0,19	0,37	0	0,19	0,75	0,47	0,09	0,19
25	0,19	0,84	0,75	0,19	0,19	0,56	0,28	0,09	0,19	0,09	0,09	0,09	0,09	0,19	0,37	0,09	0,19	0,09	0,19	0,28	0,09	0,09	0,19	0,19	0	0,47	0,37	0,09	0,19
26	1,21	5,77	5,31	1,40	0,75	3,82	1,68	0,65	0,75	0,19	0,75	0,75	0,37	1,30	2,61	0,19	1,21	0,65	1,21	1,58	0,56	0,75	1,68	0,75	0,47	0	2,24	0,19	1,21
27	0,65	3,07	2,89	0,75	0,47	2,05	0,93	0,37	0,47	0,09	0,37	0,37	0,28	0,65	1,40	0,19	0,65	0,37	0,65	0,93	0,28	0,37	0,84	0,47	0,37	2,24	0	0,09	0,65
28	0,09	0,19	0,19	0,09	0,09	0,19	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,19	0,09	0	0,09
29	0,37	1,49	1,40	0,37	0,19	1,02	0,47	0,19	0,19	0,09	0,28	0,19	0,09	0,37	0,75	0,09	0,37	0,19	0,28	0,47	0,19	0,19	0,47	0,19	0,19	1,21	0,65	0,09	0

Table A.10: Matrice de trafic avec la topologie US (Gbits/s).

A.3 Topologie européenne

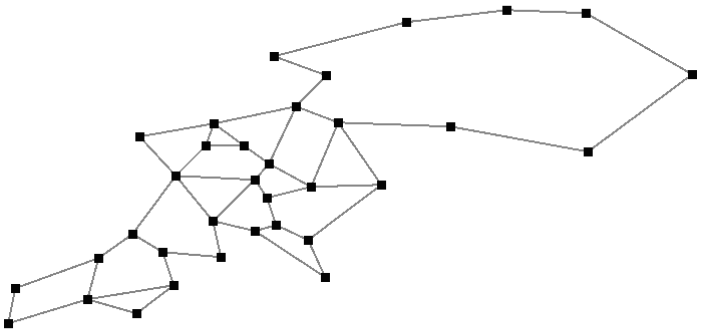


Figure A.3: Topologie européenne.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
1							x		x				x			x																		
2																	x													x	x			
3									x											x											x	x	
4						x											x							x										
5																		x						x							x			
6				x																			x						x					
7	x												x										x											
8									x				x								x							x						
9	x		x					x			x																							
10																										x	x							
11									x													x												
12																				x													x	
13	x						x	x																										
14																		x					x					x	x					
15																	x							x										
16	x																						x											
17		x		x											x															x				
18														x															x					
19					x																							x					x	
20												x														x								
21			x					x																							x		x	
22											x																x							
23						x	x							x		x												x						
24				x											x																			
25					x																								x					
26										x										x														
27										x												x												
28								x					x										x											x
29													x						x						x									
30		x				x												x																
31		x															x																	
32			x		x																x													
33			x									x																						
34																			x		x							x						

TABLE A.11: Topologie européenne.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
1	0	0	0	0	0	0	250	0	910	0	0	0	380	0	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	800	0	0	0	0	0	0	0	0	0	0	0	0	480	390	0	0	0
3	0	0	0	0	0	0	0	0	330	0	0	0	0	0	0	0	0	0	0	0	870	0	0	0	0	0	0	0	0	0	0	710	870	0
4	0	0	0	0	0	390	0	0	0	0	0	0	0	0	0	0	420	0	0	0	0	0	0	820	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200	0	0	0	0	0	440	0	0	0	0	0	0	880	0	0
6	0	0	0	390	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	720	0	0	0	0	0	0	270	0	0	0	0
7	250	0	0	0	0	0	0	0	0	0	0	0	250	0	0	0	0	0	0	0	0	0	390	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	520	0	0	0	280	0	0	0	0	0	0	0	490	0	0	0	0	0	0	260	0	0	0	0	0	0
9	910	0	330	0	0	0	0	520	0	0	500	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	450	570	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	500	0	0	0	0	0	0	0	0	0	0	0	0	810	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1110	0	0	0	0	0	0	0	0	0	0	0	0	960	0
13	380	0	0	0	0	0	250	280	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	390	0	0	0	0	520	0	0	0	0	580	340	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	760	0	0	0	0	0	0	480	0	0	0	0	0	0	0	0	0	0
16	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	630	0	0	0	0	0	0	0	0	0	0	0
17	0	800	0	420	0	0	0	0	0	0	0	0	0	0	760	0	0	0	0	0	0	0	0	0	0	0	0	0	0	400	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	390	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	500	0	0	0	0
19	0	0	0	0	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	180	0	0	0	0	370
20	0	0	0	0	0	0	0	0	0	0	1110	0	0	0	0	0	0	0	0	0	0	0	0	0	0	780	0	0	0	0	0	0	0	0
21	0	0	870	0	0	0	0	490	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	570	0	340	
22	0	0	0	0	0	0	0	0	0	0	810	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	640	0	0	0	0	0	0	0
23	0	0	0	0	0	720	390	0	0	0	0	0	0	520	0	630	0	0	0	0	0	0	0	0	0	0	0	530	0	0	0	0	0	0
24	0	0	0	820	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	440	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	710	0	0	0	0	0
26	0	0	0	0	0	0	0	0	450	0	0	0	0	0	0	0	0	0	0	780	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	570	0	0	0	0	0	0	0	0	0	0	0	0	640	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	260	0	0	0	0	0	580	0	0	0	0	0	0	0	0	530	0	0	0	0	0	0	0	0	0	0	310
29	0	0	0	0	0	0	0	0	0	0	0	0	0	340	0	0	0	0	180	0	0	0	0	0	710	0	0	0	0	0	0	0	0	0
30	0	480	0	0	0	270	0	0	0	0	0	0	0	0	0	0	500	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	390	0	0	0	0	0	0	0	0	0	0	0	0	0	0	400	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	710	0	880	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	570	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	870	0	0	0	0	0	0	0	960	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	370	0	340	0	0	0	0	0	0	310	0	0	0	0	0	0

TABLE A.12: Métriques avec la topologie européenne.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
1	0	0	0	0	0	0	126,95	0	166,02	0	0	0	126,95	0	0	185,55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9,77	0	0	0	0	0	0	0	0	0	0	0	0	146,48	136,72	0	0	0
3	0	0	0	0	0	0	0	0	175,78	0	0	0	0	0	0	0	0	0	0	0	205,08	0	0	0	0	0	0	0	0	0	0	68,36	175,78	0
4	0	0	0	0	0	146,48	0	0	0	0	0	0	0	0	0	0	136,72	0	0	0	0	0	0	29,30	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	58,59	0	0	0	0	0	39,06	0	0	0	0	0	0	58,59	0	0
6	0	0	0	146,48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	185,55	0	0	0	0	0	0	126,95	0	0	0	0
7	126,95	0	0	0	0	0	0	0	0	0	0	0	126,95	0	0	0	0	0	0	0	0	0	146,48	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	185,55	0	0	0	156,25	0	0	0	0	0	0	0	117,19	0	0	0	0	0	0	224,61	0	0	0	0	0	0
9	166,02	0	175,78	0	0	0	0	185,55	0	0	166,02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	107,42	117,19	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	166,02	0	0	0	0	0	0	0	0	0	0	0	0	156,25	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	156,25	0	0	0	0	0	0	0	0	0	0	0	0	166,02	0
13	126,95	0	0	0	0	0	126,95	156,25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	166,02	0	0	0	0	107,42	0	0	0	0	146,48	87,89	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39,06	0	0	0	0	0	0	19,53	0	0	0	0	0	0	0	0	0	0
16	185,55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	185,55	0	0	0	0	0	0	0	0	0	0	0
17	0	9,77	0	136,72	0	0	0	0	0	0	0	0	0	0	39,06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	126,95	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	166,02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	156,25	0	0	0	0
19	0	0	0	0	58,59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	68,36	0	0	0	97,66
20	0	0	0	0	0	0	0	0	0	0	0	156,25	0	0	0	0	0	0	0	0	0	0	0	0	0	107,42	0	0	0	0	0	0	0	0
21	0	0	205,08	0	0	0	0	117,19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	58,59	0	126,95
22	0	0	0	0	0	0	0	0	0	0	156,25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	146,48	0	0	0	0	0	0	0
23	0	0	0	0	0	185,55	146,48	0	0	0	0	0	0	107,42	0	185,55	0	0	0	0	0	0	0	0	0	0	0	175,78	0	0	0	0	0	0
24	0	0	0	29,30	0	0	0	0	0	0	0	0	0	0	19,53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	39,06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39,06	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	107,42	0	0	0	0	0	0	0	0	0	107,42	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	117,19	0	0	0	0	0	0	0	0	0	0	0	146,48	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	224,61	0	0	0	0	0	146,48	0	0	0	0	0	0	0	0	175,78	0	0	0	0	0	0	0	0	0	0	126,95
29	0	0	0	0	0	0	0	0	0	0	0	0	0	87,89	0	0	0	0	68,36	0	0	0	0	0	39,06	0	0	0	0	0	0	0	0	0
30	0	146,48	0	0	0	126,95	0	0	0	0	0	0	0	0	0	0	0	156,25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	136,72	0	0	0	0	0	0	0	0	0	0	0	0	0	0	126,95	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	68,36	0	58,59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	58,59	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	175,78	0	0	0	0	0	0	0	0	166,02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97,66	0	126,95	0	0	0	0	0	0	126,95	0	0	0	0	0	0

TABLE A.13: Capacités avec la topologie européenne (Gbits/s).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
1	0	0	0	0	0	0	97,66	0	107,42	0	0	0	146,48	0	0	205,08	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9,77	0	0	0	0	0	0	0	0	0	0	0	0	156,25	166,02	0	0	0
3	0	0	0	0	0	0	0	0	156,25	0	0	0	0	0	0	0	0	0	0	0	214,84	0	0	0	0	0	0	0	0	0	0	87,89	214,84	0
4	0	0	0	0	0	156,25	0	0	0	0	0	0	0	0	0	0	136,72	0	0	0	0	0	0	39,06	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	87,89	0	0	0	0	0	48,83	0	0	0	0	0	0	68,36	0	0
6	0	0	0	156,25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	185,55	0	0	0	0	0	0	68,36	0	0	0	0
7	97,66	0	0	0	0	0	0	0	0	0	0	0	87,89	0	0	0	0	0	0	0	0	0	146,48	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	185,55	0	0	0	205,08	0	0	0	0	0	0	0	107,42	0	0	0	0	0	0	253,91	0	0	0	0	0	0
9	107,42	0	156,25	0	0	0	0	185,55	0	0	214,84	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	214,84	214,84	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	214,84	0	0	0	0	0	0	0	0	0	0	0	0	214,84	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	214,84	0	0	0	0	0	0	0	0	0	0	0	0	214,84	0
13	146,48	0	0	0	0	0	87,89	205,08	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	185,55	0	0	0	0	87,89	0	0	0	0	107,42	156,25	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39,06	0	0	0	0	0	0	39,06	0	0	0	0	0	0	0	0	0	0
16	205,08	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	205,08	0	0	0	0	0	0	0	0	0	0	0
17	0	9,77	0	136,72	0	0	0	0	0	0	0	0	0	0	39,06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	166,02	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	185,55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	185,55	0	0	0	0
19	0	0	0	0	87,89	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	156,25	0	0	0	0	185,55
20	0	0	0	0	0	0	0	0	0	0	0	214,84	0	0	0	0	0	0	0	0	0	0	0	0	0	214,84	0	0	0	0	0	0	0	0
21	0	0	214,84	0	0	0	0	107,42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97,66	0	214,84
22	0	0	0	0	0	0	0	0	0	0	214,84	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	214,84	0	0	0	0	0	0	0
23	0	0	0	0	0	185,55	146,48	0	0	0	0	0	0	87,89	0	205,08	0	0	0	0	0	0	0	0	0	0	0	234,38	0	0	0	0	0	0
24	0	0	0	39,06	0	0	0	0	0	0	0	0	0	0	39,06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	48,83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48,83	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	214,84	0	0	0	0	0	0	0	0	0	214,84	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	214,84	0	0	0	0	0	0	0	0	0	0	0	214,84	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	253,91	0	0	0	0	0	107,42	0	0	0	0	0	0	0	0	234,38	0	0	0	0	0	0	0	0	0	0	195,31
29	0	0	0	0	0	0	0	0	0	0	0	0	0	156,25	0	0	0	0	156,25	0	0	0	0	0	48,83	0	0	0	0	0	0	0	0	0
30	0	156,25	0	0	0	68,36	0	0	0	0	0	0	0	0	0	0	0	185,55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	166,02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	166,02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	87,89	0	68,36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97,66	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	214,84	0	0	0	0	0	0	0	0	214,84	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	185,55	0	214,84	0	0	0	0	0	0	195,31	0	0	0	0	0	0

TABLE A.14: Capacités avec la topologie européenne avec la protection GMPLS (Gbits/s).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
1	0	0,88	7,81	0,41	0,42	0,37	1,67	2,62	2,34	0,49	1,17	0,53	1,61	0,46	0,49	14,96	2,95	0,58	1,52	2,53	0,75	0,48	5,45	0,38	1,24	0,71	1,14	0,96	0,61	0,44	0,58	1,98	0,44	0,41
2	0,88	0	1,88	0,37	0,19	0,34	0,66	0,85	0,31	0,20	0,20	0,21	0,24	0,19	0,41	3,32	4,49	0,45	1,08	1,59	0,27	0,20	1,41	0,18	0,39	0,26	0,89	0,18	0,23	0,38	0,45	1,31	0,19	0,18
3	7,81	1,88	0	1,42	3,56	1,37	2,67	7,63	4,84	1,49	3,68	3,83	1,65	1,46	1,49	7,99	4,95	1,58	3,53	4,53	4,38	1,48	4,18	1,38	2,24	1,72	3,14	1,39	1,61	1,44	1,58	9,96	3,60	3,53
4	0,41	0,37	1,42	0	0,07	0,11	0,19	0,38	0,20	0,09	0,08	0,10	0,12	0,08	0,17	2,85	3,32	0,11	0,61	1,12	0,15	0,08	2,36	0,12	0,27	0,14	0,42	0,06	0,12	0,15	0,22	0,84	0,07	0,07
5	0,42	0,19	3,56	0,07	0	0,06	0,19	0,38	0,20	0,09	0,08	0,10	0,13	0,08	0,09	2,85	1,33	0,11	1,56	1,13	0,31	0,08	0,95	0,06	0,55	0,14	0,43	0,06	0,23	0,08	0,11	2,13	0,08	0,14
6	0,37	0,34	1,37	0,11	0,06	0	0,35	0,33	0,19	0,07	0,07	0,08	0,11	0,13	0,07	7,00	3,20	0,19	0,57	1,07	0,14	0,07	2,24	0,09	0,26	0,13	0,38	0,10	0,10	0,12	0,10	0,80	0,06	0,05
7	1,67	0,66	2,67	0,19	0,19	0,35	0	1,48	1,77	0,26	0,24	0,31	1,05	0,58	0,26	13,83	2,49	0,35	1,07	2,08	0,53	0,25	4,32	0,15	1,01	0,49	0,69	0,39	0,38	0,21	0,35	1,53	0,21	0,18
8	2,62	0,85	7,63	0,38	0,38	0,33	1,48	0	2,25	0,45	1,08	0,49	1,52	1,05	0,45	5,91	2,87	0,54	1,45	2,45	1,79	0,44	5,27	0,34	1,20	0,68	1,07	0,87	0,57	0,40	0,54	4,77	0,40	0,93
9	2,34	0,31	4,84	0,20	0,20	0,19	1,77	2,25	0	0,22	0,42	0,23	0,51	0,21	0,22	8,42	1,85	0,24	1,14	1,64	0,56	0,43	1,47	0,19	0,40	0,27	0,95	0,38	0,25	0,20	0,24	3,42	0,41	0,20
10	0,49	0,20	1,49	0,09	0,09	0,07	0,26	0,45	0,22	0	0,10	0,12	0,14	0,10	0,11	2,92	1,41	0,13	0,69	2,99	0,17	0,20	1,02	0,08	0,29	0,32	1,25	0,08	0,13	0,09	0,13	0,92	0,09	0,09
11	1,17	0,20	3,68	0,08	0,08	0,07	0,24	1,08	0,42	0,10	0	0,11	0,14	0,09	0,10	2,90	1,38	0,12	0,67	1,17	0,17	0,19	1,00	0,07	0,29	0,16	1,20	0,07	0,13	0,09	0,12	0,90	0,09	0,08
12	0,53	0,21	3,83	0,10	0,10	0,08	0,31	0,49	0,23	0,12	0,11	0	0,15	0,11	0,12	2,96	1,45	0,14	0,73	3,09	0,18	0,11	1,06	0,09	0,30	0,34	0,54	0,09	0,14	0,10	0,14	0,96	0,21	0,10
13	1,61	0,24	1,65	0,12	0,13	0,11	1,05	1,52	0,51	0,14	0,14	0,15	0	0,27	0,14	7,70	1,56	0,17	0,85	1,35	0,42	0,14	2,94	0,12	0,33	0,20	0,65	0,23	0,17	0,13	0,17	1,08	0,13	0,25
14	0,46	0,19	1,46	0,08	0,08	0,13	0,58	1,05	0,21	0,10	0,09	0,11	0,27	0	0,10	7,23	1,37	0,24	1,65	1,16	0,16	0,09	2,47	0,07	0,57	0,15	0,47	0,14	0,25	0,17	0,12	0,89	0,08	0,15
15	0,49	0,41	1,49	0,17	0,09	0,07	0,26	0,45	0,22	0,11	0,10	0,12	0,14	0,10	0	2,92	3,51	0,13	0,69	1,19	0,17	0,10	1,02	0,16	0,29	0,16	0,50	0,08	0,13	0,09	0,26	0,92	0,09	0,09
16	14,96	3,32	7,99	2,85	2,85	7,00	13,83	5,91	8,42	2,92	2,90	2,96	7,70	7,23	2,92	0	7,81	3,01	6,39	7,39	3,18	2,91	17,62	2,81	3,67	3,15	6,00	7,04	3,04	2,87	3,01	6,85	2,87	2,84
17	2,95	4,49	4,95	3,32	1,33	3,20	2,49	2,87	1,85	1,41	1,38	1,45	1,56	1,37	3,51	7,81	0	1,49	3,35	4,36	1,67	1,39	4,01	3,23	2,16	1,63	2,97	1,30	1,52	3,39	3,74	3,81	1,35	1,32
18	0,58	0,45	1,58	0,11	0,11	0,19	0,35	0,54	0,24	0,13	0,12	0,14	0,17	0,24	0,13	3,01	1,49	0	0,78	1,29	0,19	0,12	2,78	0,10	0,32	0,18	0,59	0,20	0,31	0,23	0,15	1,01	0,12	0,11
19	1,52	1,08	3,53	0,61	1,56	0,57	1,07	1,45	1,14	0,69	0,67	0,73	0,85	1,65	0,69	6,39	3,35	0,78	0	2,93	2,38	0,68	2,58	0,58	3,61	0,92	1,54	1,46	2,02	0,64	0,78	5,96	0,64	1,53
20	2,53	1,59	4,53	1,12	1,13	1,07	2,08	2,45	1,64	2,99	1,17	3,09	1,35	1,16	1,19	7,39	4,36	1,29	2,93	0	1,46	1,18	3,59	1,08	1,95	3,55	2,55	1,09	1,31	1,15	1,29	3,39	2,86	1,11
21	0,75	0,27	4,38	0,15	0,31	0,14	0,53	1,79	0,56	0,17	0,17	0,18	0,42	0,16	0,17	3,18	1,67	0,19	2,38	1,46	0	0,17	1,28	0,14	0,36	0,23	0,76	0,29	0,20	0,16	0,19	2,96	0,32	0,30
22	0,48	0,20	1,48	0,08	0,08	0,07	0,25	0,44	0,43	0,20	0,19	0,11	0,14	0,09	0,10	2,91	1,39	0,12	0,68	1,18	0,17	0	1,01	0,07	0,29	0,16	1,22	0,08	0,13	0,09	0,12	0,91	0,09	0,08
23	5,45	1,41	4,18	2,36	0,95	2,24	4,32	5,27	1,47	1,02	1,00	1,06	2,94	2,47	1,02	17,62	4,01	2,78	2,58	3,59	1,28	1,01	0	0,91	1,77	1,25	2,20	2,28	2,85	2,43	1,11	3,04	0,97	2,35
24	0,38	0,18	1,38	0,12	0,06	0,09	0,15	0,34	0,19	0,08	0,07	0,09	0,12	0,07	0,16	2,81	3,23	0,10	0,58	1,08	0,14	0,07	0,91	0	0,27	0,13	0,39	0,05	0,11	0,07	0,10	0,81	0,06	0,06
25	1,24	0,39	2,24	0,27	0,55	0,26	1,01	1,20	0,40	0,29	0,29	0,30	0,33	0,57	0,29	3,67	2,16	0,32	3,61	1,95	0,36	0,29	1,77	0,27	0	0,35	1,25	0,27	0,64	0,28	0,32	4,18	0,28	0,27
26	0,71	0,26	1,72	0,14	0,14	0,13	0,49	0,68	0,27	0,32	0,16	0,34	0,20	0,15	0,16	3,15	1,63	0,18	0,92	3,55	0,23	0,16	1,25	0,13	0,35	0	1,81	0,13	0,19	0,15	0,18	1,15	0,15	0,14
27	1,14	0,89	3,14	0,42	0,43	0,38	0,69	1,07	0,95	1,25	1,20	0,54	0,65	0,47	0,50	6,00	2,97	0,59	1,54	2,55	0,76	1,22	2,20	0,39	1,25	1,81	0	0,39	0,62	0,45	0,59	2,00	0,45	0,42
28	0,96	0,18	1,39	0,06	0,06	0,10	0,39	0,87	0,38	0,08	0,07	0,09	0,23	0,14	0,08	7,04	1,30	0,20	1,46	1,09	0,29	0,08	2,28	0,05	0,27	0,13	0,39	0	0,22	0,07	0,10	0,81	0,07	0,12
29	0,61	0,23	1,61	0,12	0,23	0,10	0,38	0,57	0,25	0,13	0,13	0,14	0,17	0,25	0,13	3,04	1,52	0,31	2,02	1,31	0,20	0,13	2,85	0,11	0,64	0,19	0,62	0,22	0	0,12	0,16	1,04	0,12	0,23
30	0,44	0,38	1,44	0,15	0,08	0,12	0,21	0,40	0,20	0,09	0,09	0,10	0,13	0,17	0,09	2,87	3,39	0,23	0,64	1,15	0,16	0,09	2,43	0,07	0,28	0,15	0,45	0,07	0,12	0	0,23	0,87	0,08	0,07
31	0,58	0,45	1,58	0,22	0,11	0,10	0,35	0,54	0,24	0,13	0,12	0,14	0,17	0,12	0,26	3,01	3,74	0,15	0,78	1,29	0,19	0,12	1,11	0,10	0,32	0,18	0,59	0,10	0,16	0,23	0	1,01	0,12	0,11
32	1,98	1,31	9,96	0,84	2,13	0,80	1,53	4,77	3,42	0,92	0,90	0,96	1,08	0,89	0,92	6,85	3,81	1,01	5,96	3,39	2,96	0,91	3,04	0,81	4,18	1,15	2,00	0,81	1,04	0,87	1,01	0	2,18	2,10
33	0,44	0,19	3,60	0,07	0,08	0,06	0,21	0,40	0,41	0,09	0,09	0,21	0,13	0,08	0,09	2,87	1,35	0,12	0,64	2,86	0,32	0,09	0,97	0,06	0,28	0,15	0,45	0,07	0,12	0,08	0,12	2,18	0	0,07
34	0,41	0,18	3,53	0,07	0,14	0,05	0,18	0,93	0,20	0,09	0,08	0,10	0,25	0,15	0,09	2,84	1,32	0,11	1,53	1,11	0,30	0,08	2,35	0,06	0,27	0,14	0,42	0,12	0,23	0,07	0,11	2,10	0,07	0

Table A.15: Matrice de trafic avec la topologie européenne (Gbits/s).

Annexe B

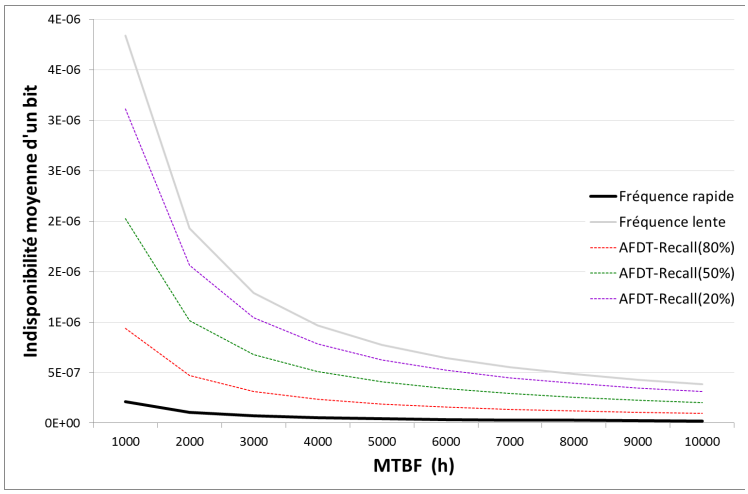
Résultats analytiques

Sommaire

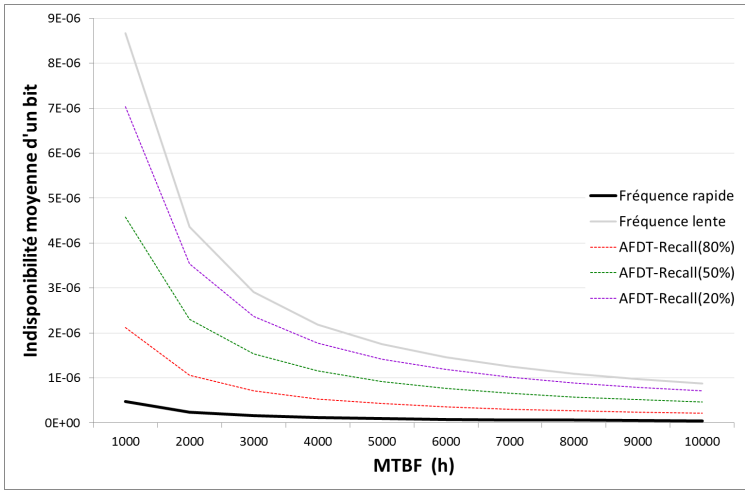
B.1 Détection autonome des pannes (AFDT)

B.1.1 Étude de la disponibilité

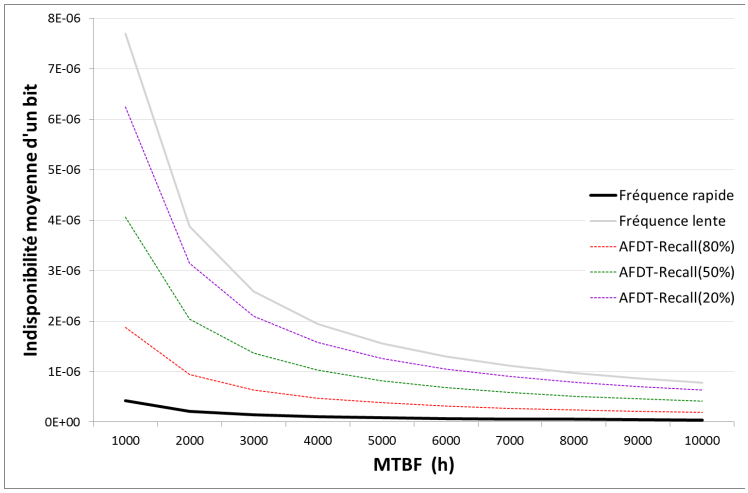
B.1.1.1 Influence de la probabilité de panne



(a) A

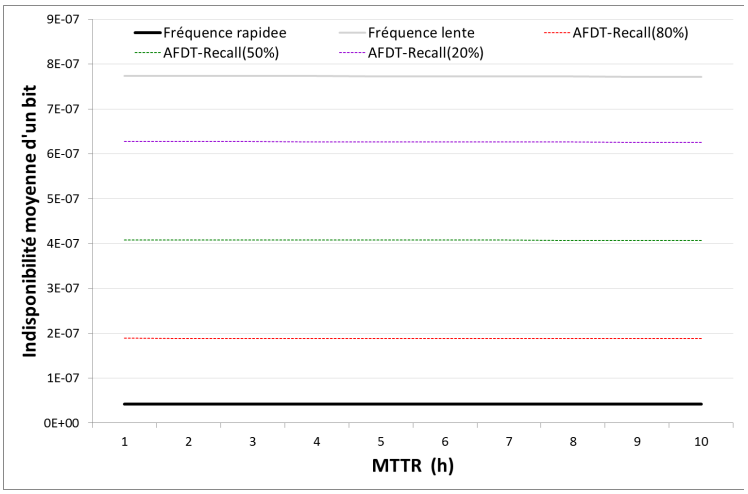


(b) US

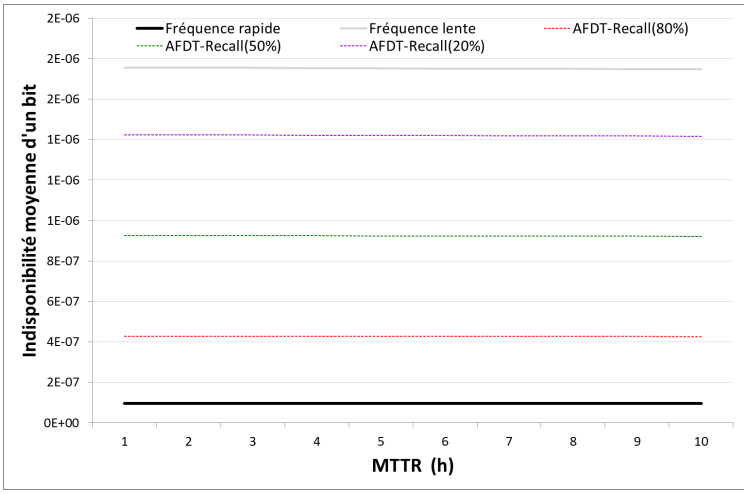


(c) EU

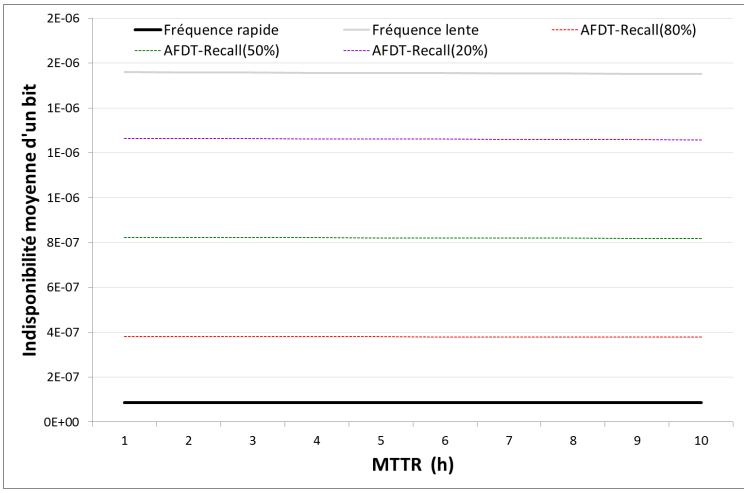
FIGURE B.1: Impact du MTBF sur la disponibilité.



(a) A



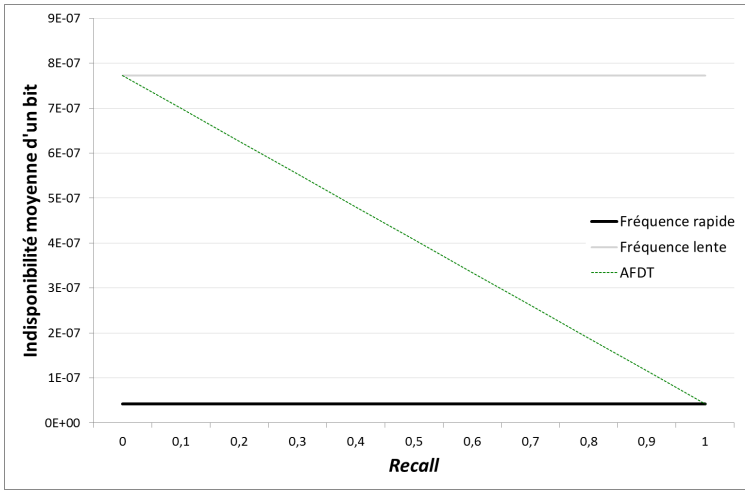
(b) US



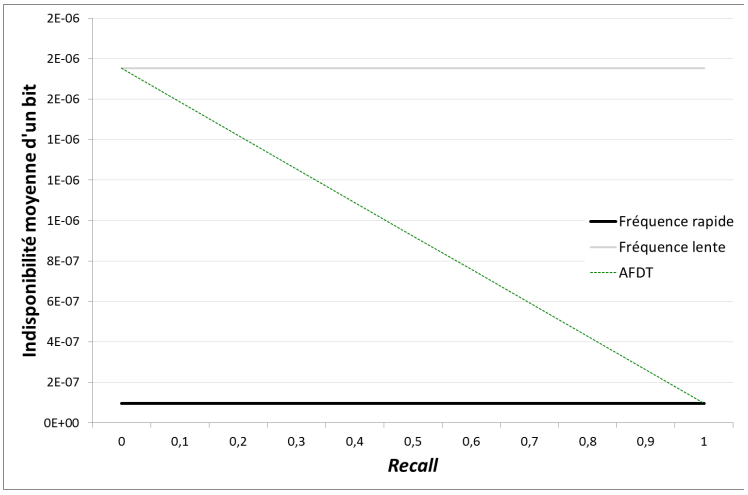
(c) EU

FIGURE B.2: Impact du MTTR sur la disponibilité.

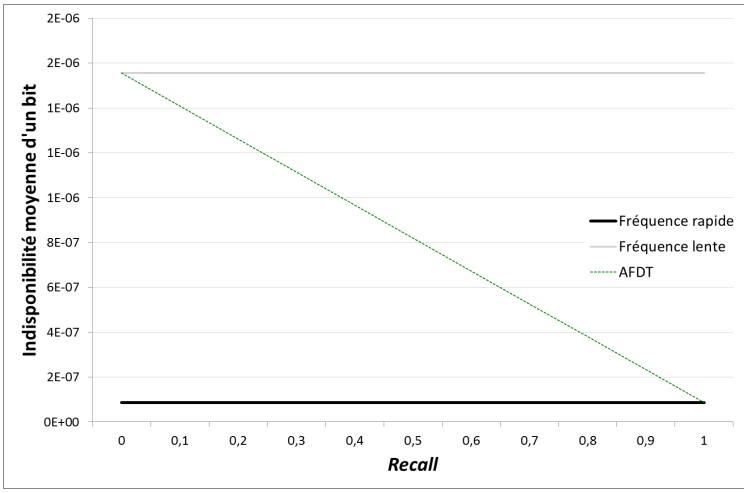
B.1.1.2 Les conséquences de la prédiction de pannes



(a) A



(b) US

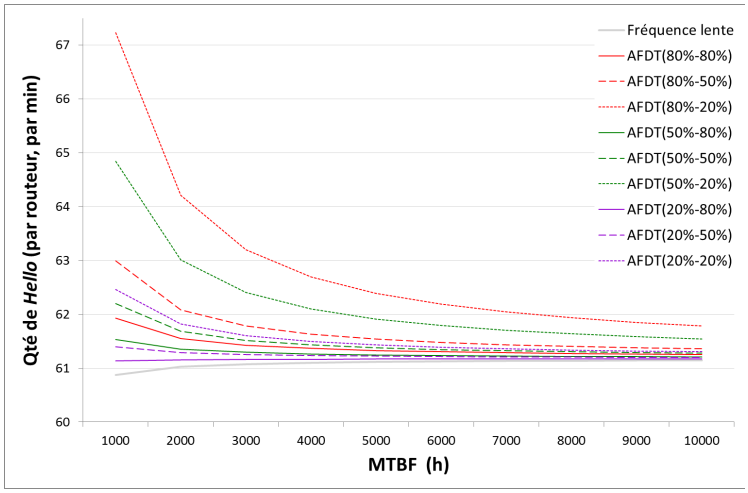


(c) EU

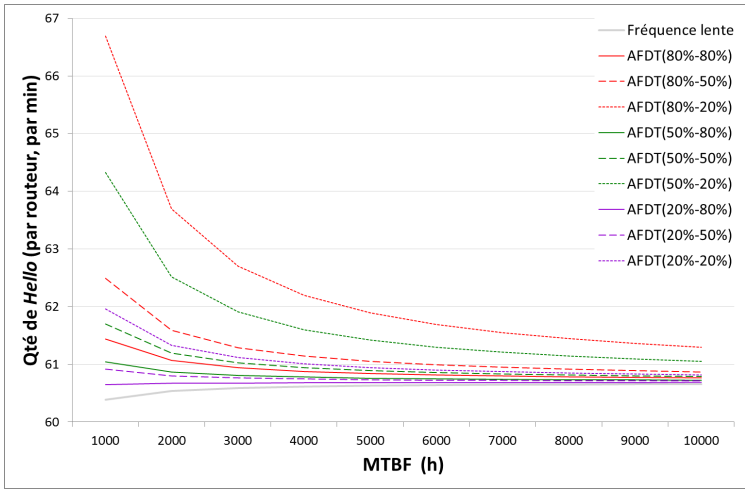
Figure B.3: Impact du *Recall* sur la disponibilité.

B.1.2 Étude de la quantité de message de contrôle à traiter

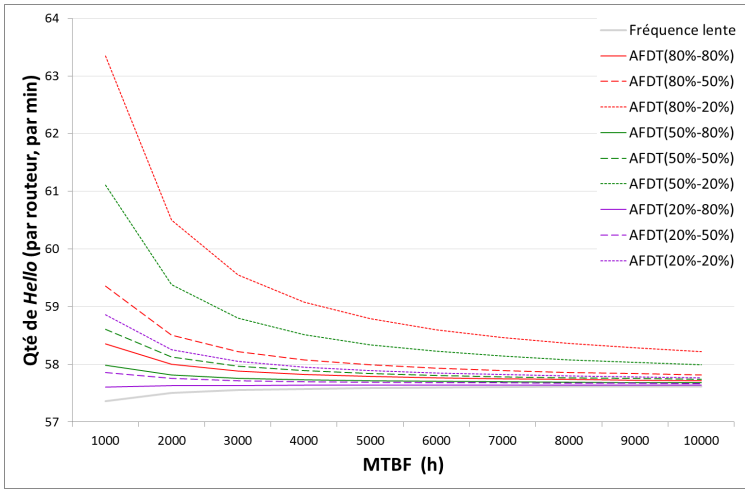
B.1.2.1 Influence de la probabilité de panne



(a) A

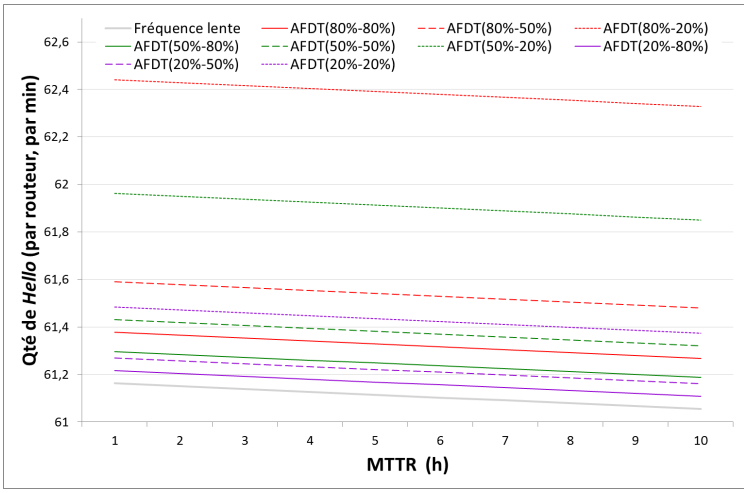


(b) US

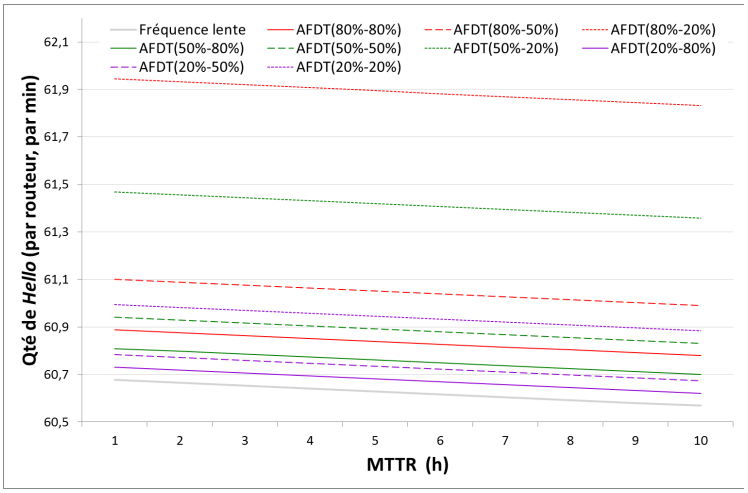


(c) EU

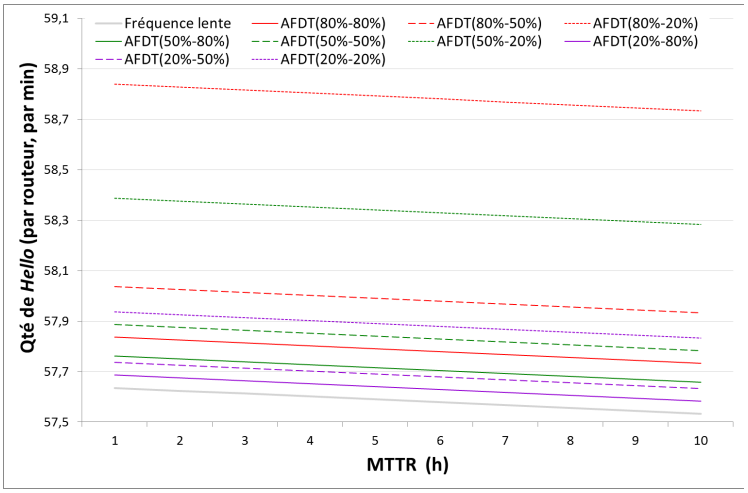
FIGURE B.4: Impact du MTBF sur le nombre de messages *Hello*.



(a) A



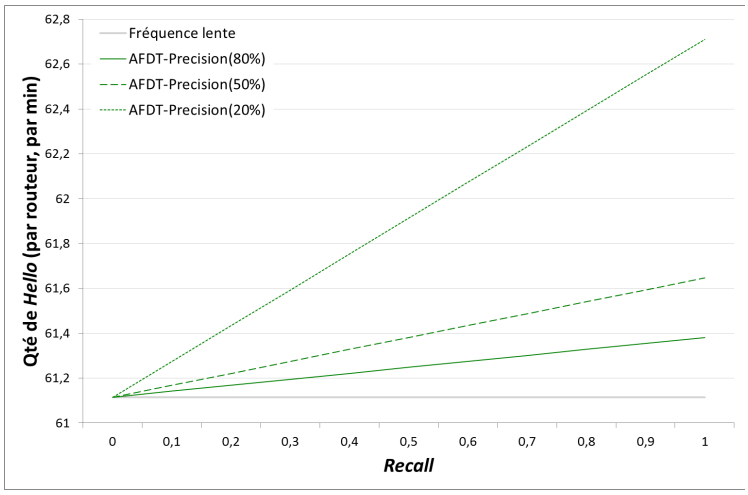
(b) US



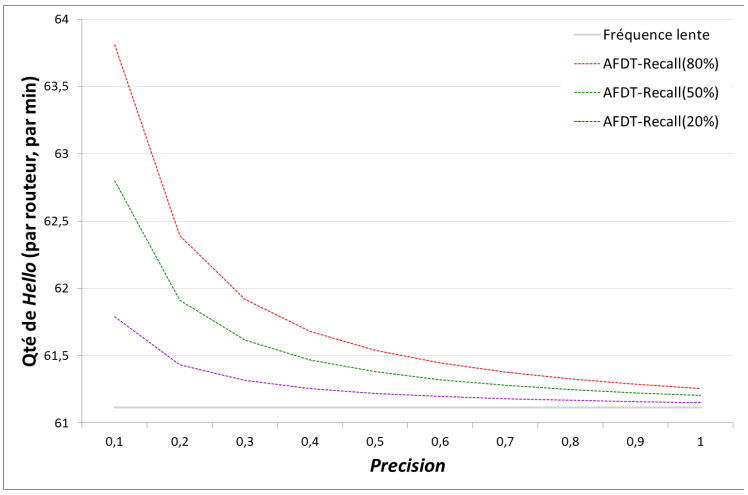
(c) EU

FIGURE B.5: Impact du MTTR sur le nombre de messages *Hello*.

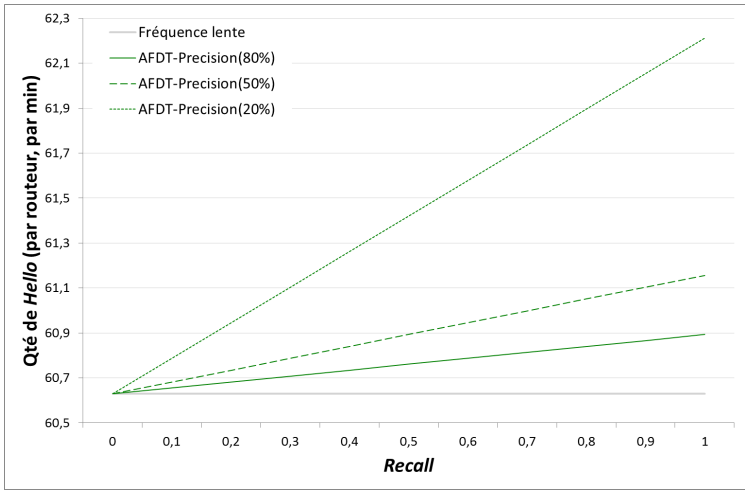
B.1.2.2 Les conséquences de la prédiction de pannes



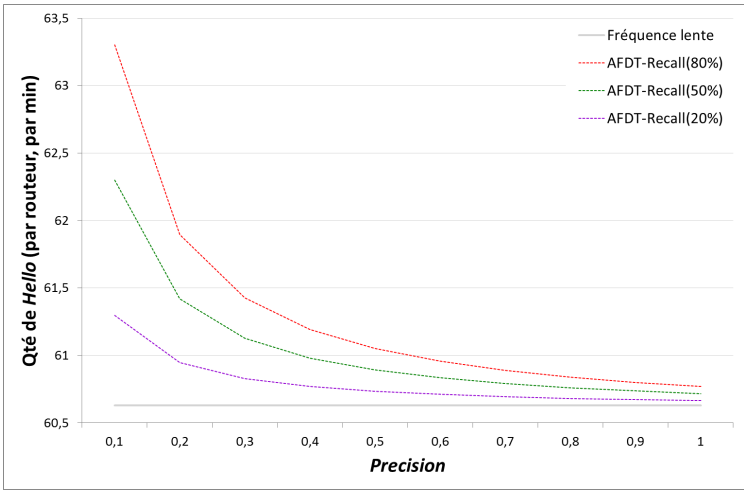
(a) A



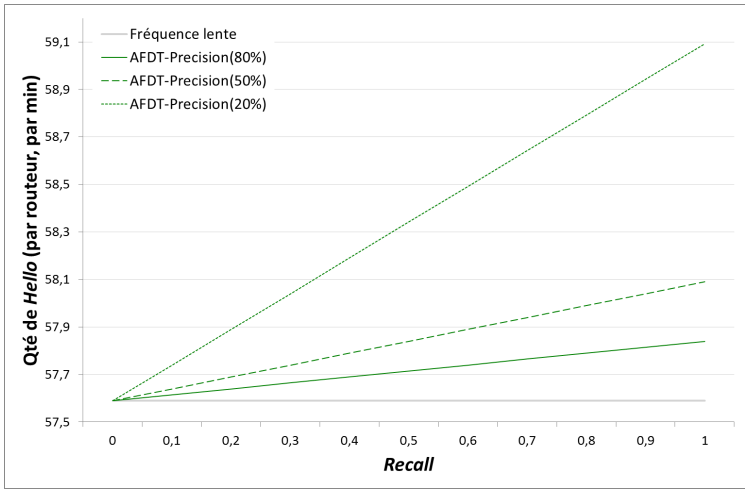
(a) A



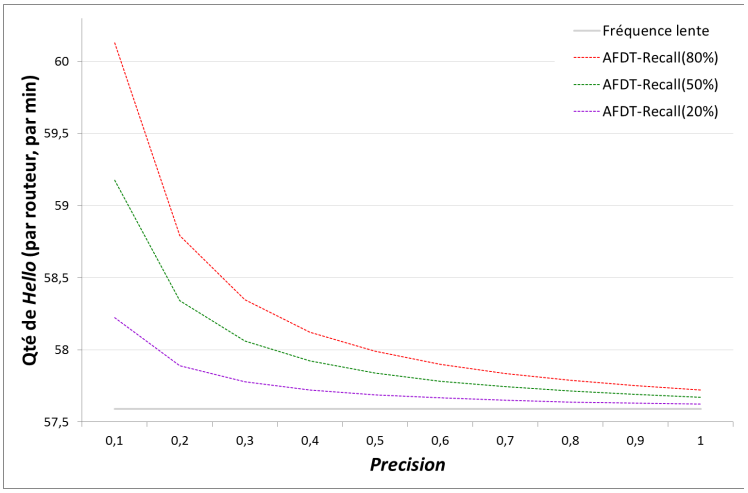
(b) US



(b) US



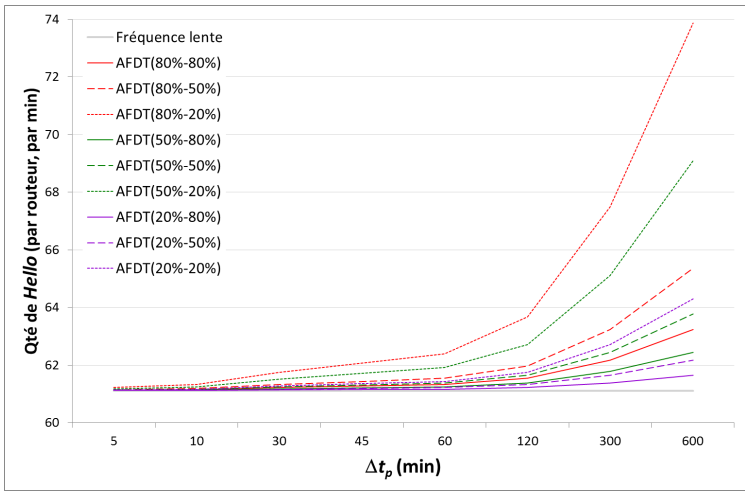
(c) EU



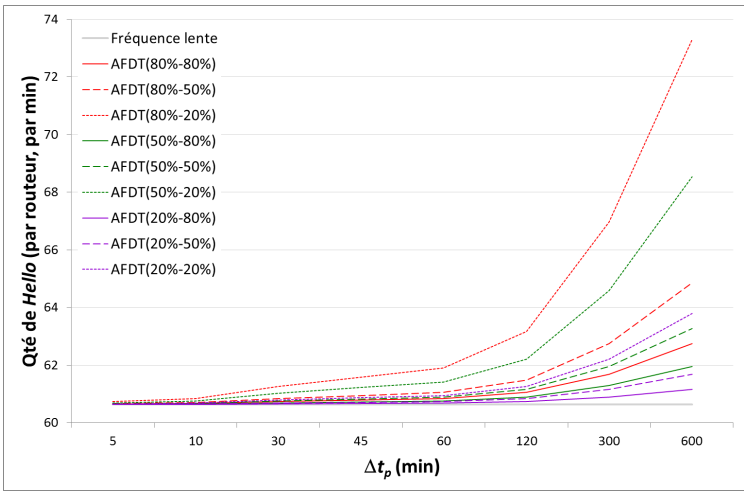
(c) EU

Figure B.6: Impact du *Recall* sur le nombre de messages *Hello*.

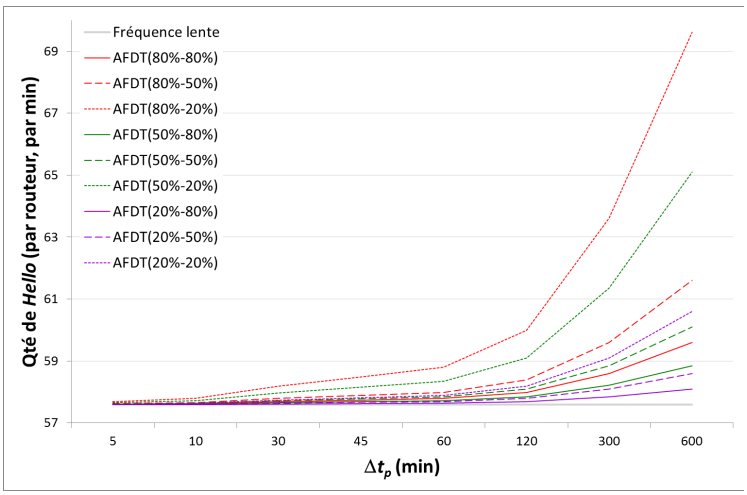
Figure B.7: Impact de la Precision sur le nombre de messages *Hello*.



(a) A



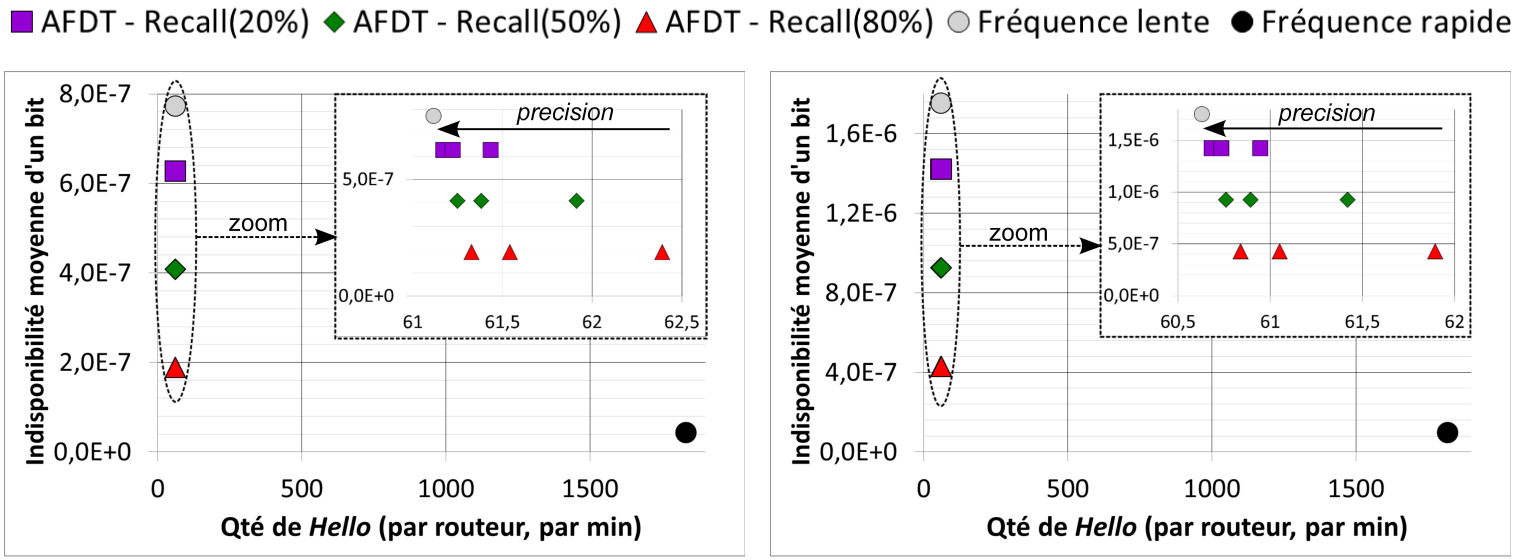
(b) US



(c) EU

FIGURE B.8: Impact de Δt_p sur le nombre de messages *Hello*.

B.1.3 Étude conjointe de la disponibilité et de la quantité de message de contrôle à traiter



(a) A

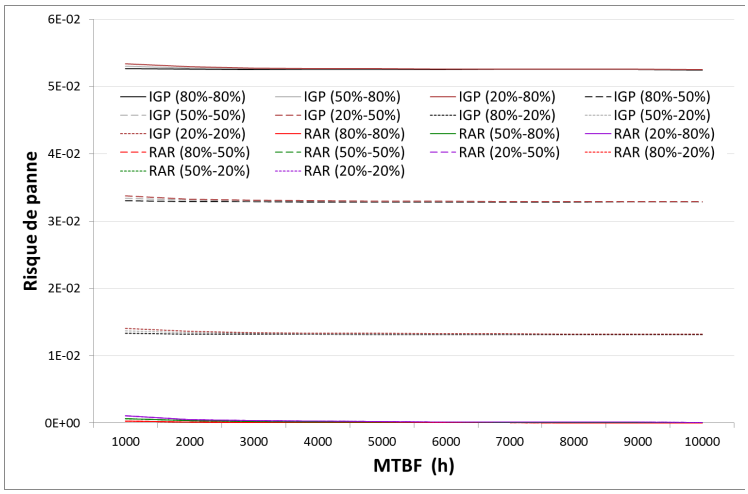
(b) US

(c) EU

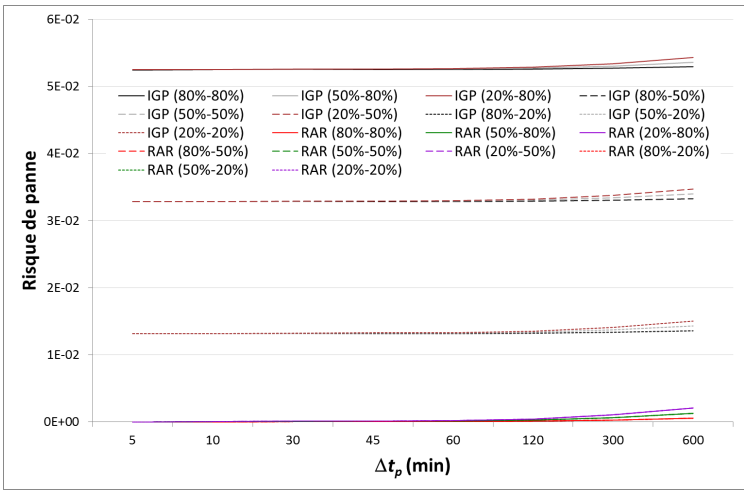
Figure B.9: Ratio indisponibilité / nombre de message *Hello*.

B.2 Routage sensible au risque de pannes (RAR)

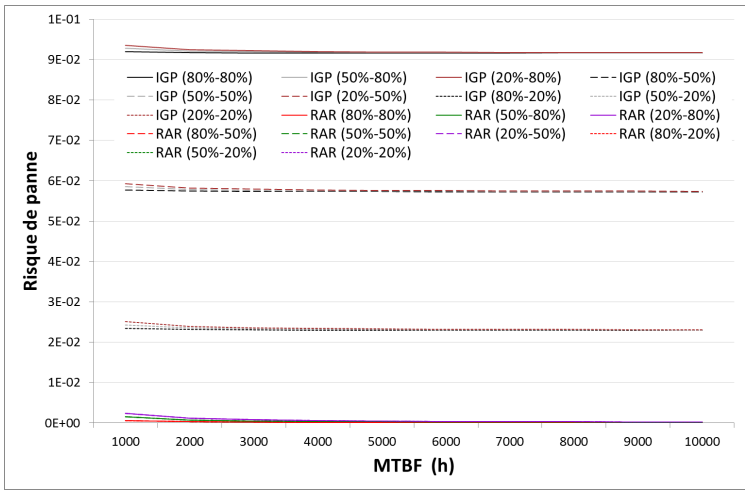
B.2.1 Étude du risque de panne



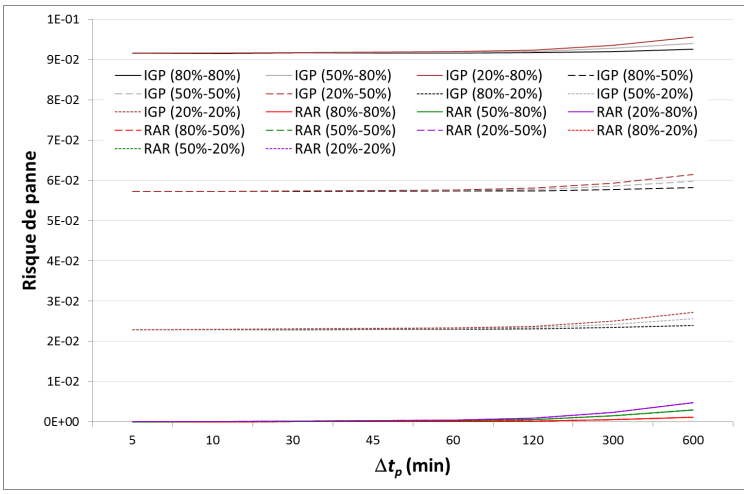
(a) A



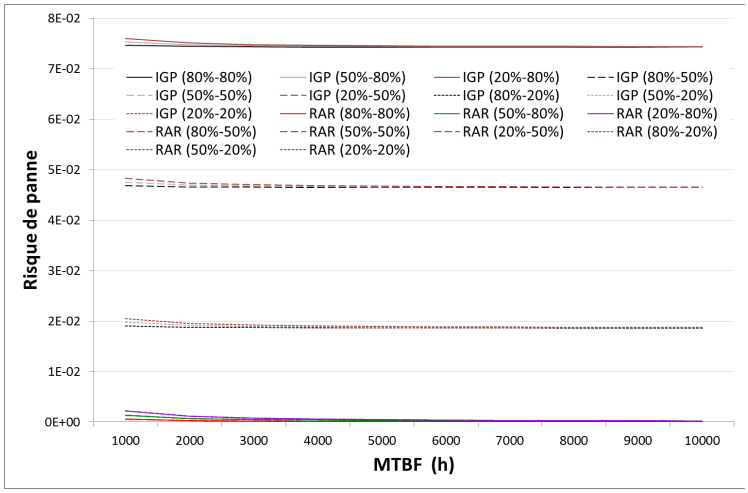
(a) A



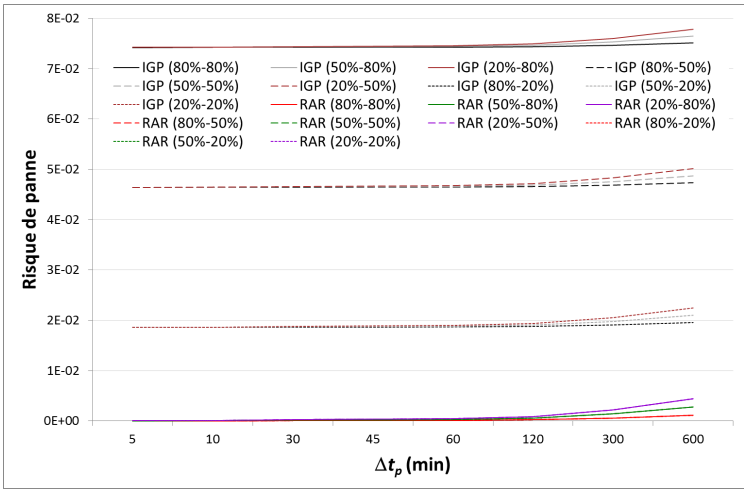
(b) US



(b) US



(c) EU



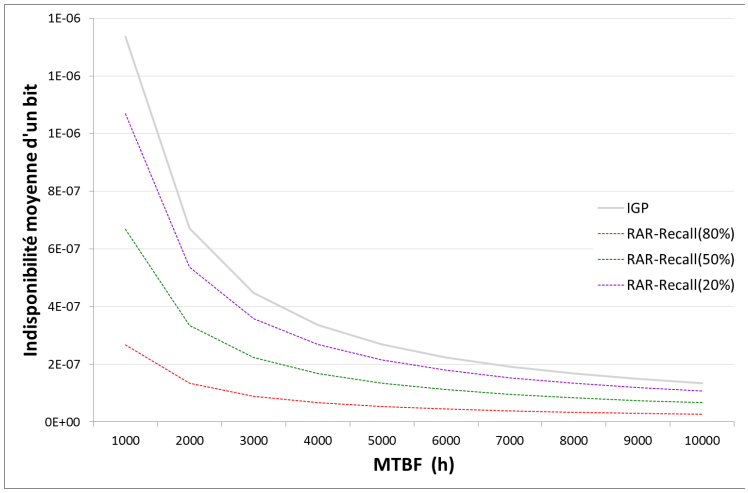
(c) EU

FIGURE B.10: Impact du MTBF sur le risque de panne.

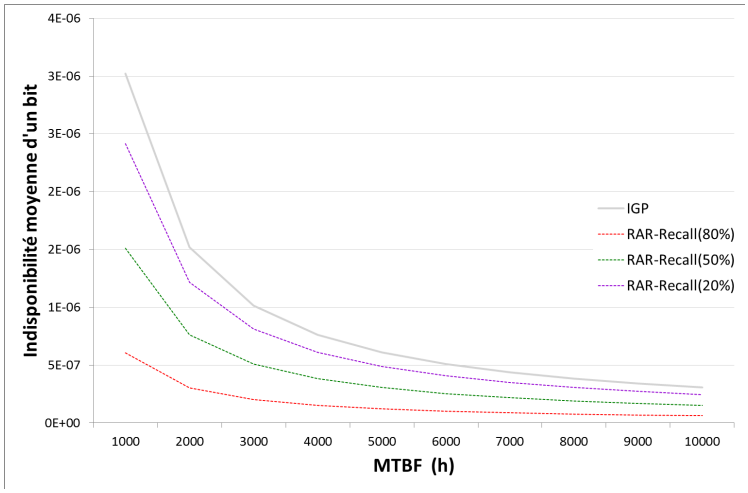
FIGURE B.11: Impact de Δt_p sur le risque de panne.

B.2.2 Étude de la disponibilité

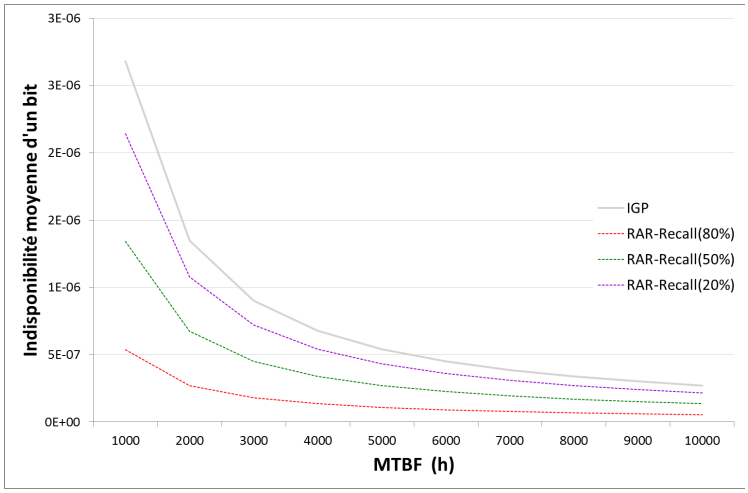
B.2.2.1 Influence de la probabilité de panne



(a) A

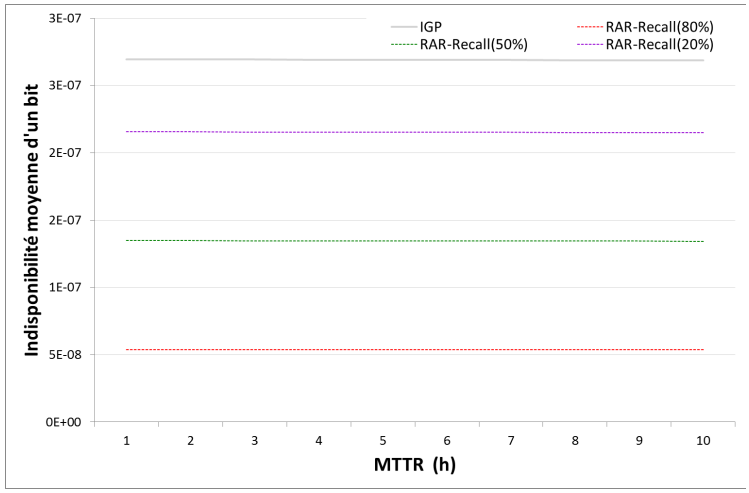


(b) US

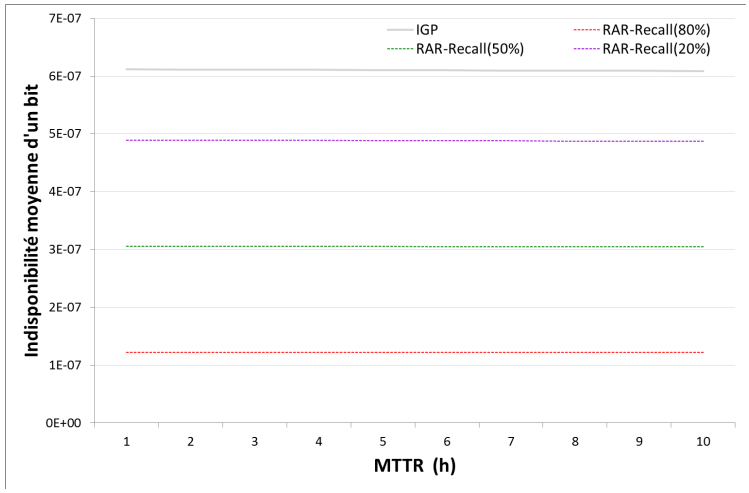


(c) EU

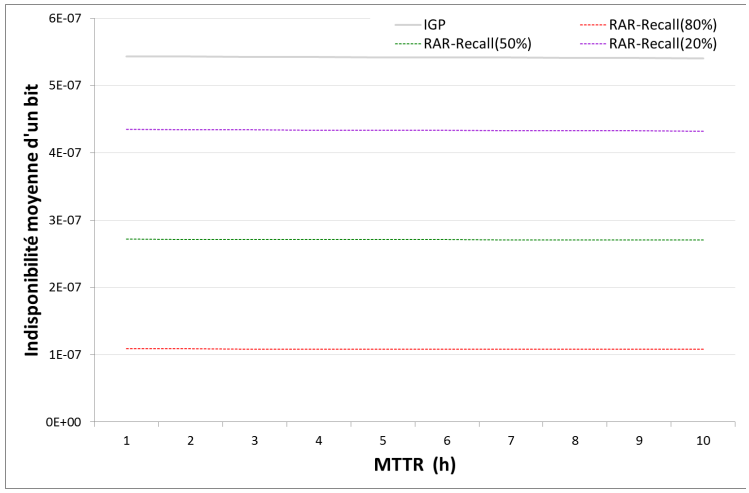
FIGURE B.12: Impact du MTBF sur la disponibilité.



(a) A



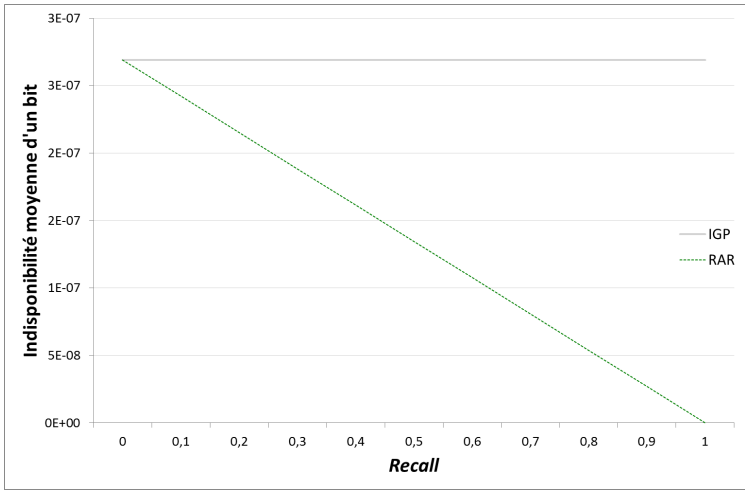
(b) US



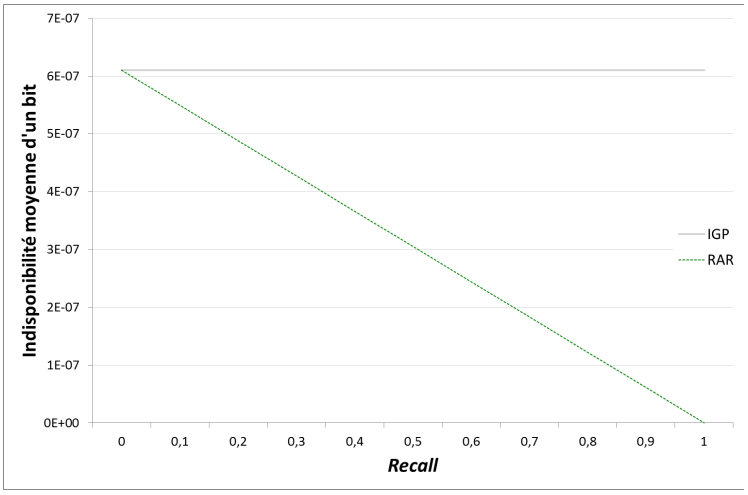
(c) EU

FIGURE B.13: Impact du MTTR sur la disponibilité.

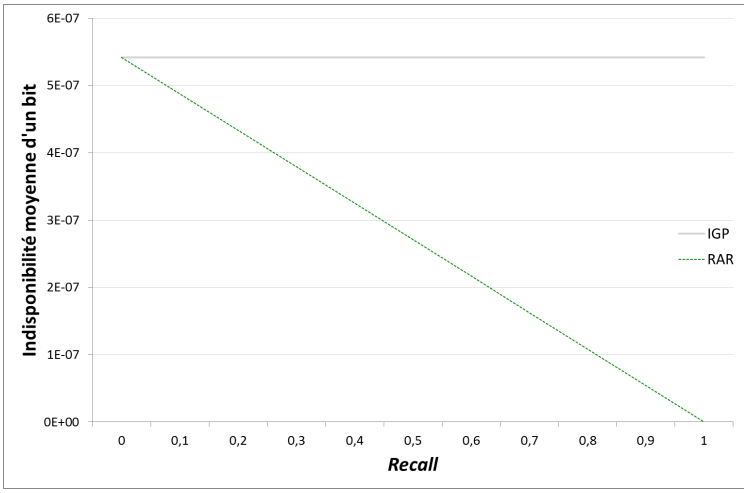
B.2.2.2 Les conséquences de la prédiction de pannes



(a) A



(b) US

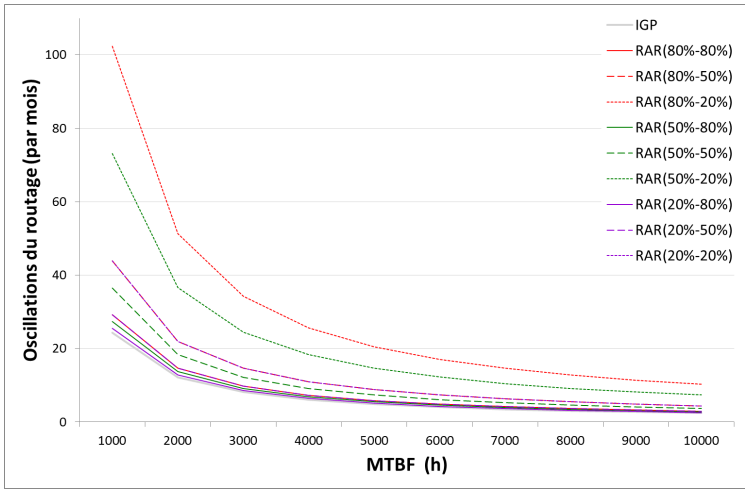


(c) EU

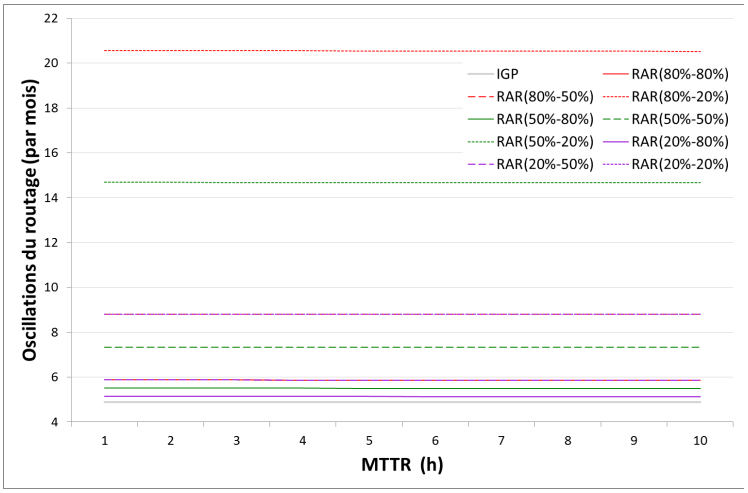
FIGURE B.14: Impact du *Recall* sur la disponibilité.

B.2.3 Étude de la stabilité du routage

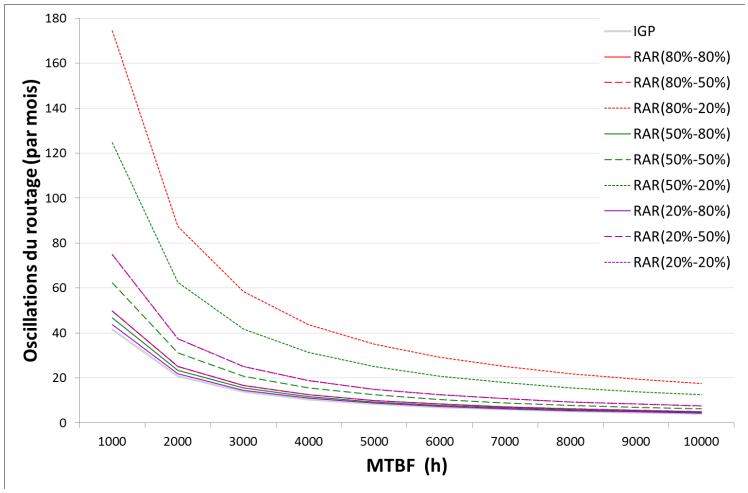
B.2.3.1 Influence de la probabilité de panne



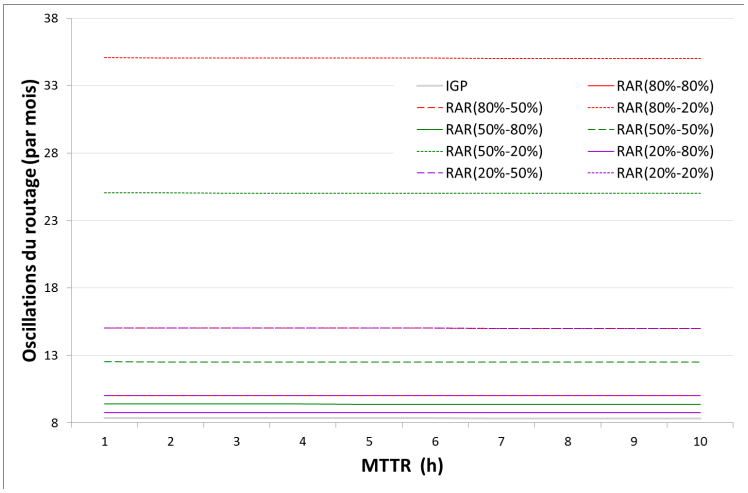
(a) A



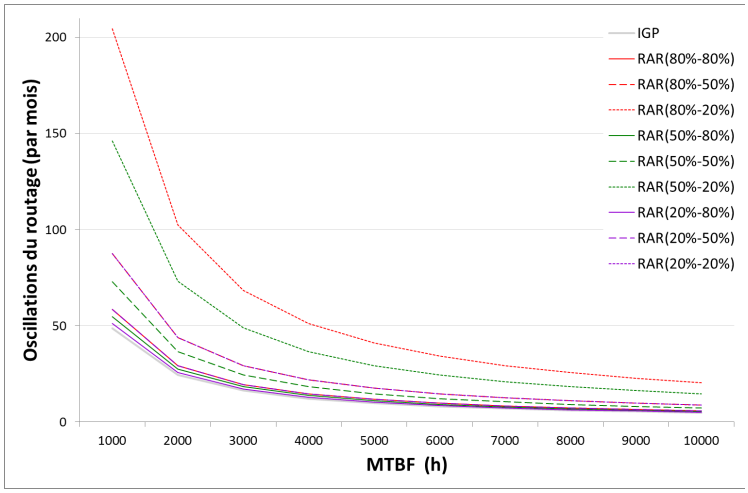
(a) A



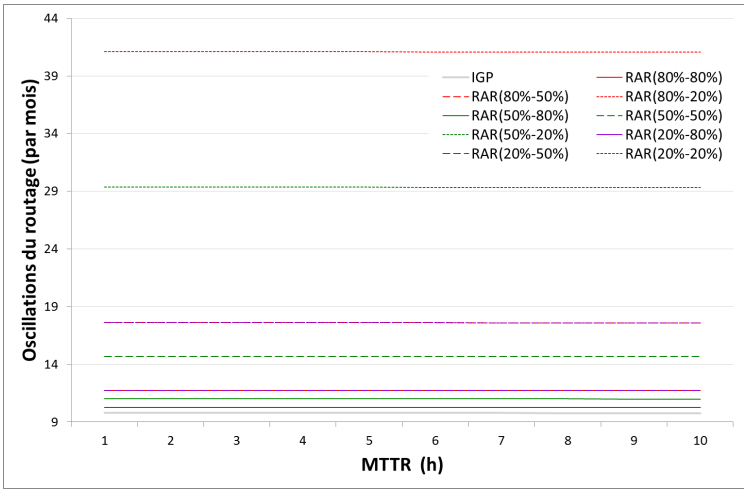
(b) US



(b) US



(c) EU

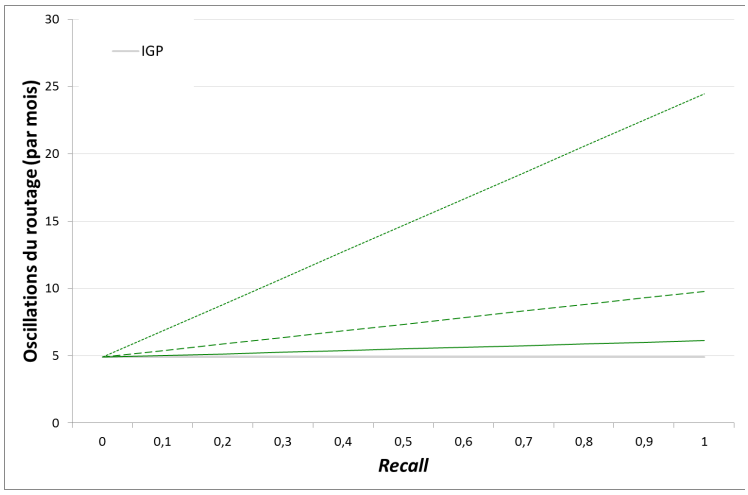


(c) EU

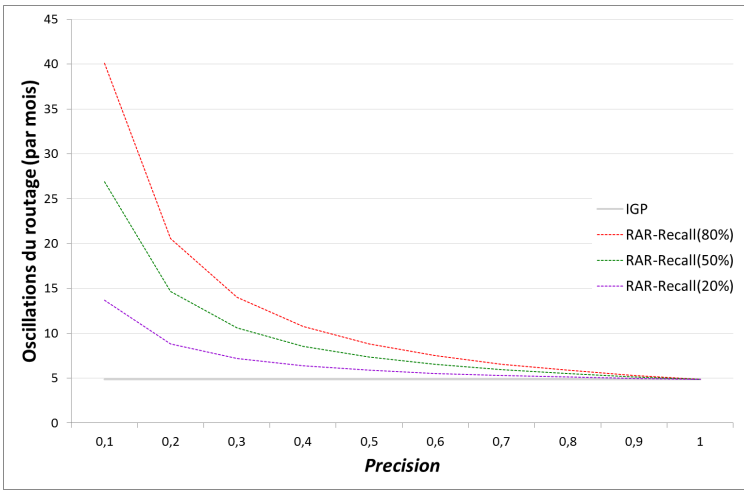
Figure B.15: Impact du MTBF sur le nombre d'oscillations du routage.

Figure B.16: Impact du MTTR sur le nombre d'oscillations du routage.

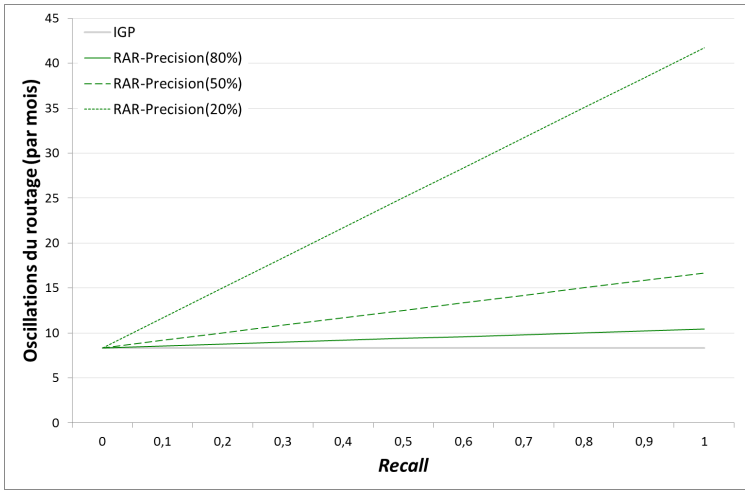
B.2.3.2 Les conséquences de la prédiction de pannes



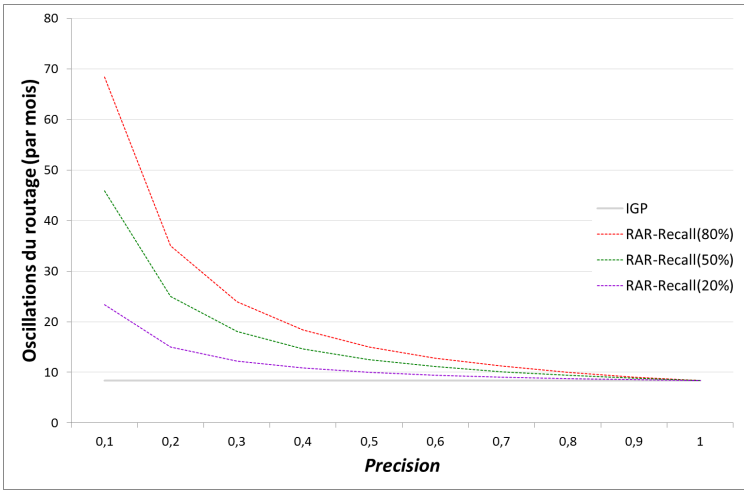
(a) A



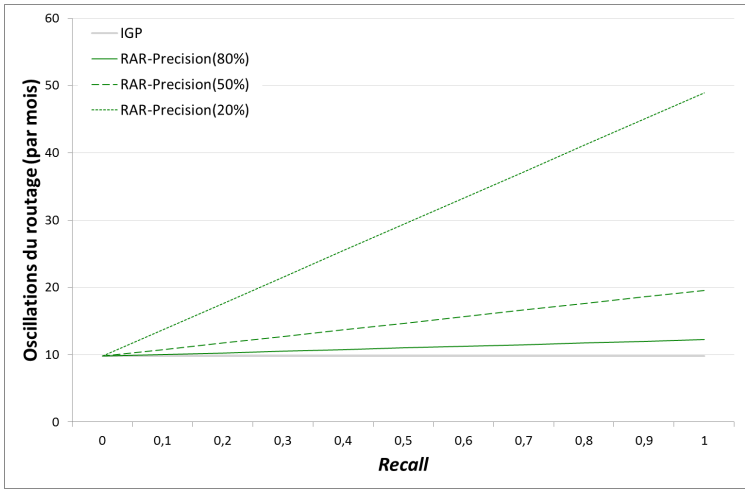
(a) A



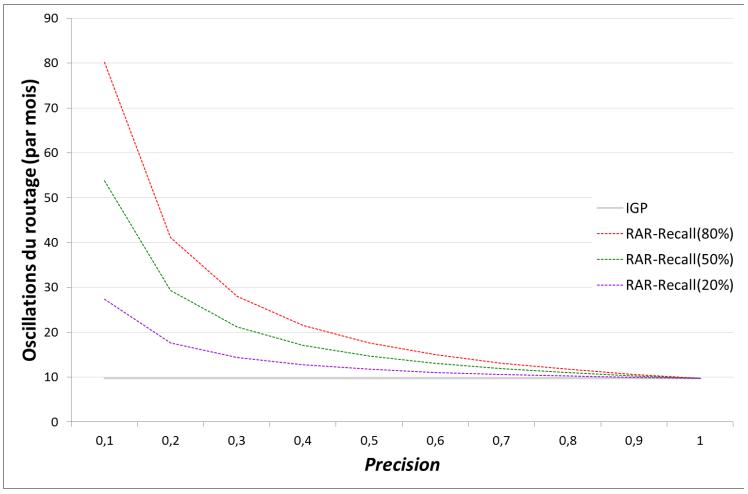
(b) US



(b) US



(c) EU

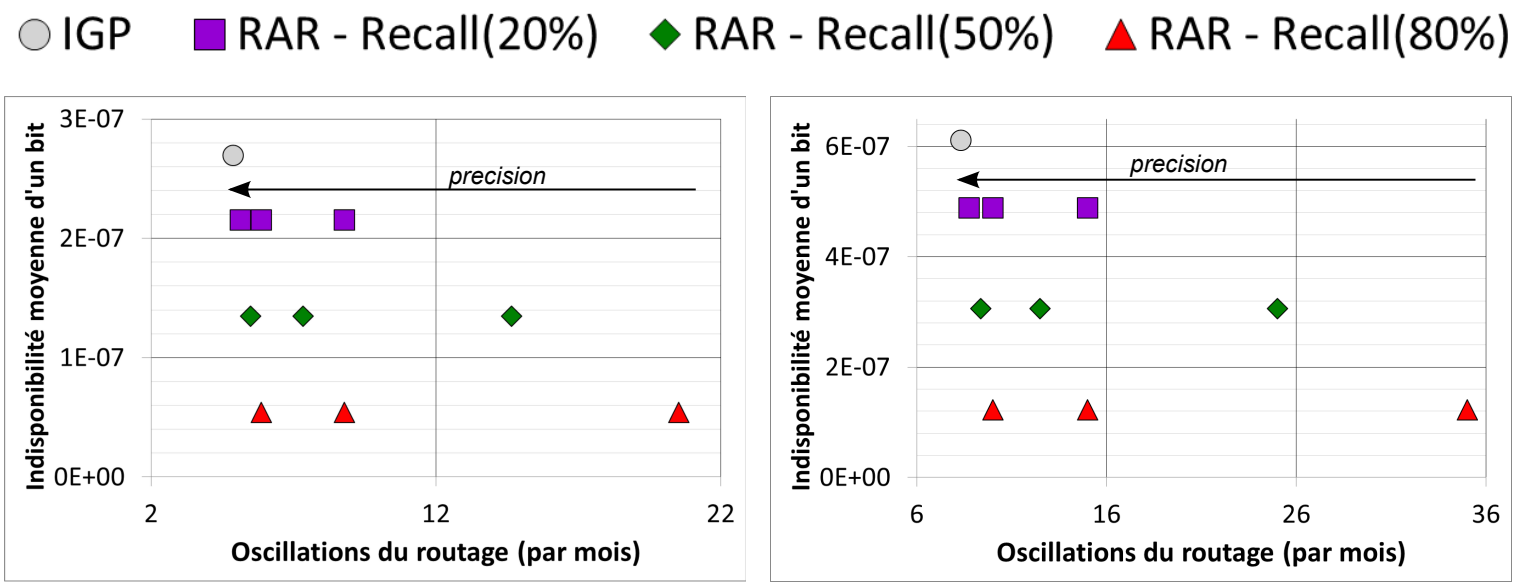


(c) EU

FIGURE B.17: Impact du *Recall* sur le nombre d'oscillations du routage.

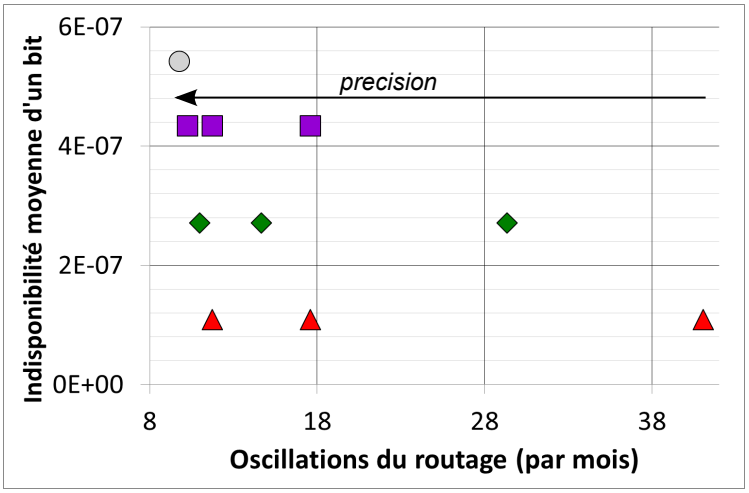
FIGURE B.18: Impact de la *Precision* sur le nombre d'oscillations du routage.

B.2.4 Étude conjointe de la disponibilité et de la stabilité du routage



(a) A

(b) US



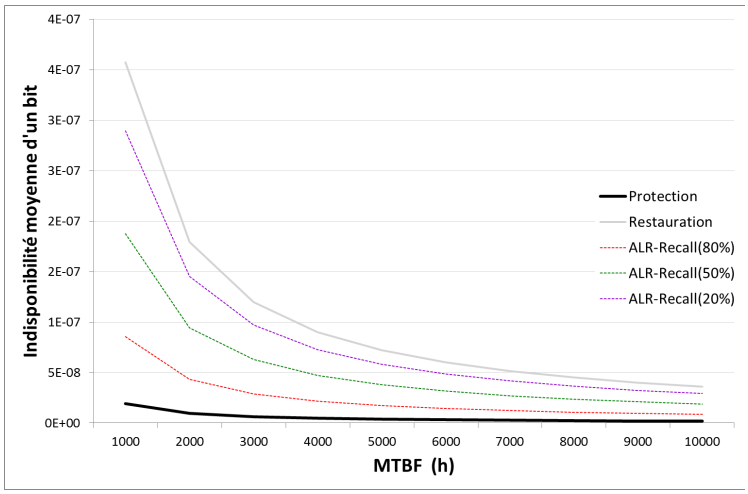
(c) EU

Figure B.19: Ratio indisponibilité / nombre d'oscillation du routage.

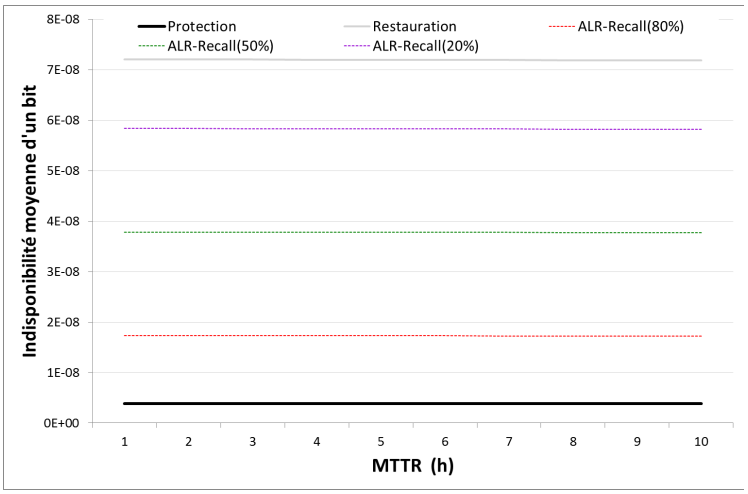
B.3 Mécanisme de résilience adaptatif (ALR)

B.3.1 Étude de la disponibilité

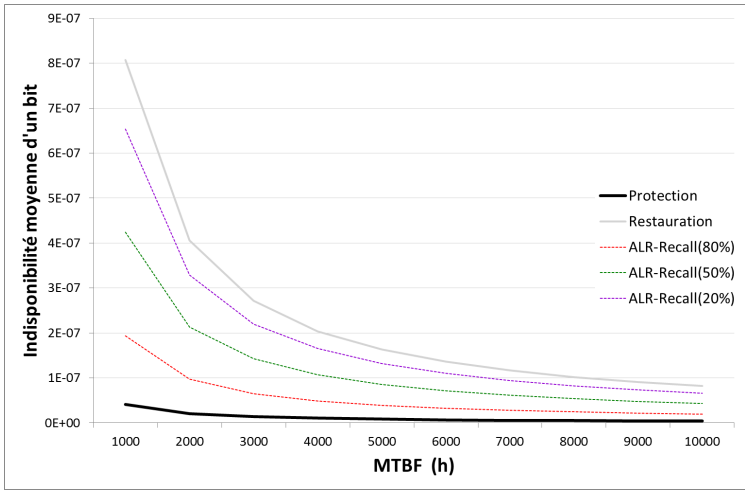
B.3.1.1 Influence de la probabilité de panne



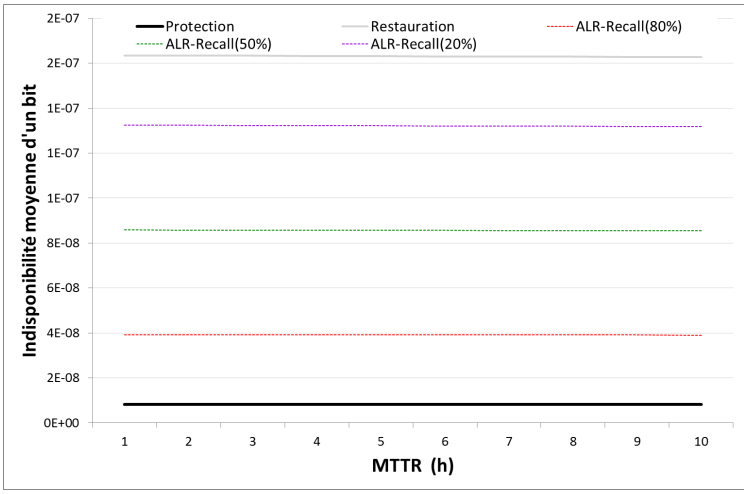
(a) A



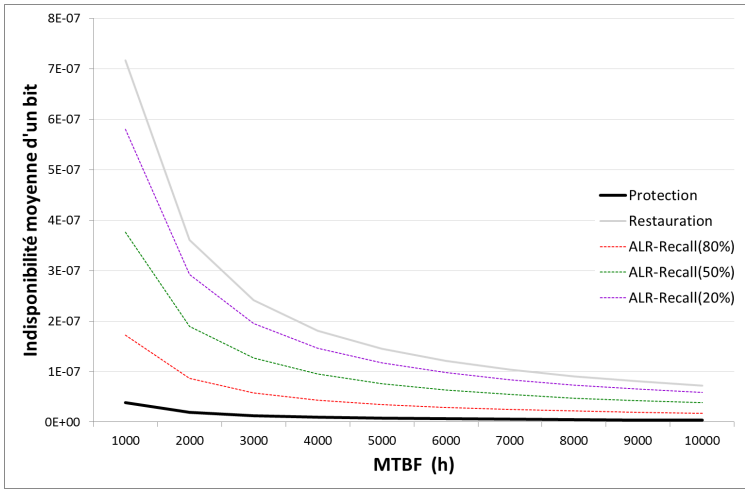
(a) A



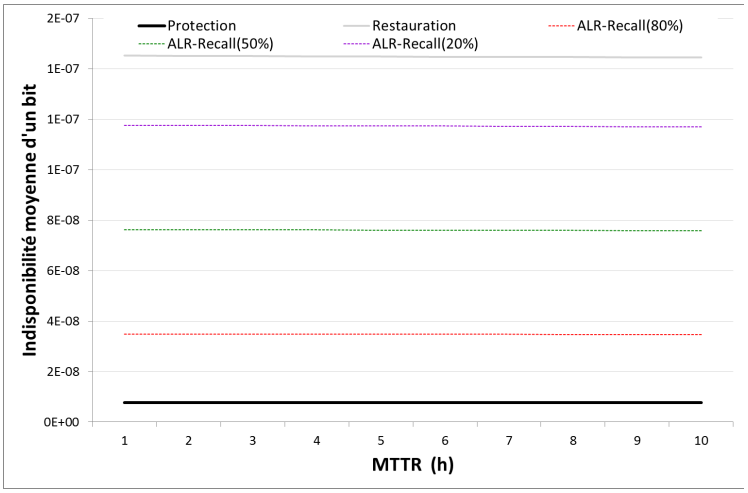
(b) US



(b) US



(c) EU

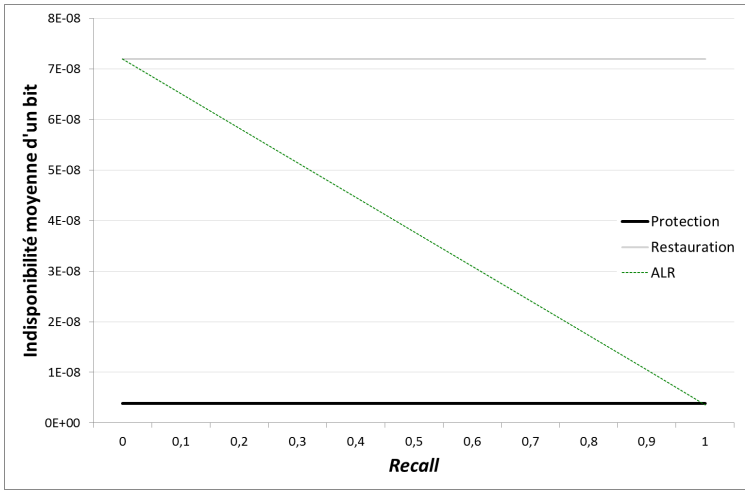


(c) EU

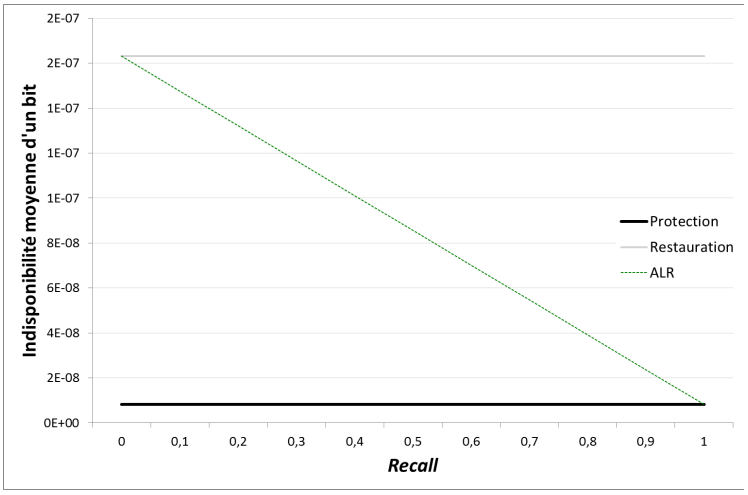
FIGURE B.20: Impact du MTBF sur la disponibilité.

FIGURE B.21: Impact du MTTR sur la disponibilité.

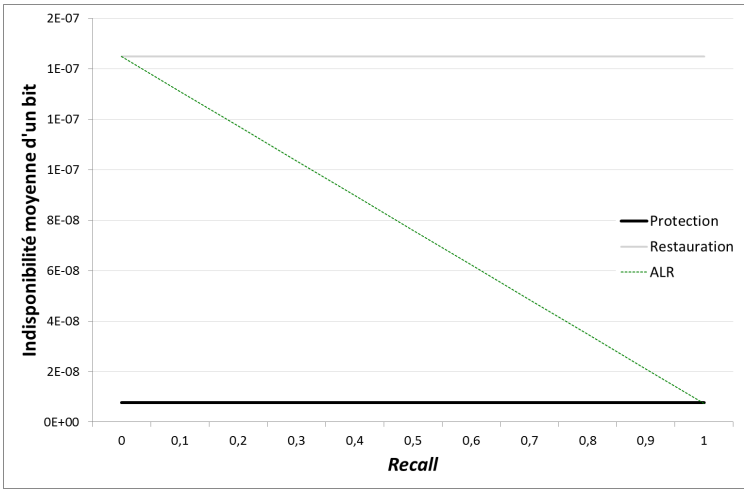
B.3.1.2 Les conséquences de la prédiction de pannes



(a) A



(b) US

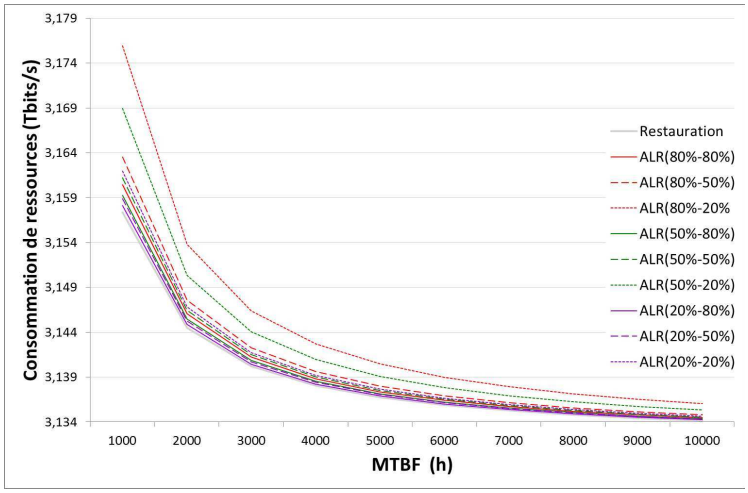


(c) EU

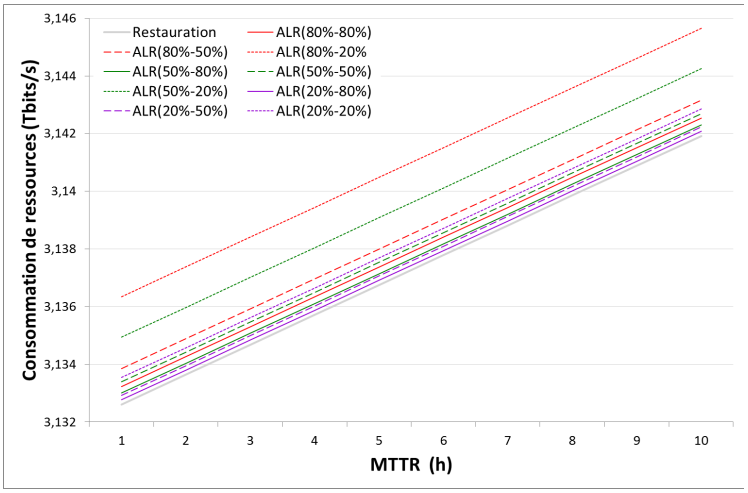
Figure B.22: Impact du *Recall* sur la disponibilité.

B.3.2 Étude de la consommation de ressources

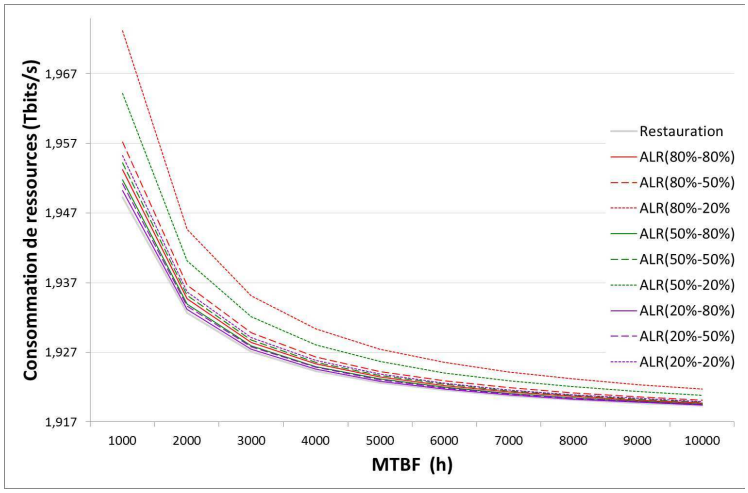
B.3.2.1 Influence de la probabilité de panne



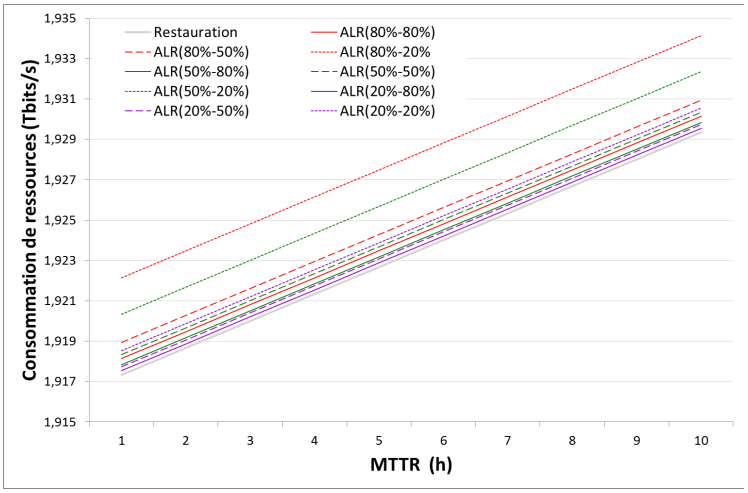
(a) A



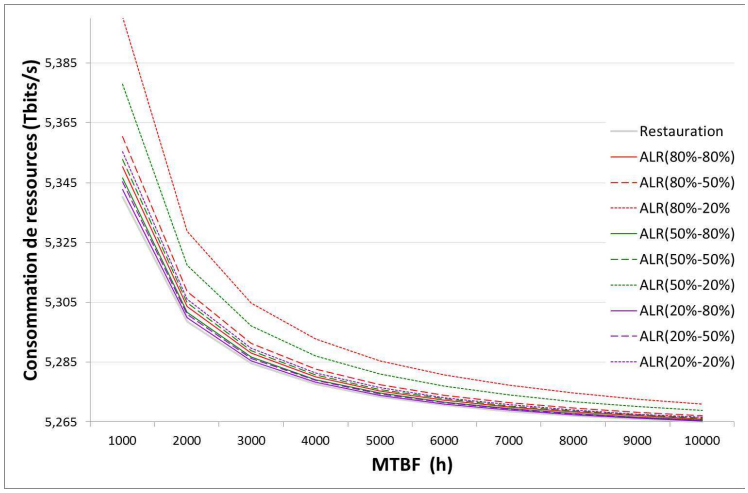
(a) A



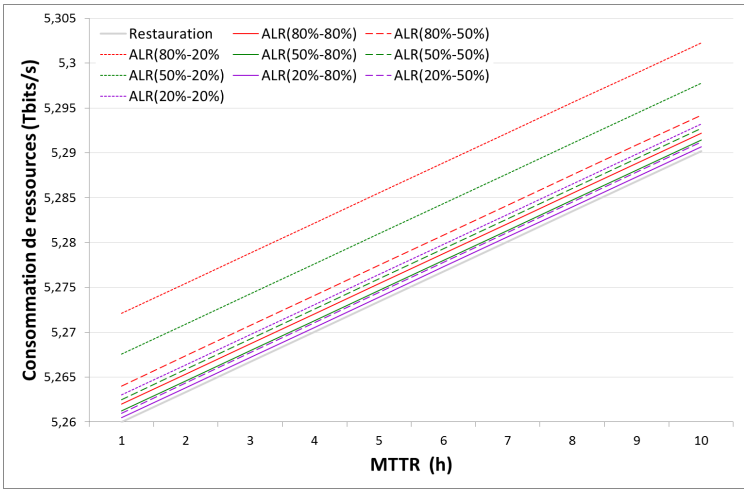
(b) US



(b) US



(c) EU

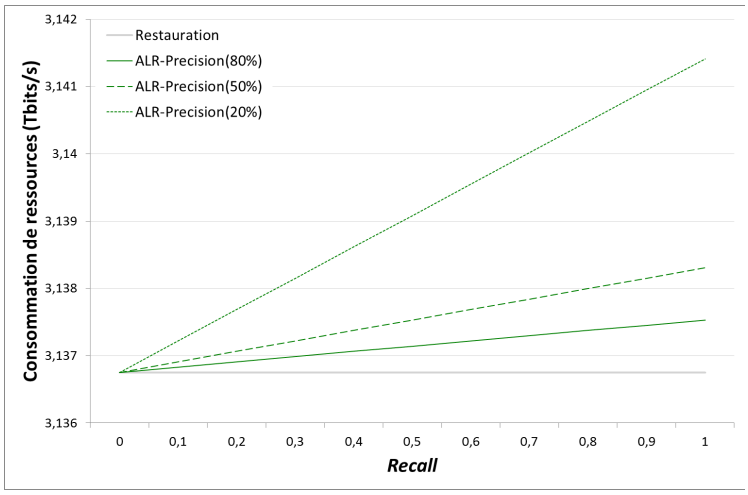


(c) EU

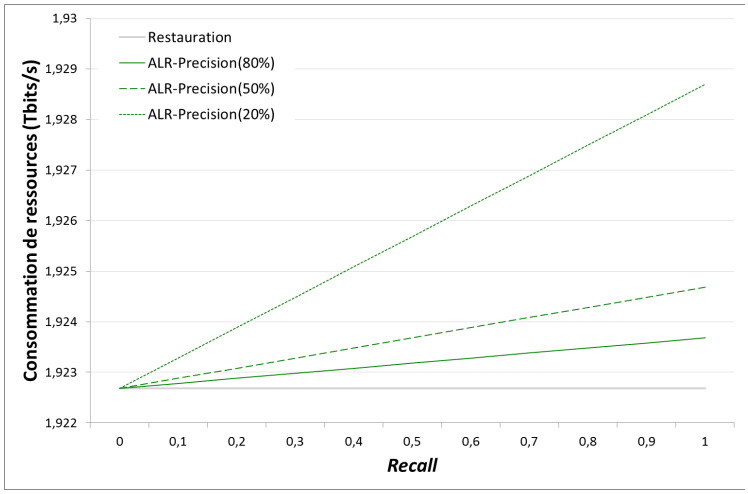
FIGURE B.23: Impact du MTBF sur l'utilisation des ressources.

FIGURE B.24: Impact du MTTR sur l'utilisation des ressources.

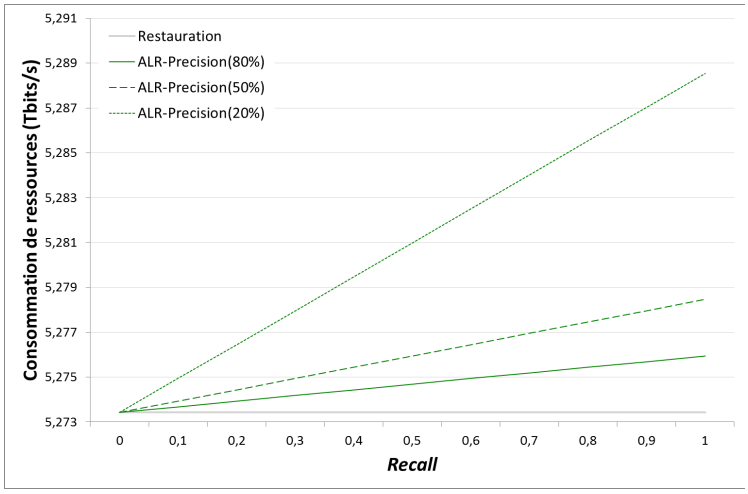
B.3.2.2 Les conséquences de la prédiction de pannes



(a) A

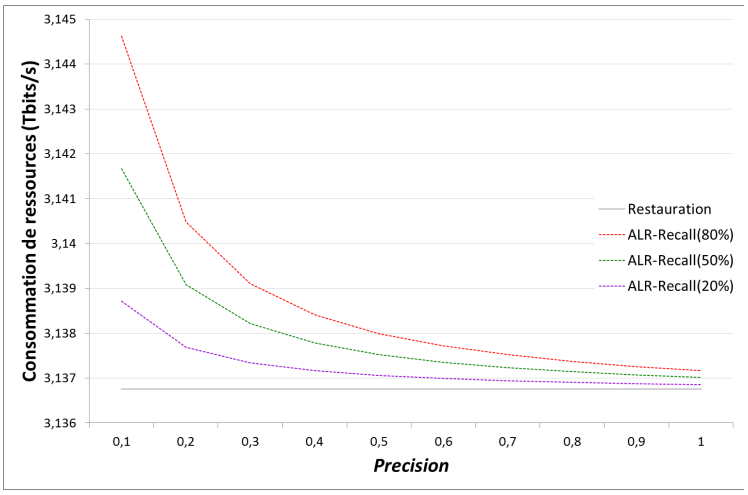


(b) US

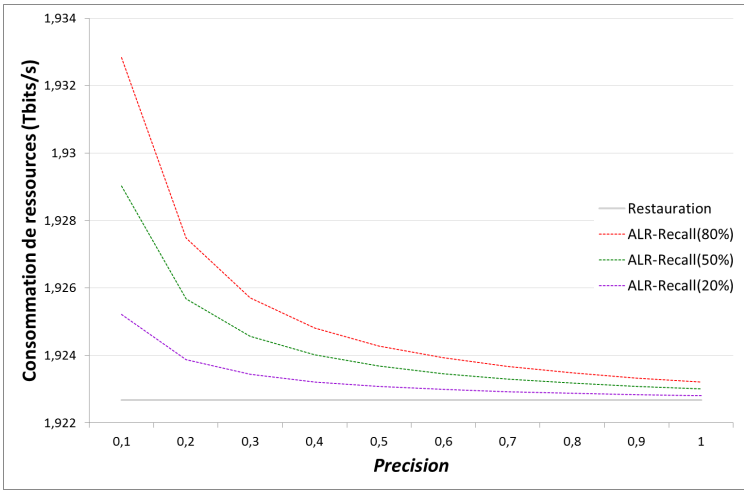


(c) EU

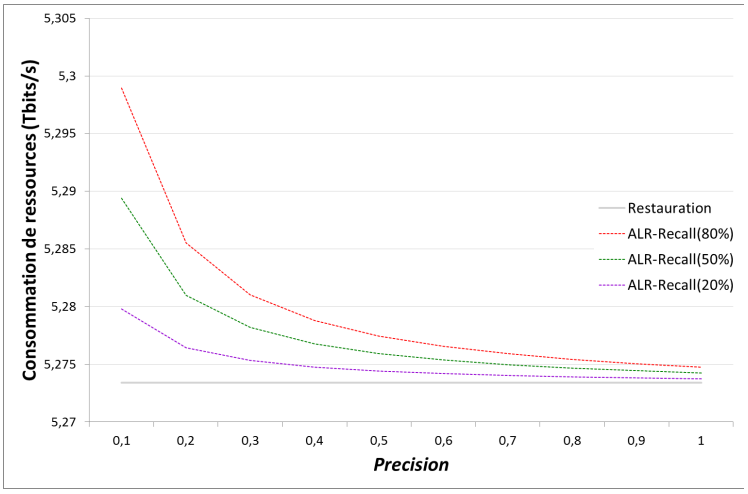
FIGURE B.25: Impact du *Recall* sur l'utilisation des ressources.



(a) A

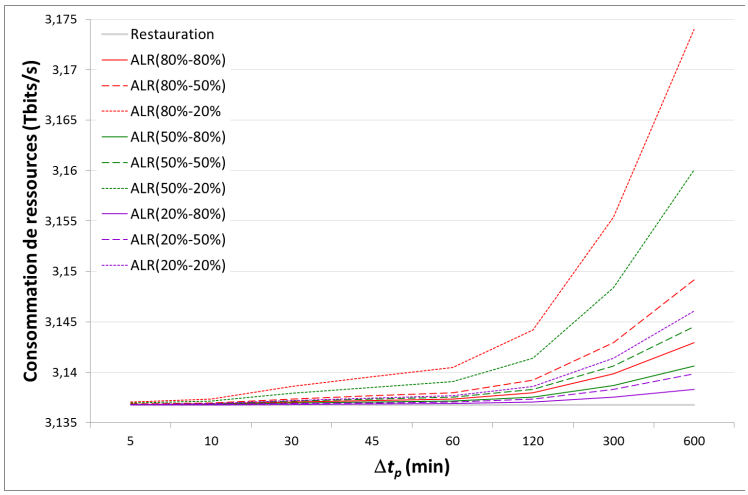


(b) US

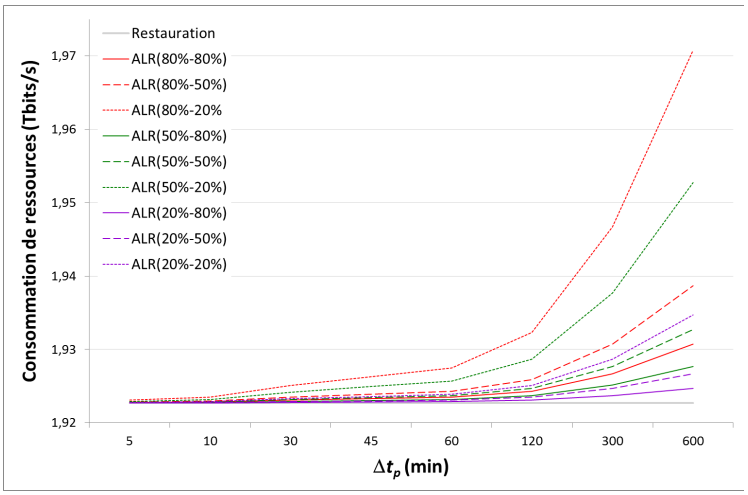


(c) EU

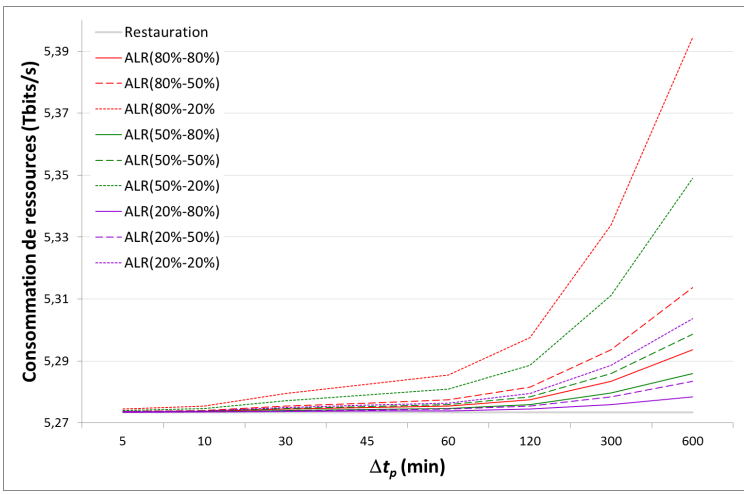
FIGURE B.26: Impact de la *Precision* sur l'utilisation des ressources.



(a) A



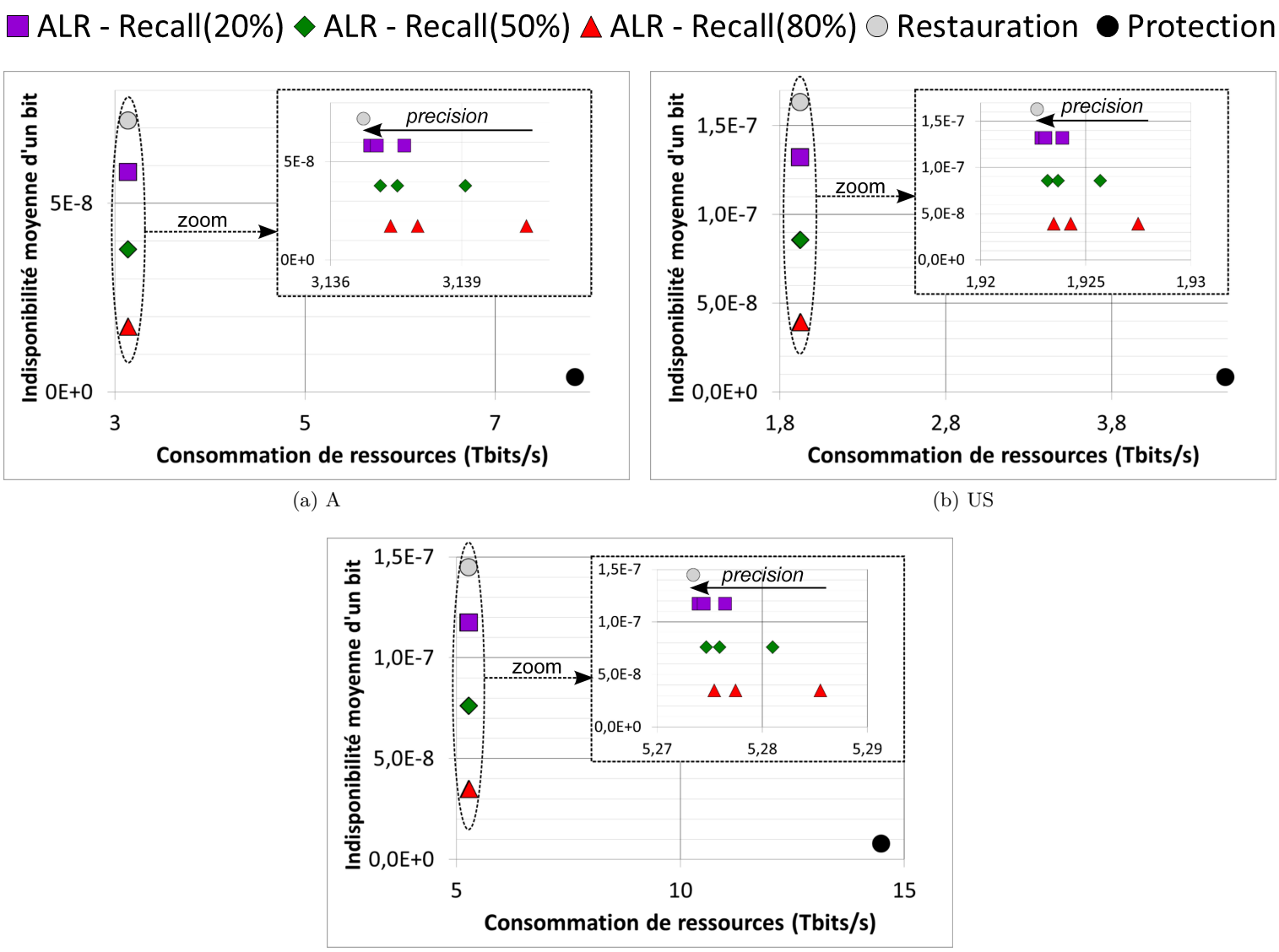
(b) US



(c) EU

FIGURE B.27: Impact de Δt_p sur l'utilisation des ressources.

B.3.3 Étude conjointe de la disponibilité et de la consommation de ressources



(a) A

(b) US

(c) EU

Figure B.28: Ratio indisponibilité / utilisation des ressources.

Annexe C

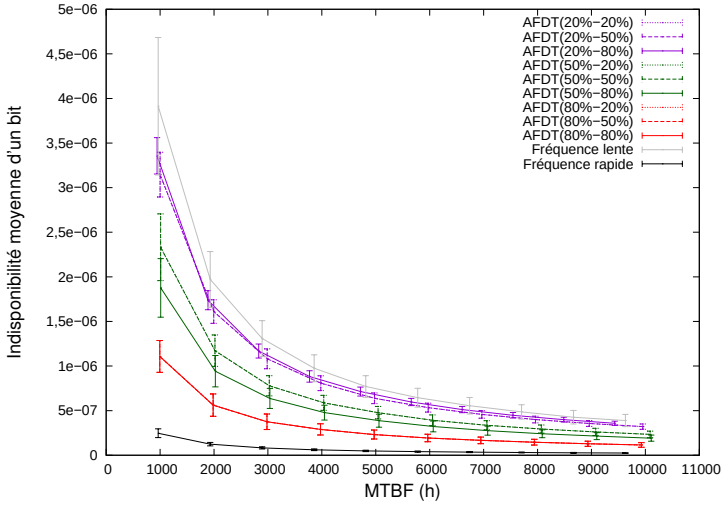
Résultats de simulation

Sommaire

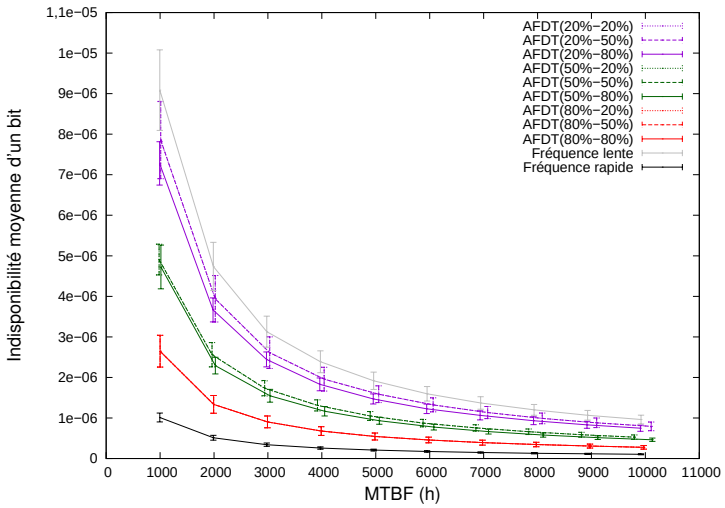
C.1 Détection autonome des pannes (AFDT)

C.1.1 Étude de la disponibilité

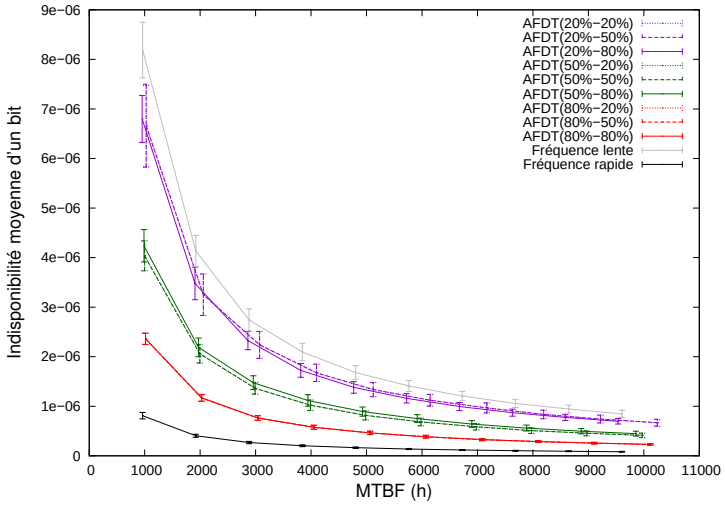
C.1.1.1 Influence de la probabilité de panne



(a) A

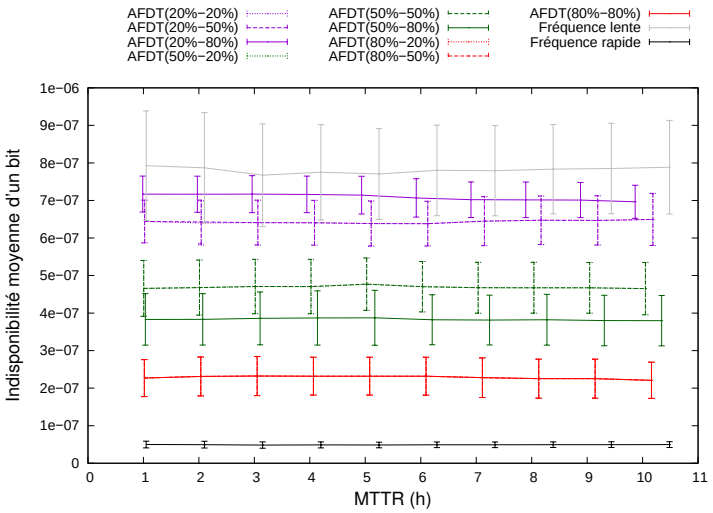


(b) US

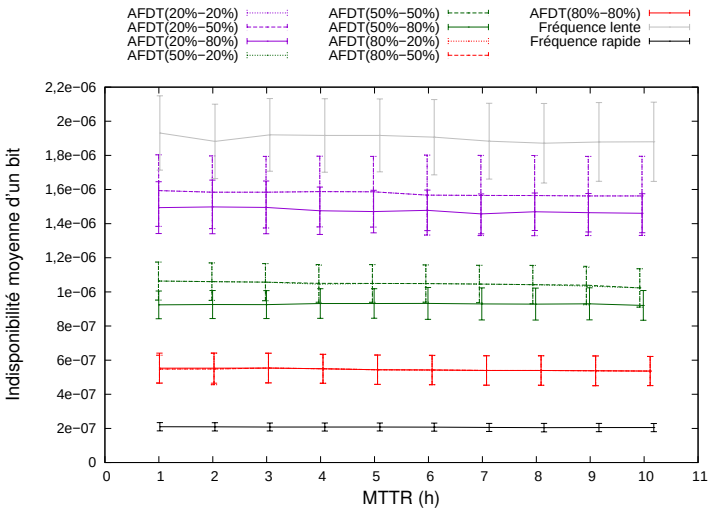


(c) EU

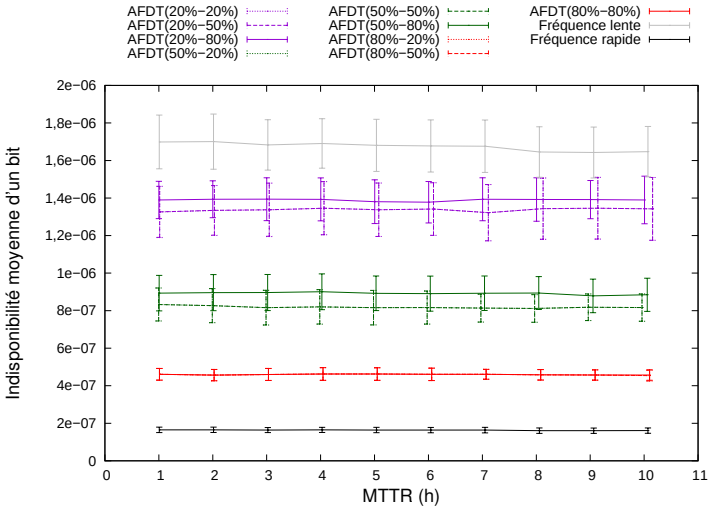
FIGURE C.1: Impact du MTBF sur la disponibilité.



(a) A



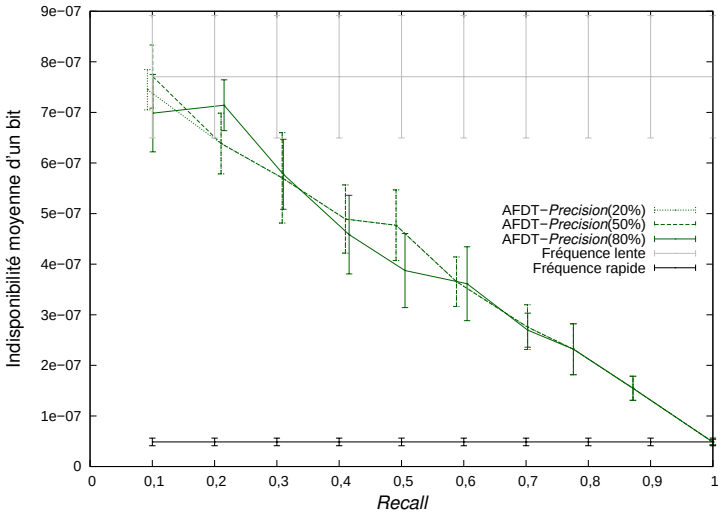
(b) US



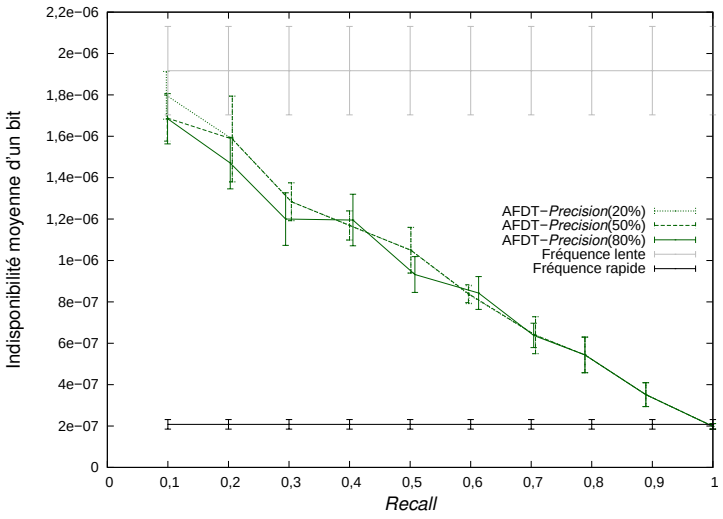
(c) EU

FIGURE C.2: Impact du MTTR sur la disponibilité.

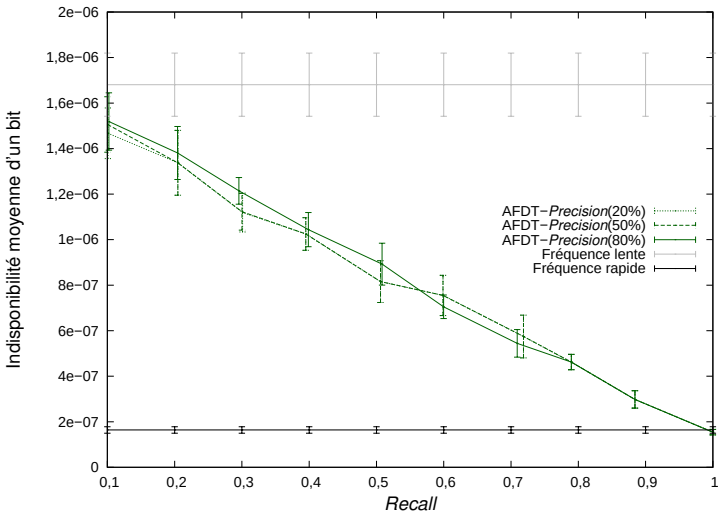
C.1.1.2 Les conséquences de la prédiction de pannes



(a) A

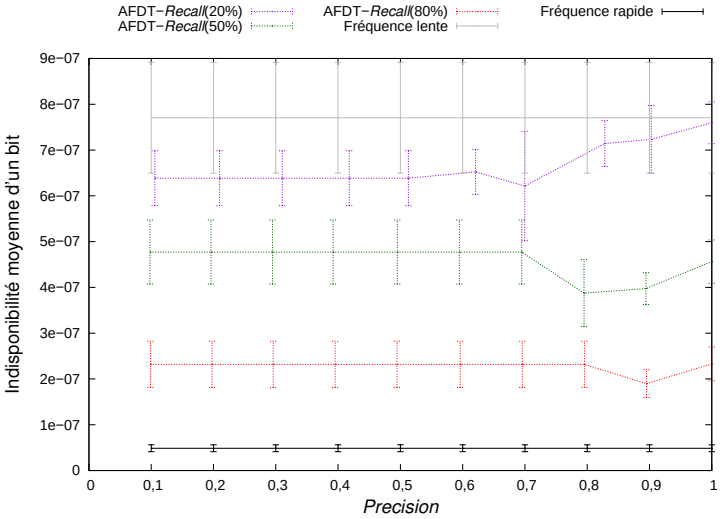


(b) US

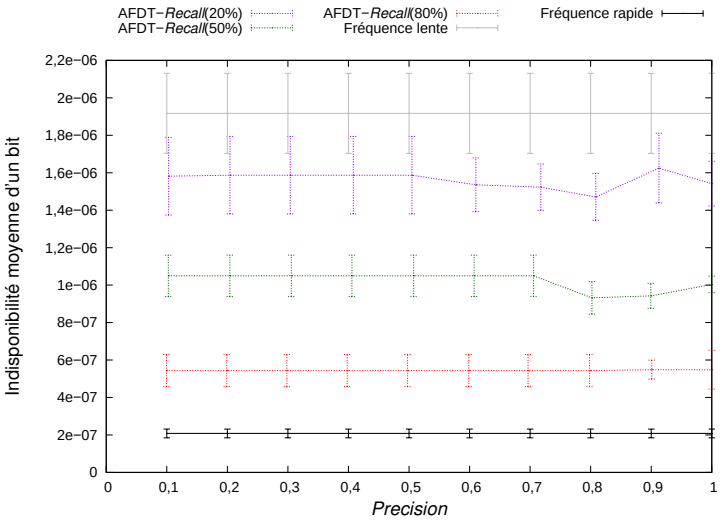


(c) EU

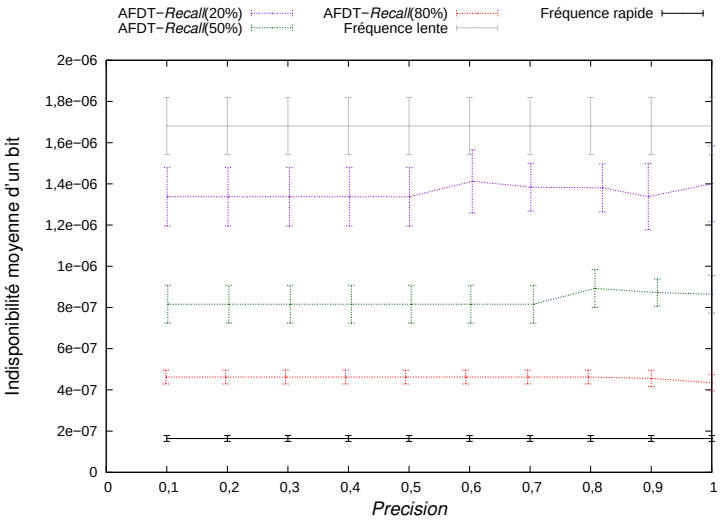
FIGURE C.3: Impact du *Recall* sur la disponibilité.



(a) A

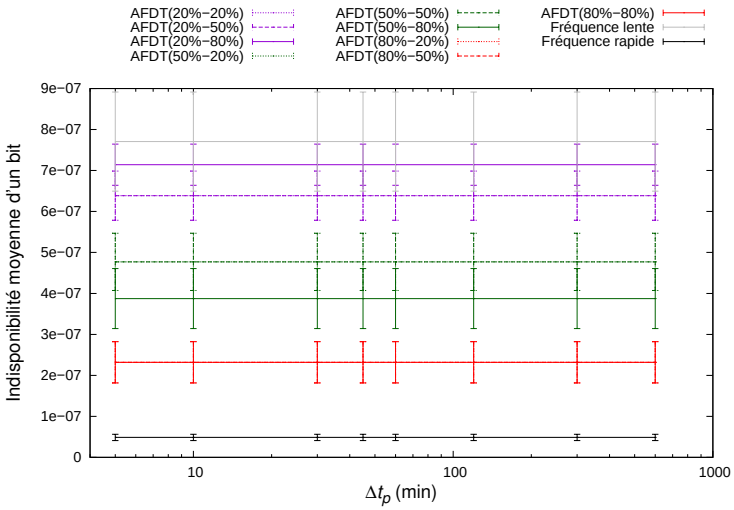


(b) US

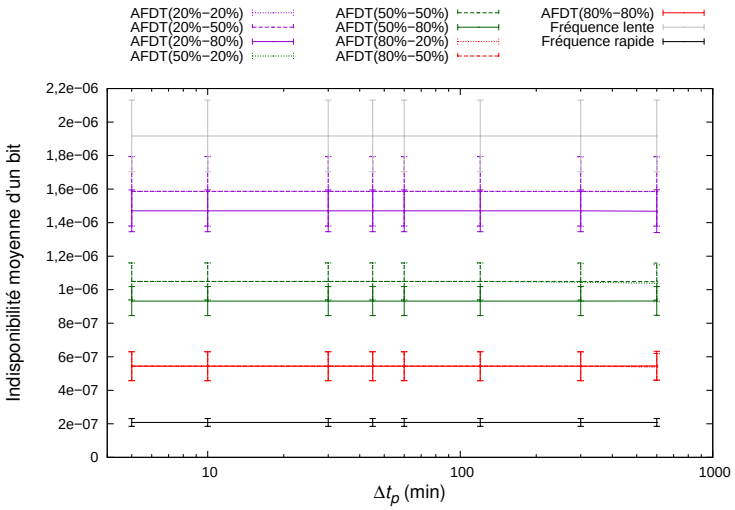


(c) EU

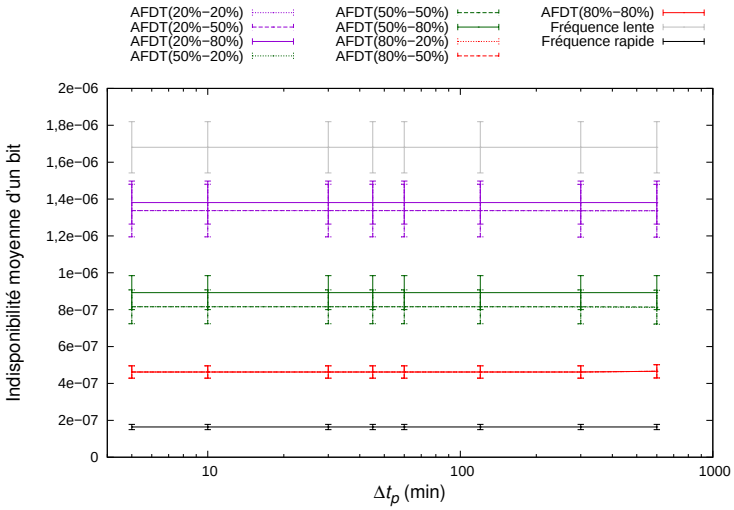
FIGURE C.4: Impact de la *Precision* sur la disponibilité.



(a) A



(b) US

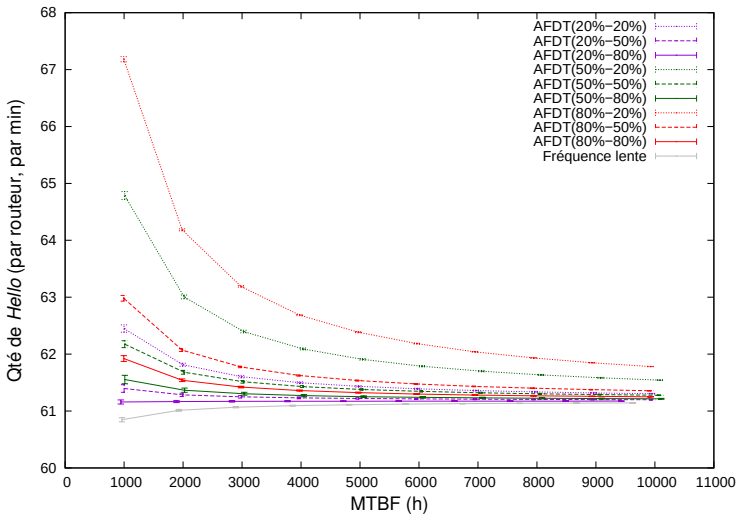


(c) EU

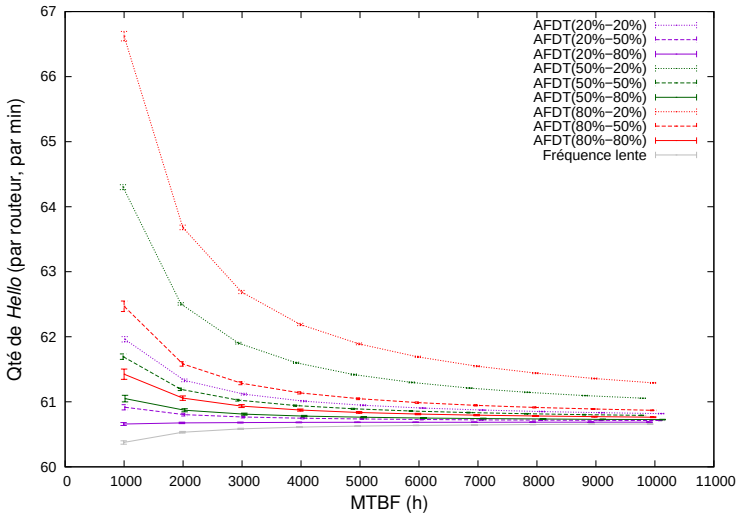
FIGURE C.5: Impact de Δt_p sur la disponibilité.

C.1.2 Étude de la quantité de message de contrôle à traiter

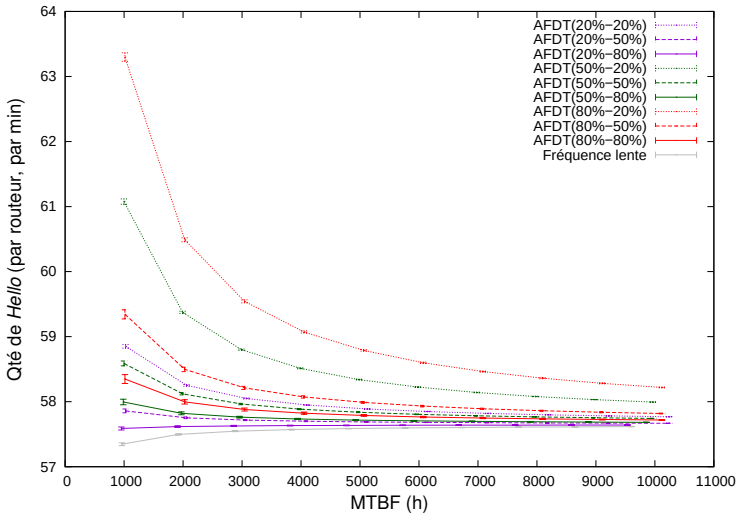
C.1.2.1 Influence de la probabilité de panne



(a) A

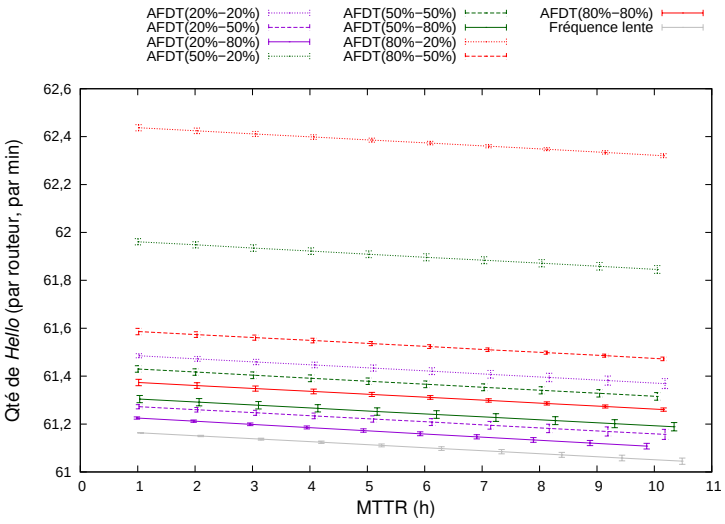


(b) US

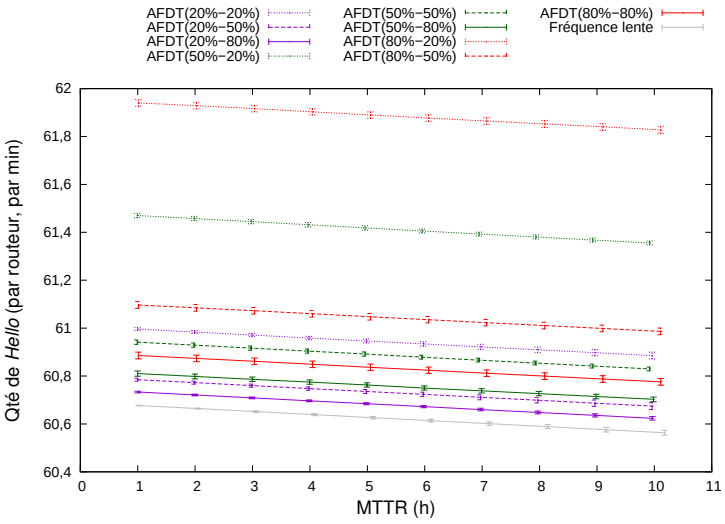


(c) EU

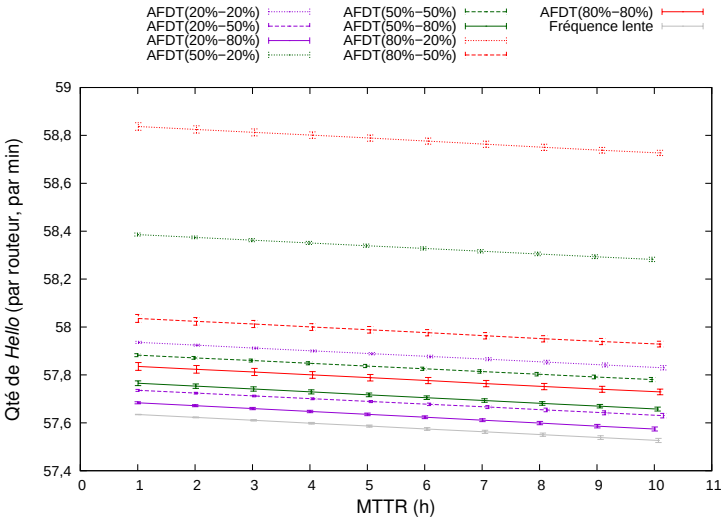
FIGURE C.6: Impact du MTBF sur le nombre de messages *Hello*.



(a) A

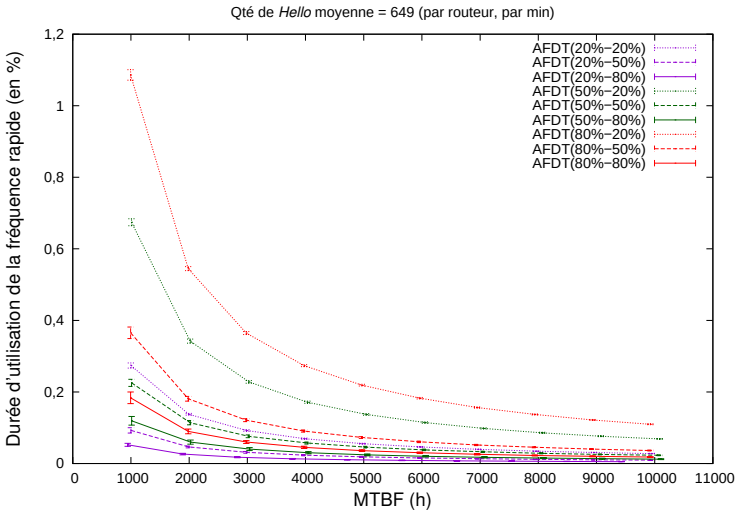


(b) US

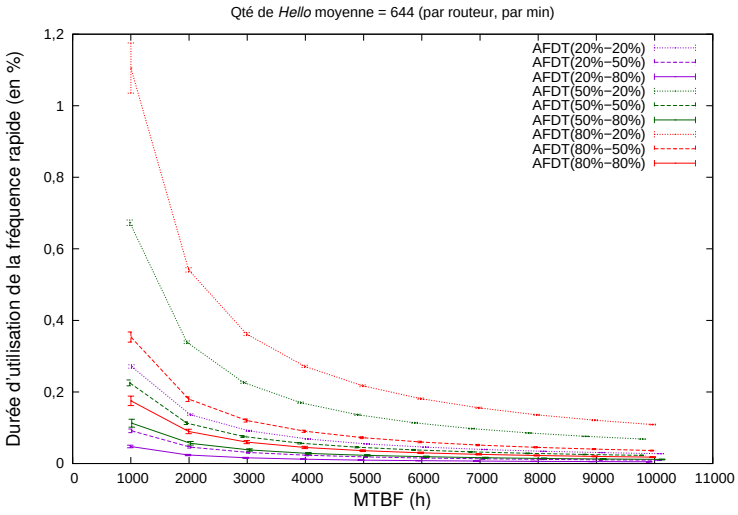


(c) EU

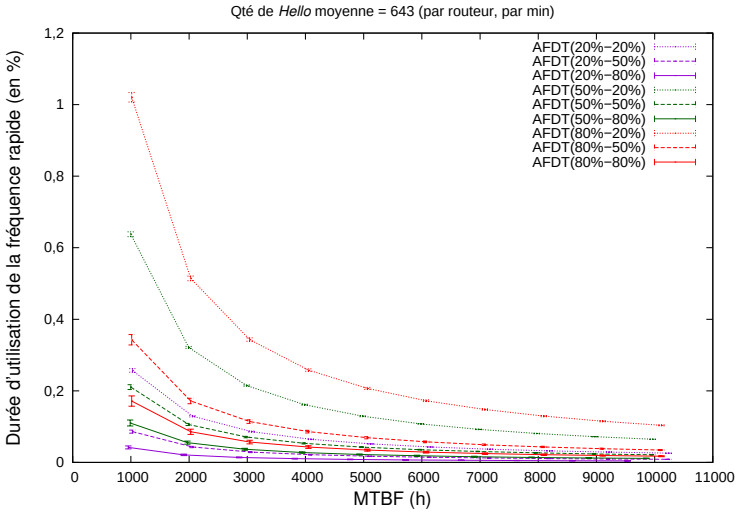
FIGURE C.7: Impact du MTTR sur le nombre de messages *Hello*.



(a) A



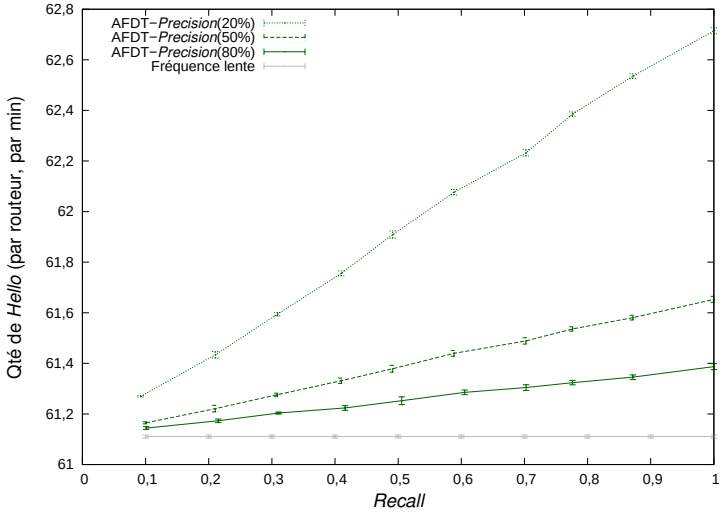
(b) US



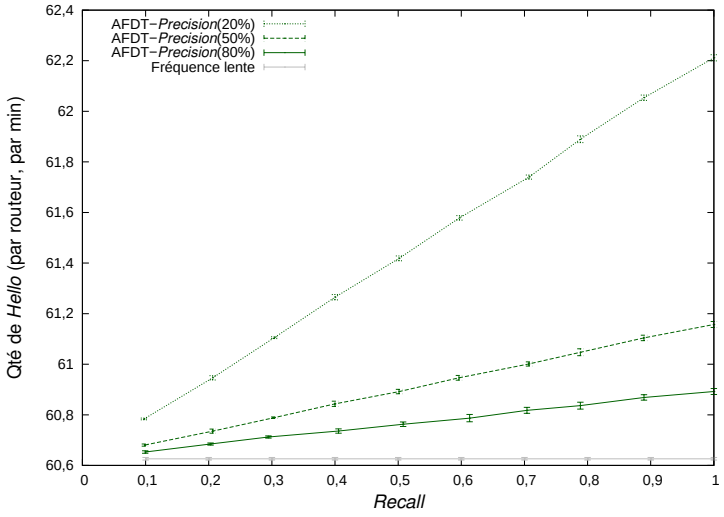
(c) EU

FIGURE C.8: Impact du MTBF sur la durée de réception des messages *Hello* à une fréquence élevée.

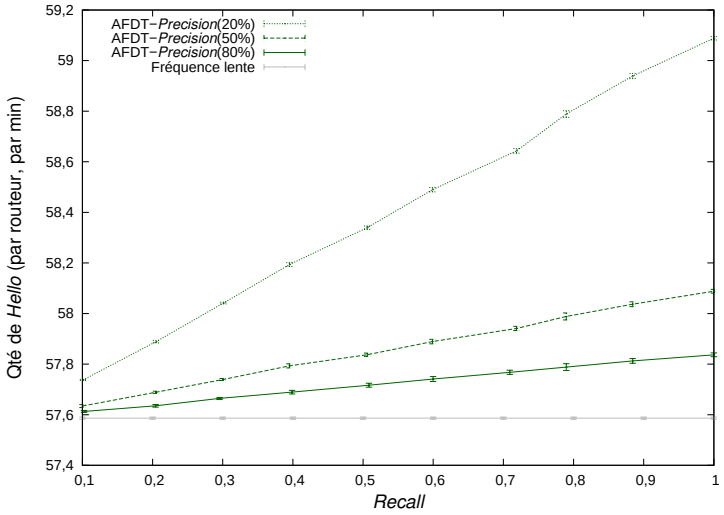
C.1.2.2 Les conséquences de la prédiction de pannes



(a) A

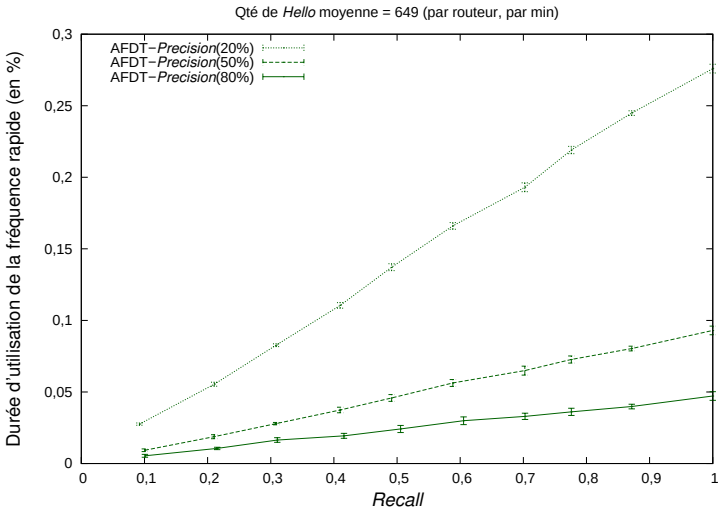


(b) US

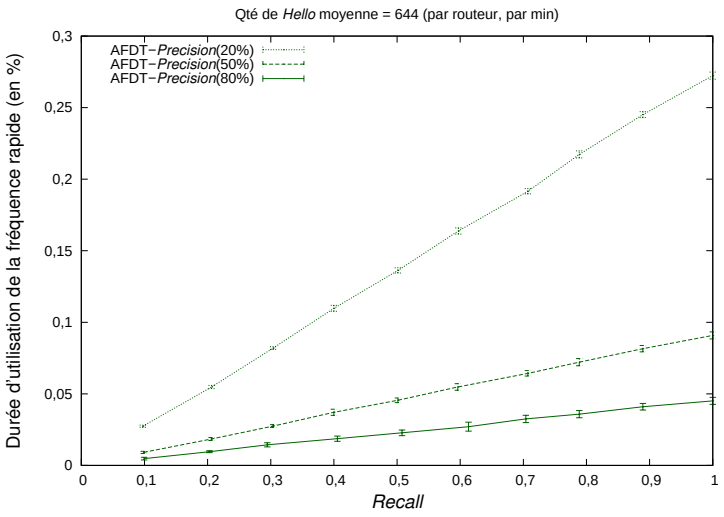


(c) EU

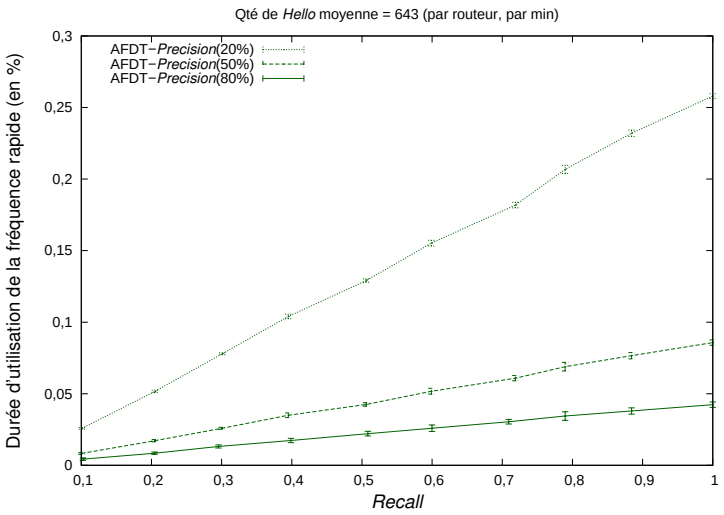
FIGURE C.9: Impact du *Recall* sur le nombre de messages *Hello*.



(a) A

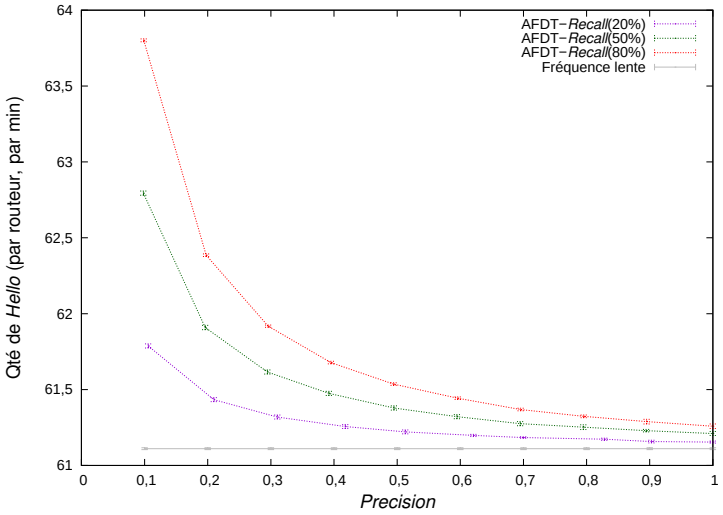


(b) US

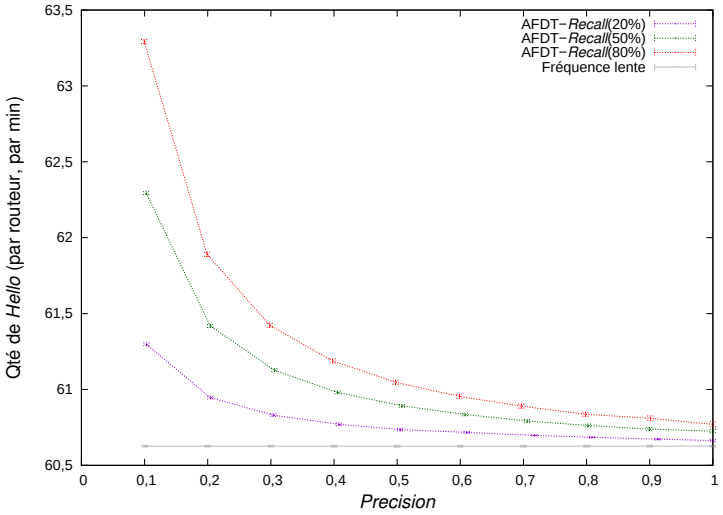


(c) EU

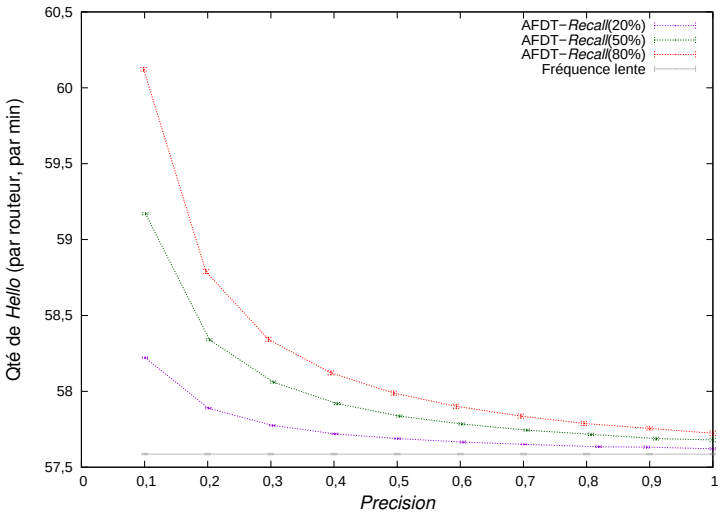
FIGURE C.10: Impact du *Recall* sur la durée de réception des messages *Hello* à une fréquence élevée.



(a) A

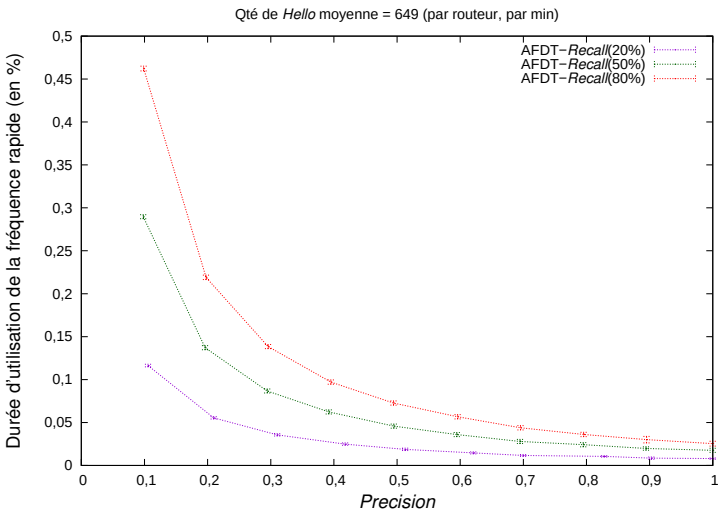


(b) US

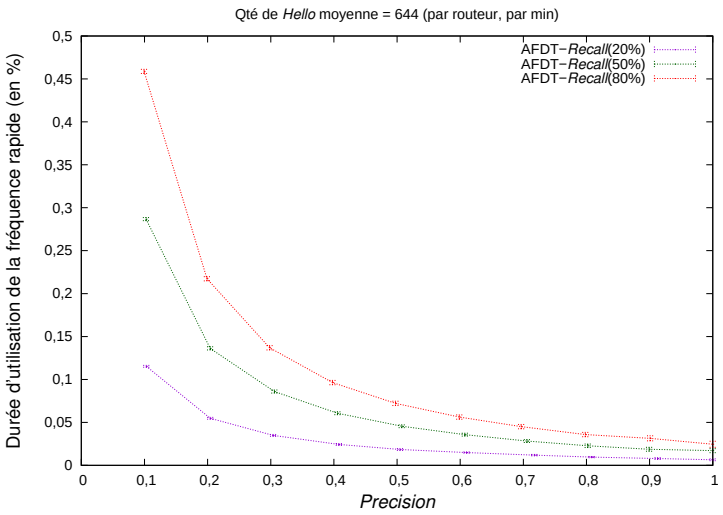


(c) EU

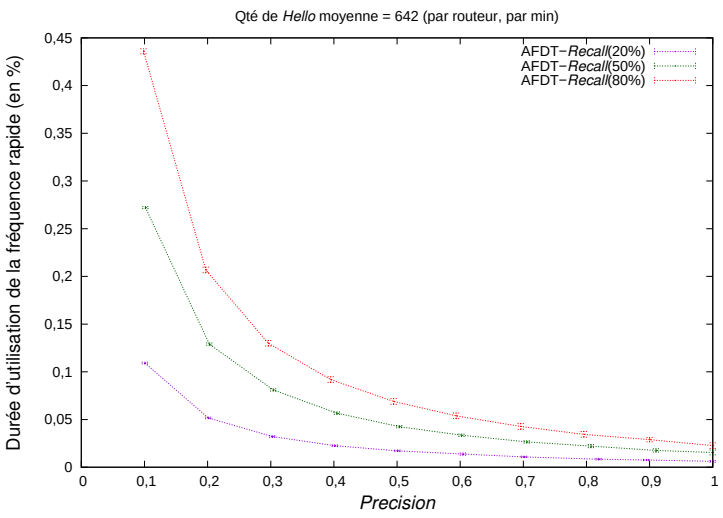
FIGURE C.11: Impact de la *Precision* sur le nombre de messages *Hello*.



(a) A

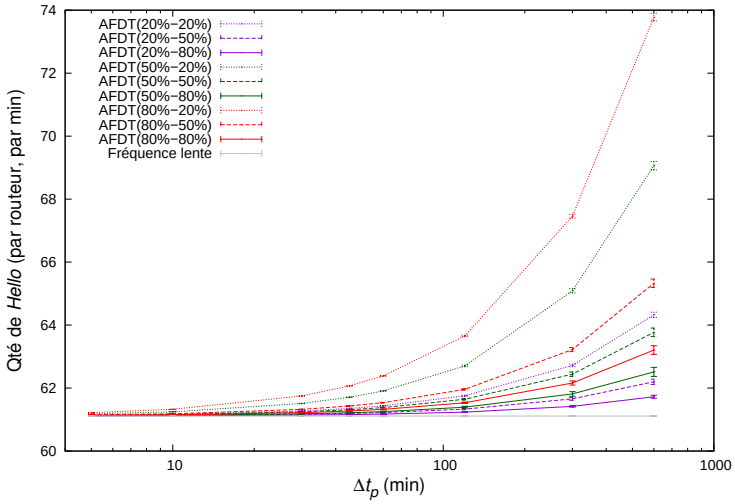


(b) US

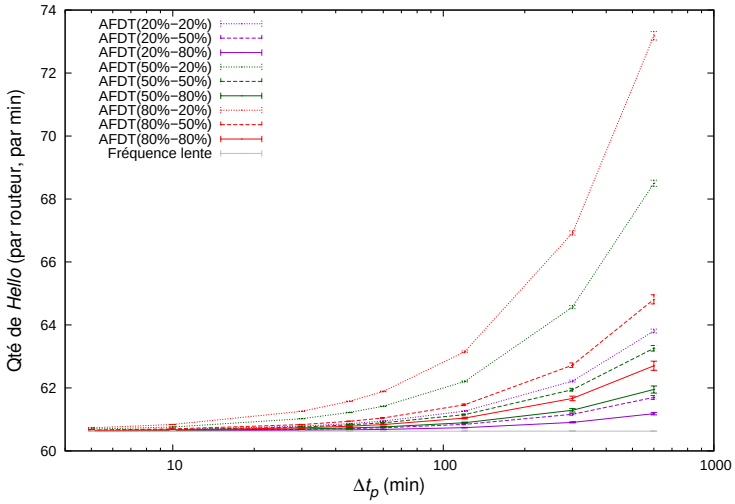


(c) EU

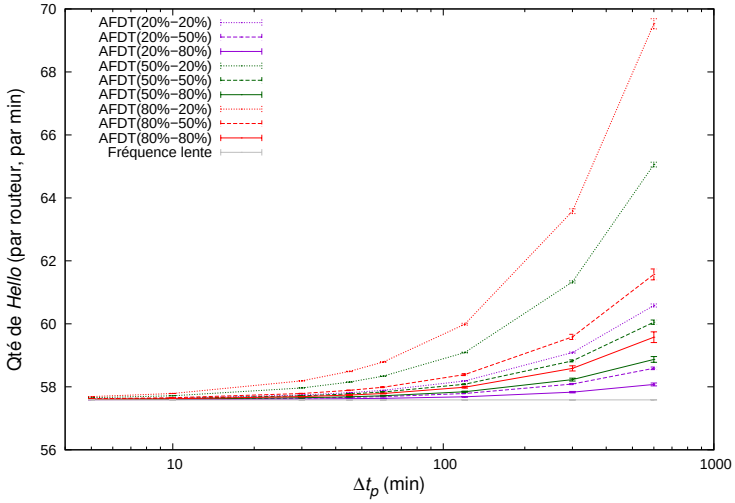
FIGURE C.12: Impact de la *Precision* sur la durée de réception des messages *Hello* à une fréquence élevée.



(a) A

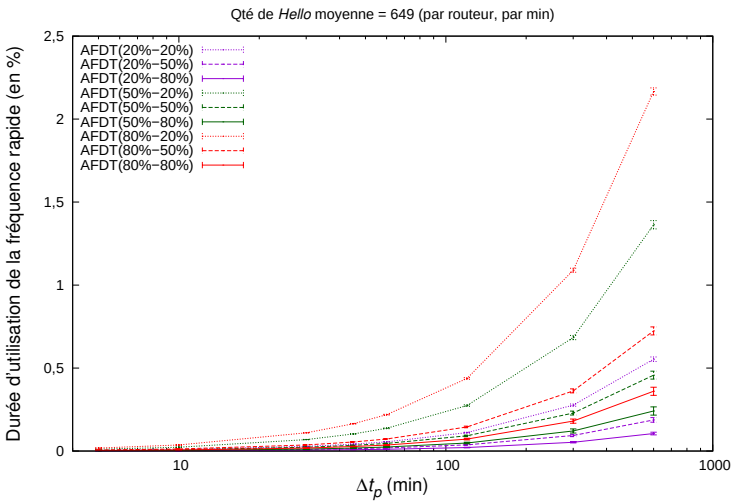


(b) US

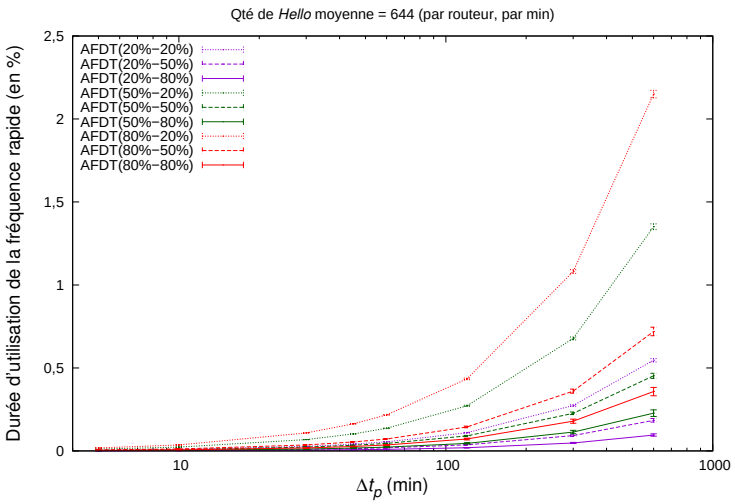


(c) EU

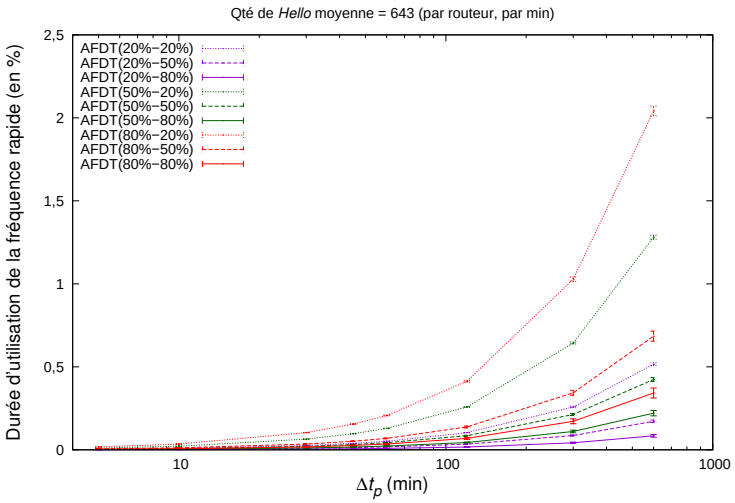
FIGURE C.13: Impact de Δt_p sur le nombre de messages *Hello*.



(a) A



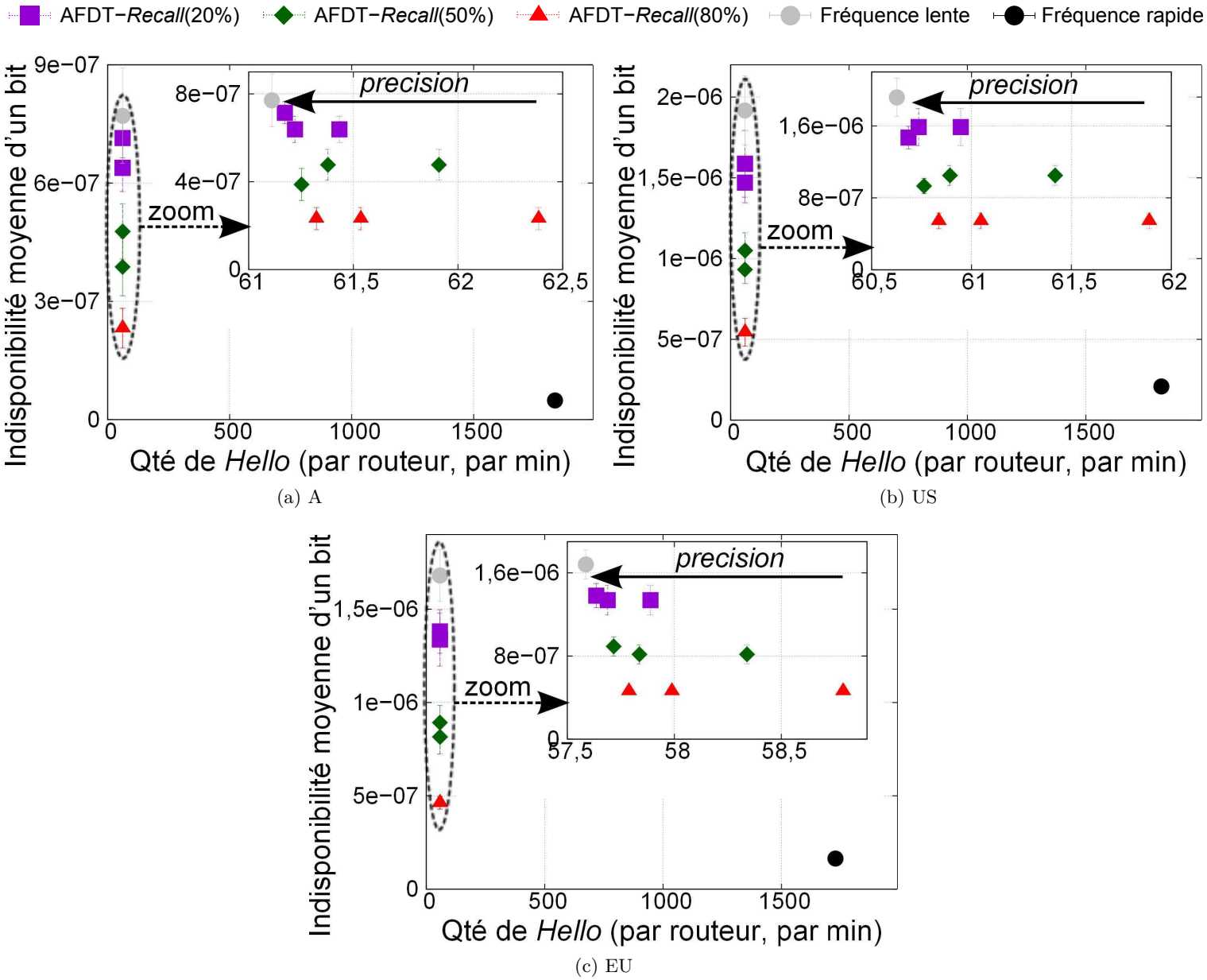
(b) US



(c) EU

FIGURE C.14: Impact de Δt_p sur la durée de réception des messages *Hello* à une fréquence élevée.

C.1.3 Étude conjointe de la disponibilité et de la quantité de message de contrôle à traiter



(a) A

(b) US

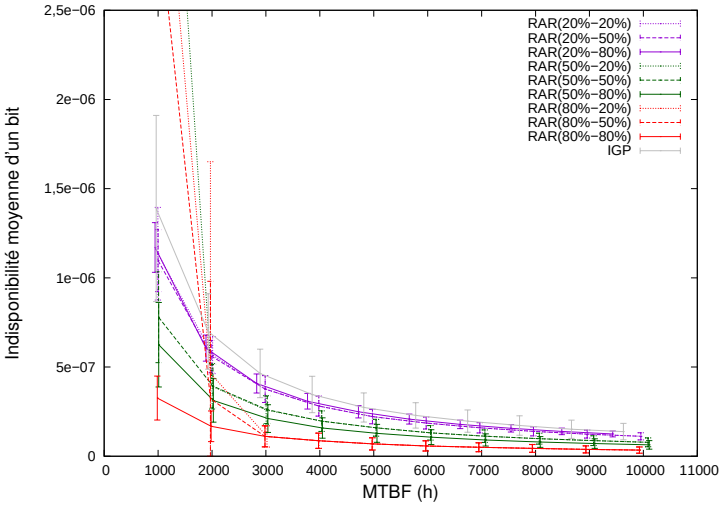
(c) EU

FIGURE C.15: Ratio indisponibilité / nombre de message *Hello*.

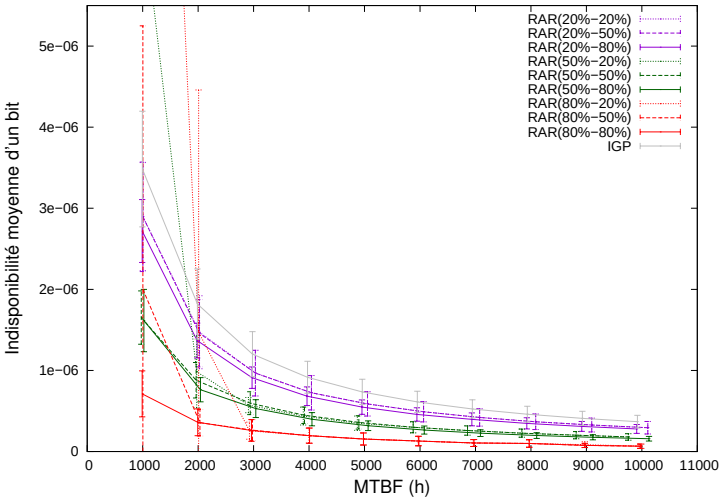
C.2 Routage sensible au risque de pannes (RAR)

C.2.1 Étude de la disponibilité

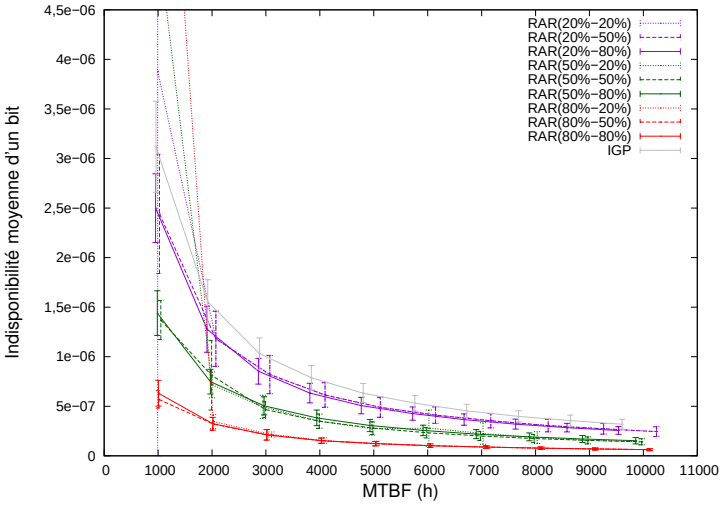
C.2.1.1 Influence de la probabilité de panne



(a) A

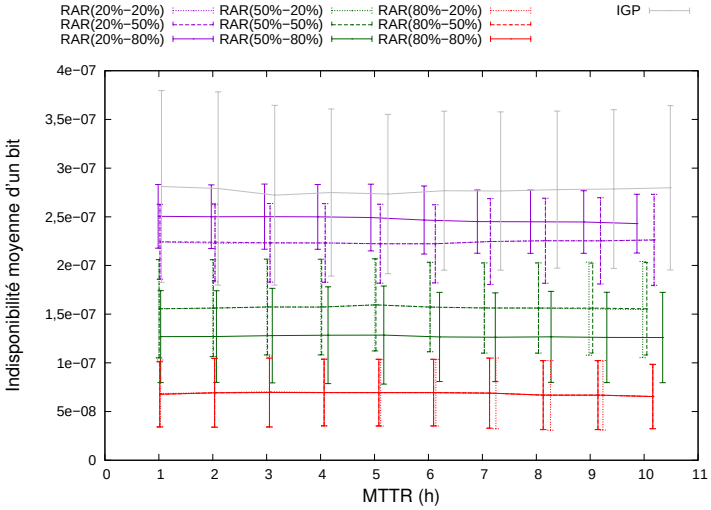


(b) US

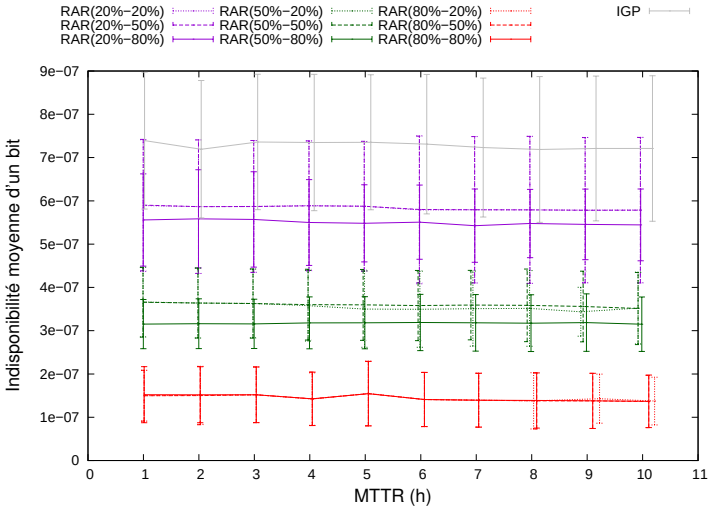


(c) EU

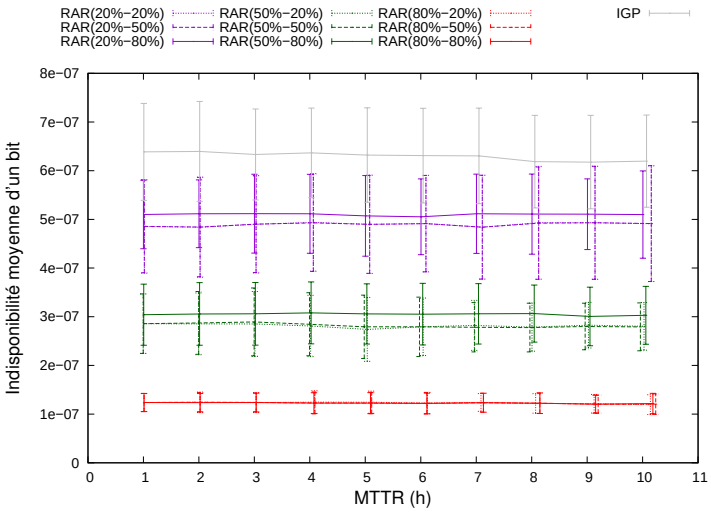
FIGURE C.16: Impact du MTBF sur la disponibilité.



(a) A



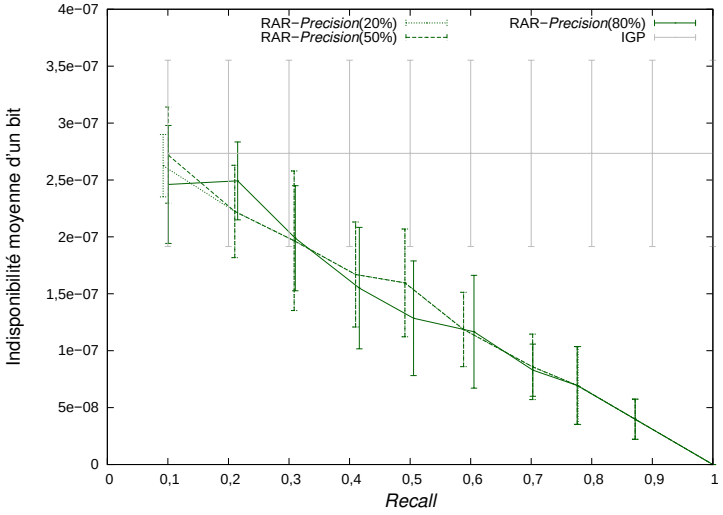
(b) US



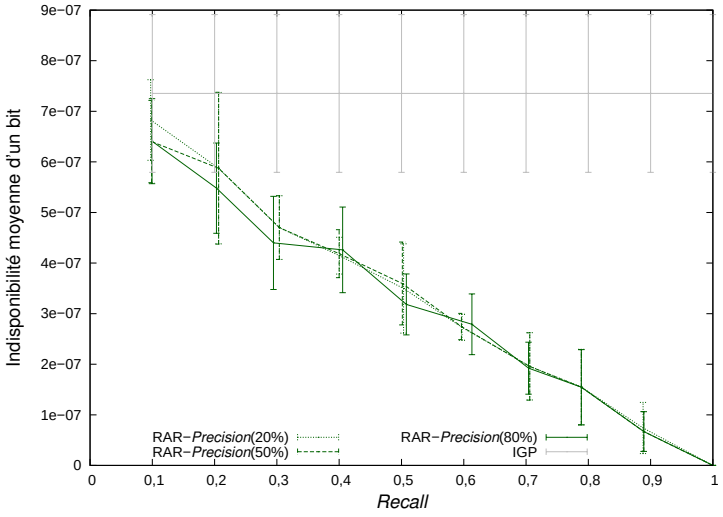
(c) EU

FIGURE C.17: Impact du MTTR sur la disponibilité.

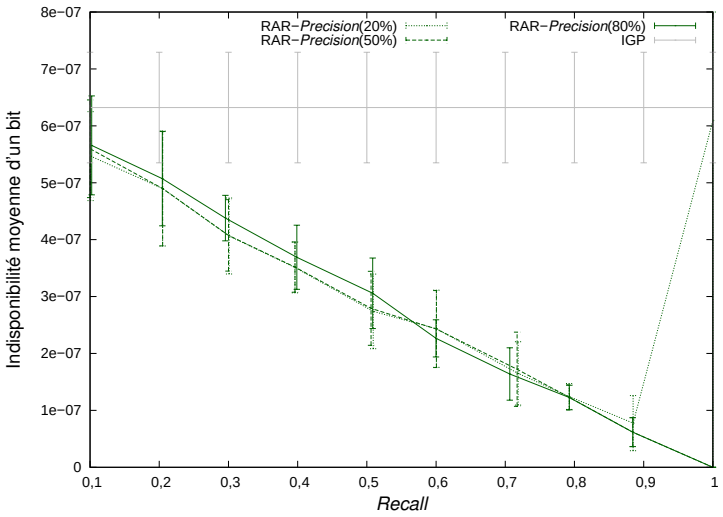
C.2.1.2 Les conséquences de la prédiction de pannes



(a) A

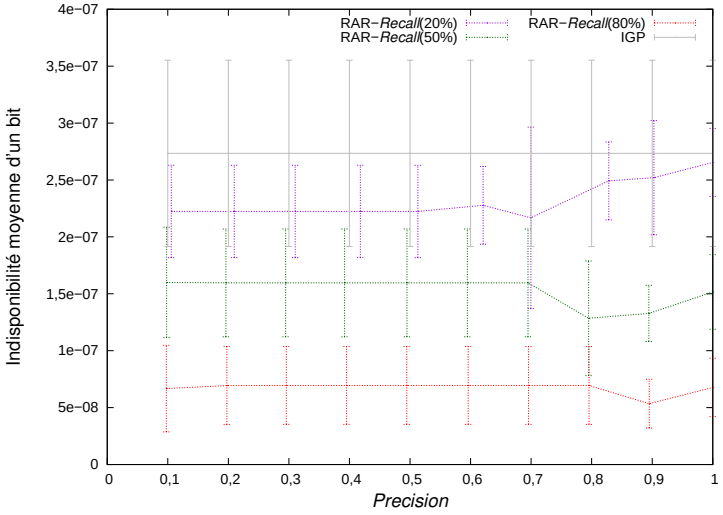


(b) US

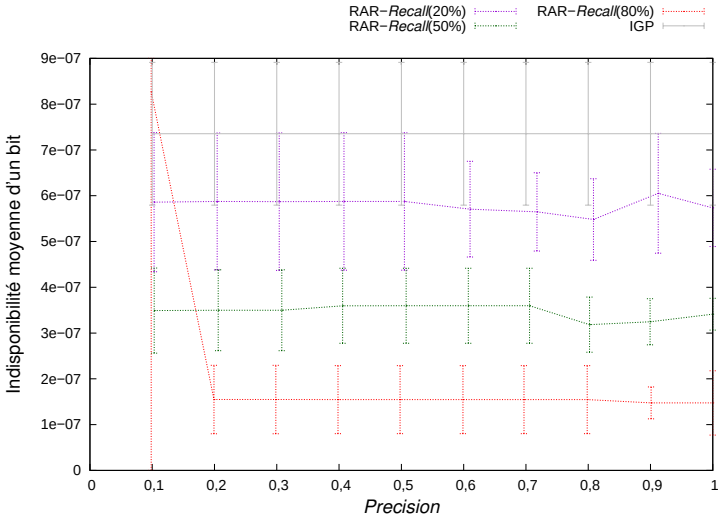


(c) EU

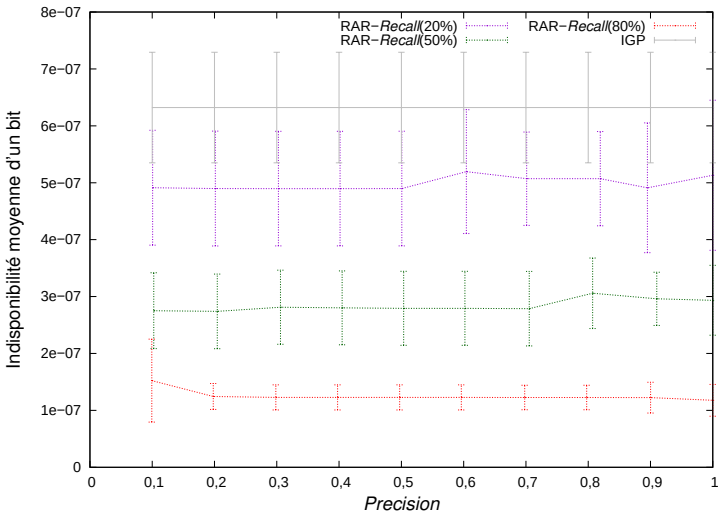
FIGURE C.18: Impact du *Recall* sur la disponibilité.



(a) A

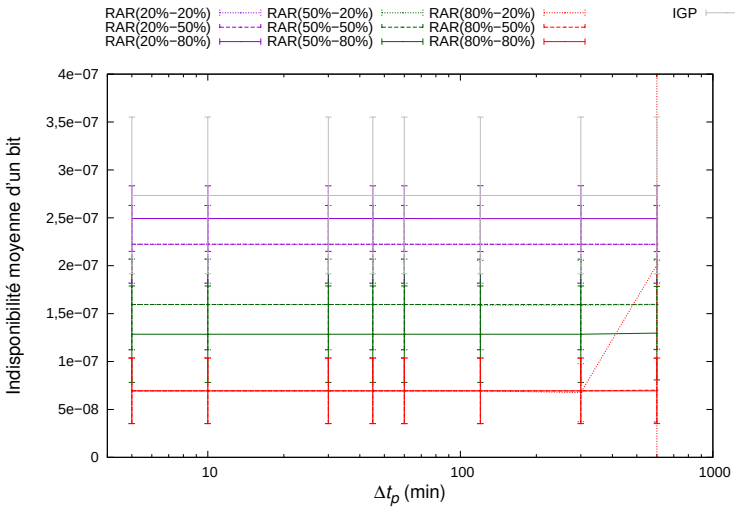


(b) US

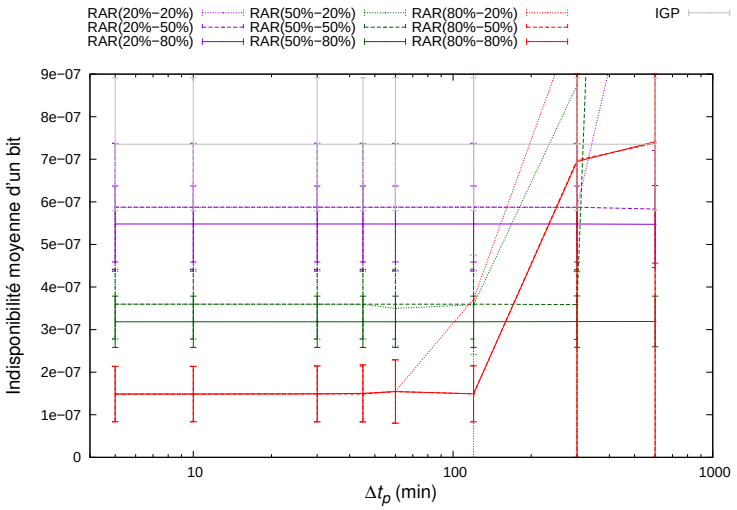


(c) EU

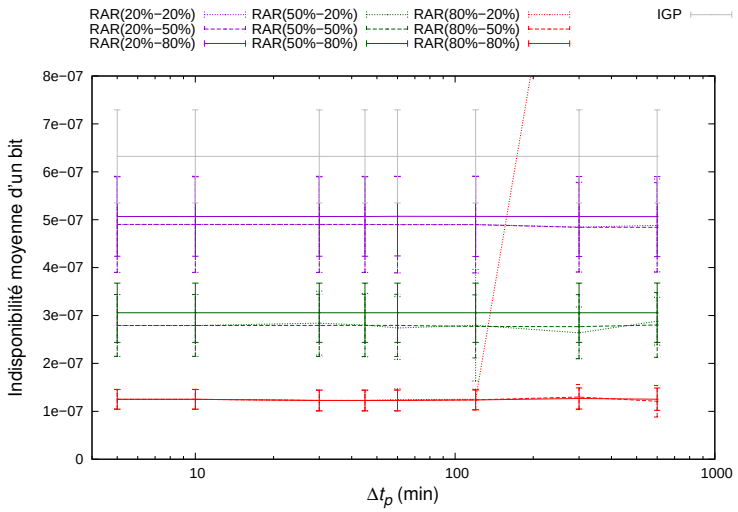
FIGURE C.19: Impact de la *Precision* sur la disponibilité.



(a) A



(b) US

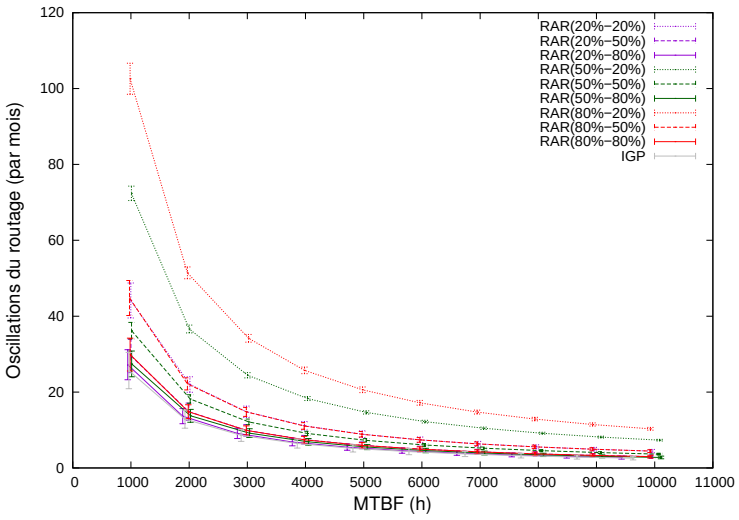


(c) EU

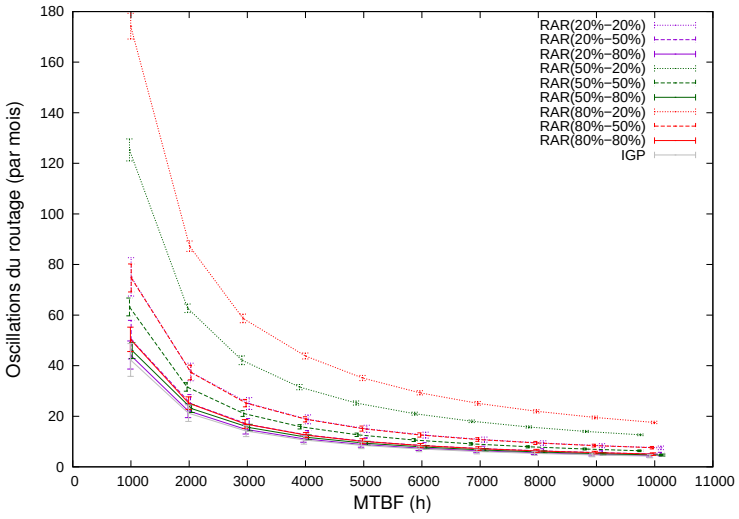
Figure C.20: Impact de Δt_p sur la disponibilité.

C.2.2 Étude de la stabilité du routage

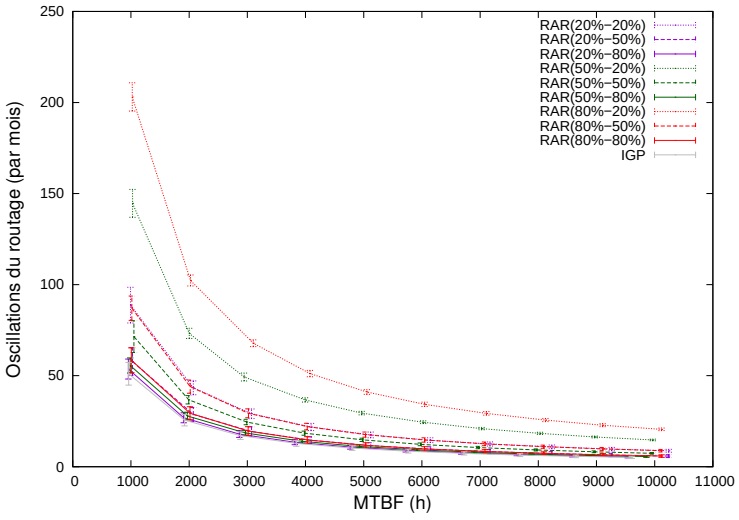
C.2.2.1 Influence de la probabilité de panne



(a) A

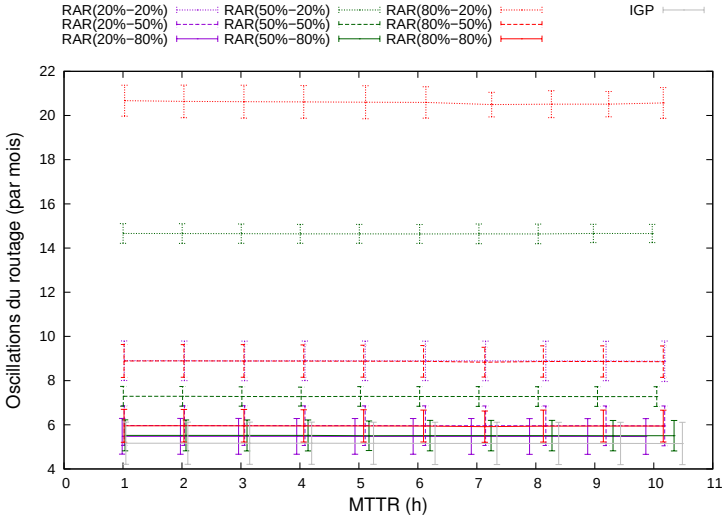


(b) US

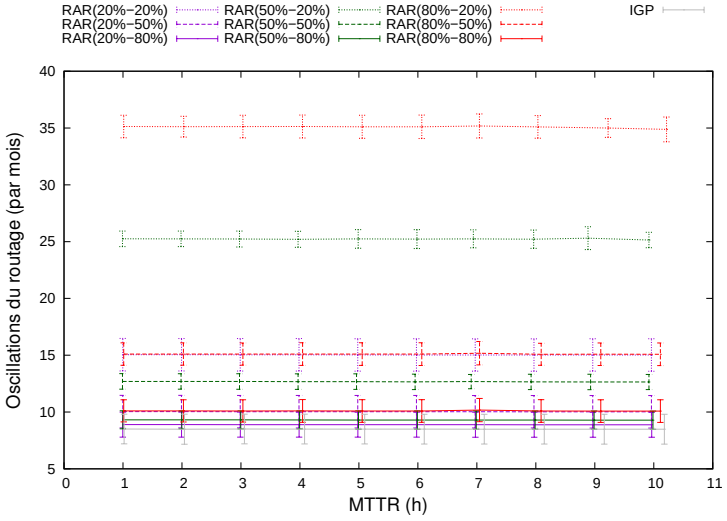


(c) EU

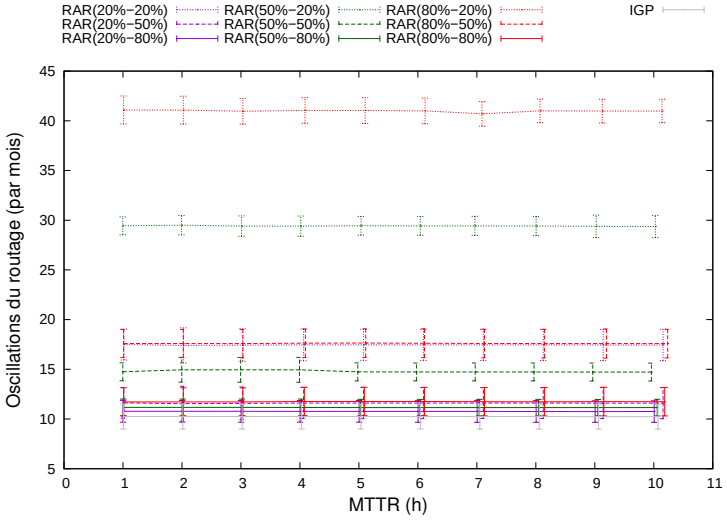
FIGURE C.21: Impact du MTBF sur le nombre d'oscillations du routage.



(a) A



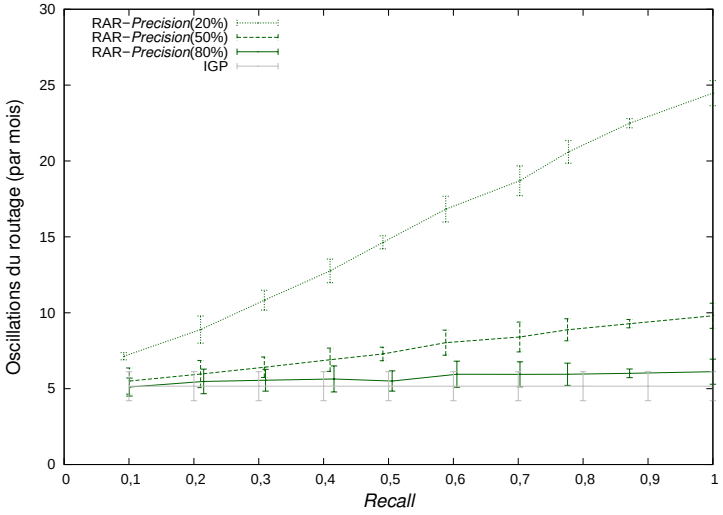
(b) US



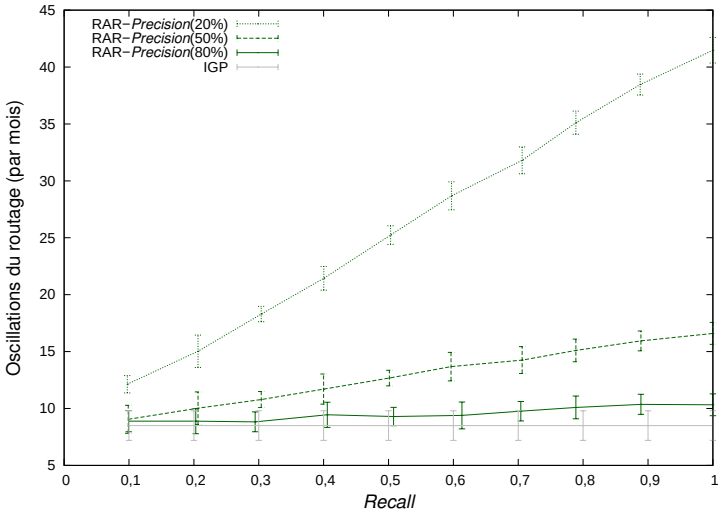
(c) EU

FIGURE C.22: Impact du MTTR sur le nombre d'oscillations du routage.

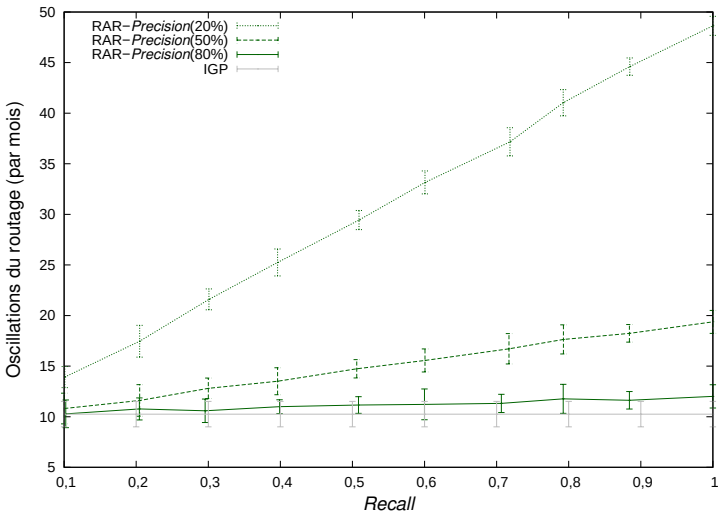
C.2.2.2 Les conséquences de la prédiction de pannes



(a) A

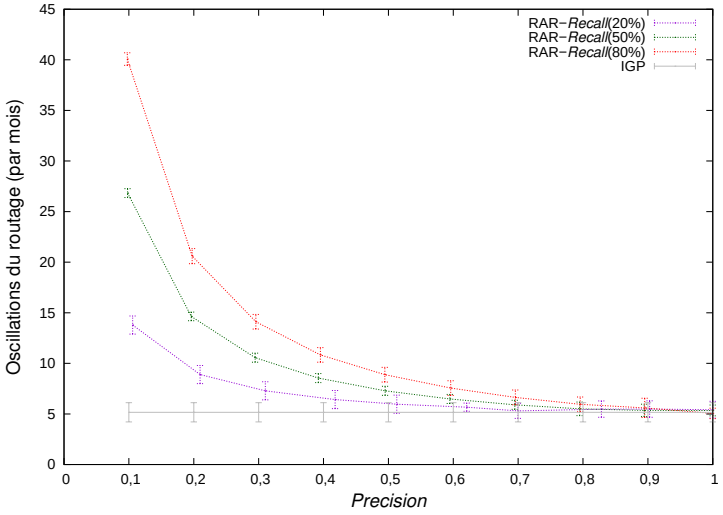


(b) US

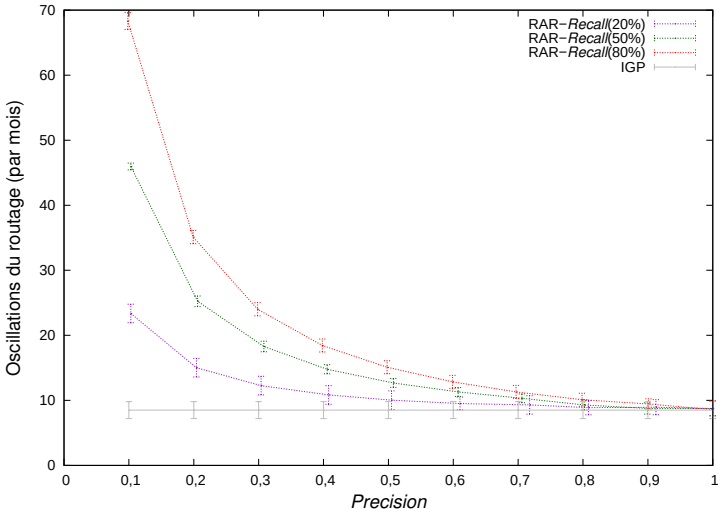


(c) EU

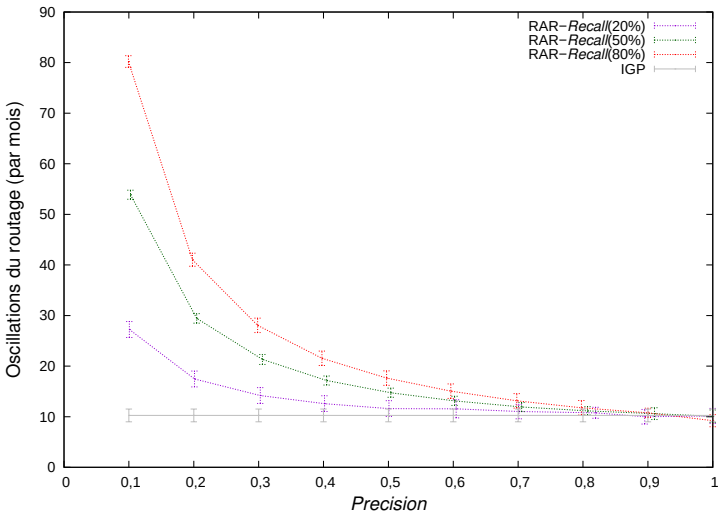
Figure C.23: Impact du *Recall* sur le nombre d'oscillations du routage.



(a) A

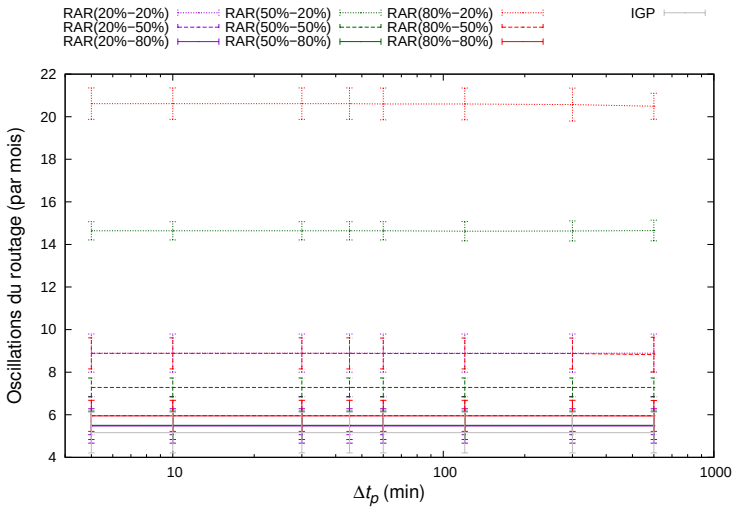


(b) US

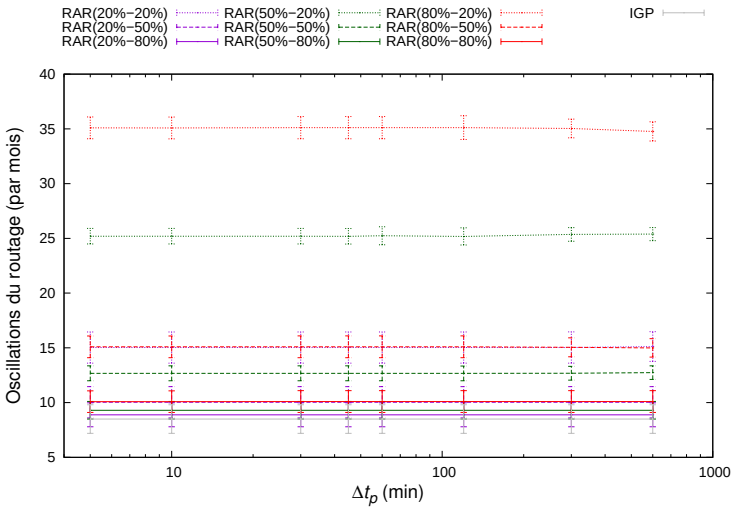


(c) EU

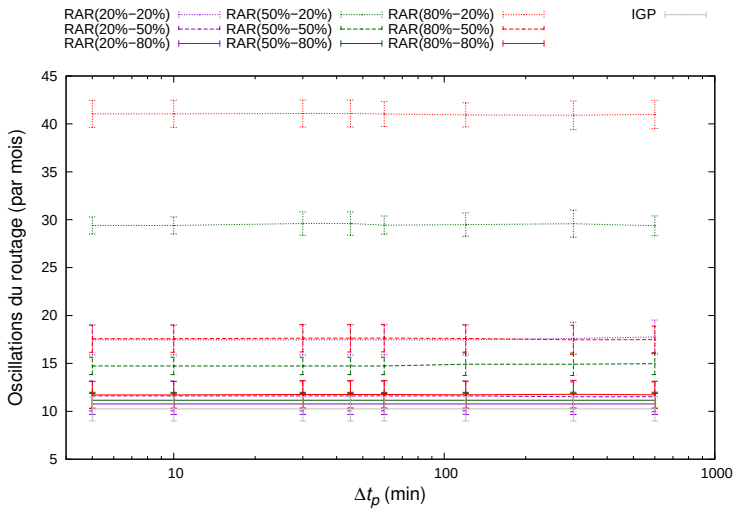
Figure C.24: Impact de la *Precision* sur le nombre d'oscillations du routage.



(a) A



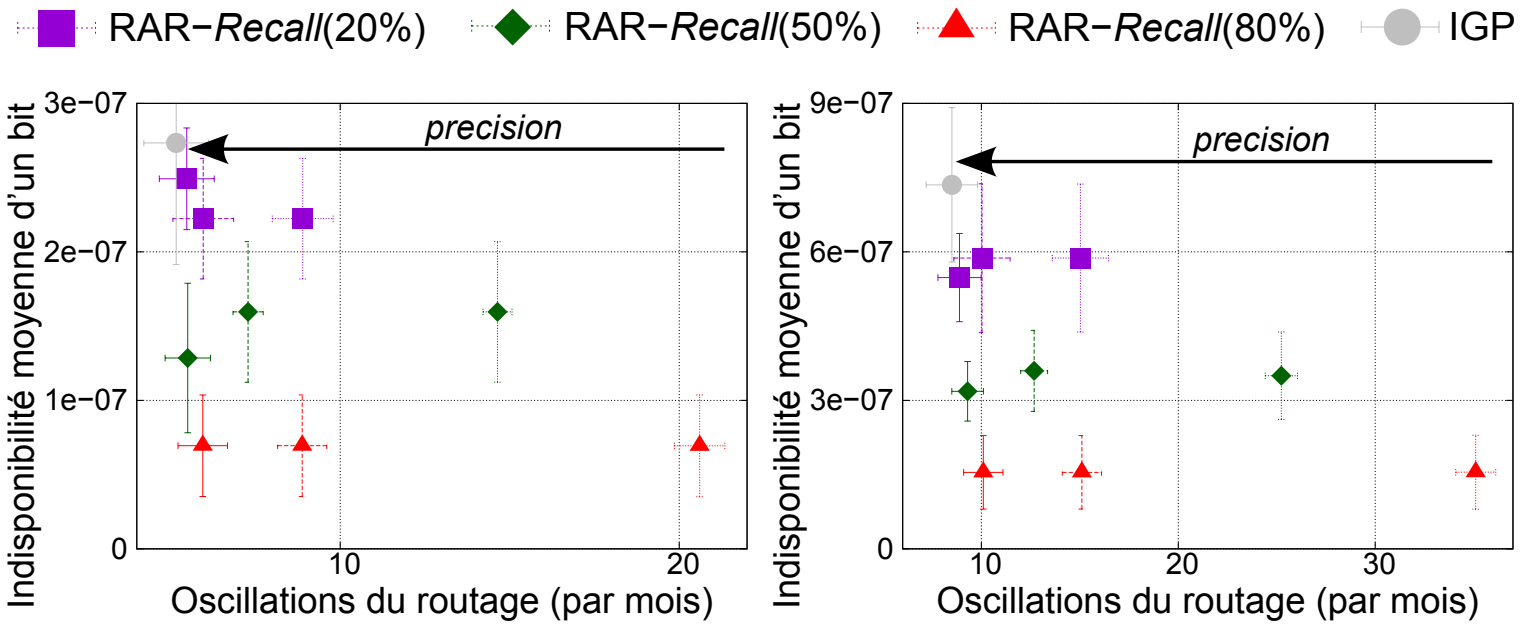
(b) US



(c) EU

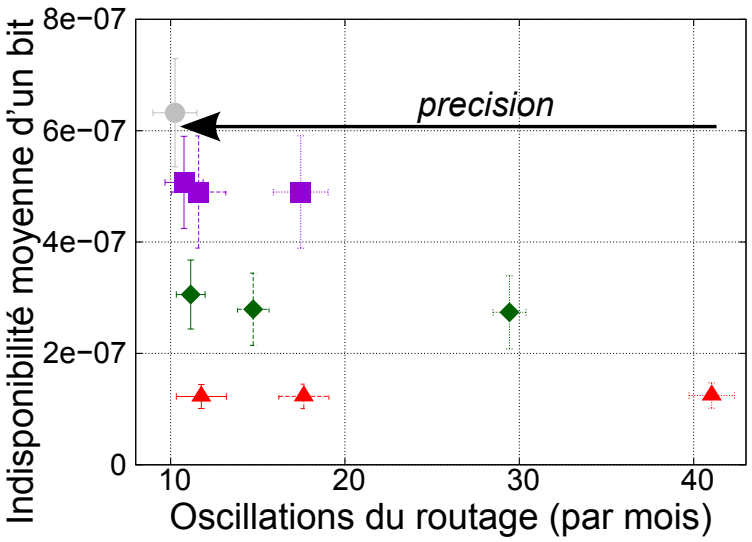
FIGURE C.25: Impact de Δt_p sur le nombre d'oscillations du routage.

C.2.3 Étude conjointe de la disponibilité et de la stabilité du routage



(a) A

(b) US



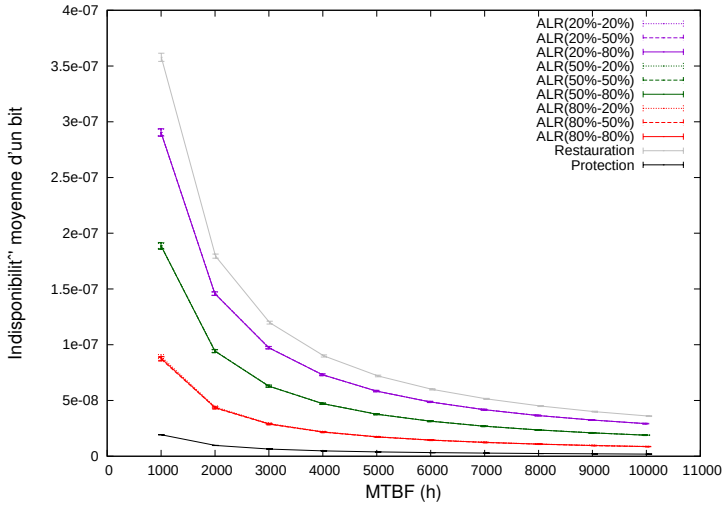
(c) EU

Figure C.26: Ratio indisponibilité / nombre d'oscillation du routage.

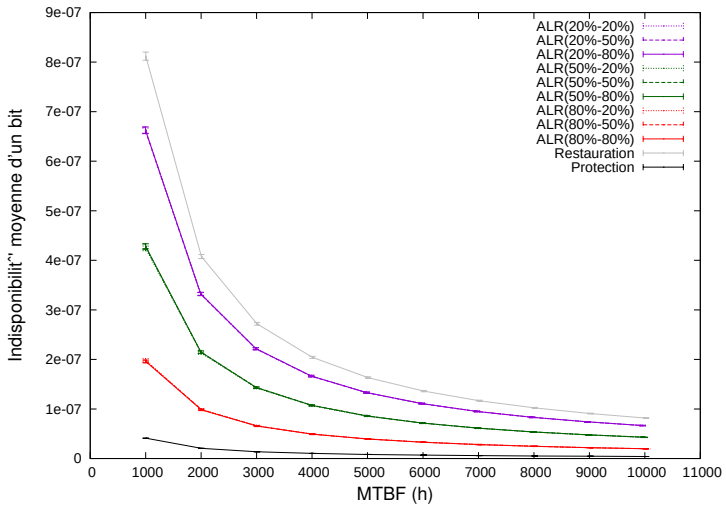
C.3 Mécanisme de résilience adaptatif (ALR)

C.3.1 Étude de la disponibilité

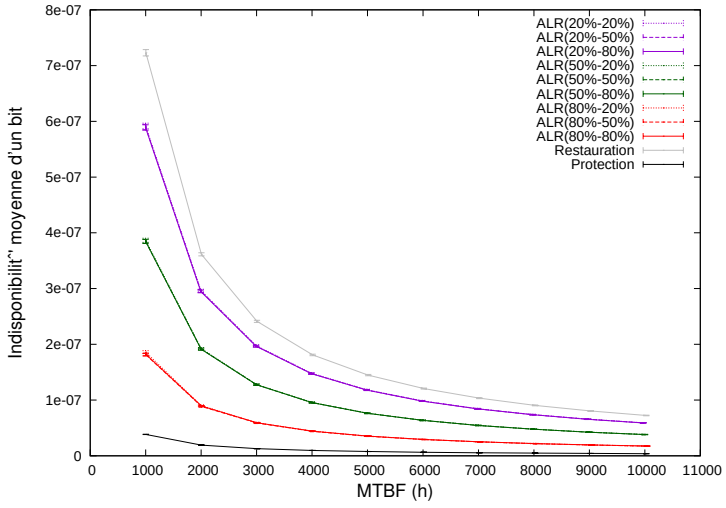
C.3.1.1 Influence de la probabilité de panne



(a) A

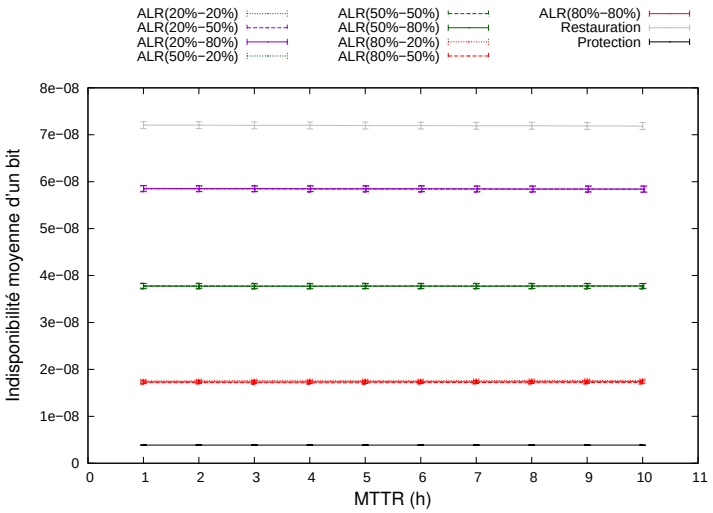


(b) US

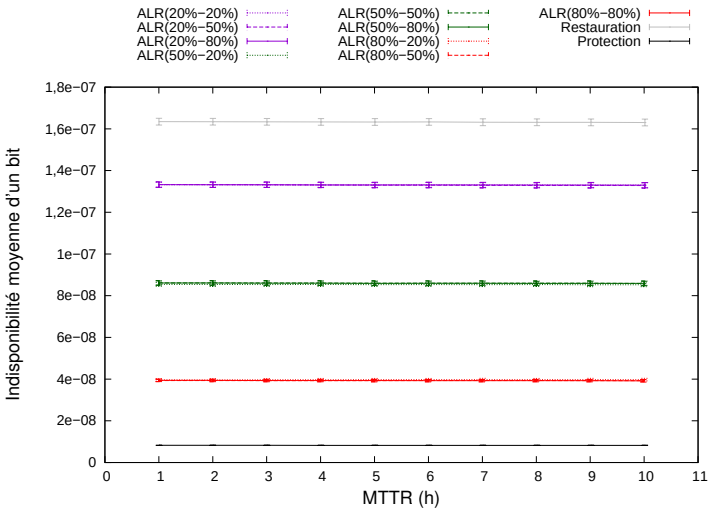


(c) EU

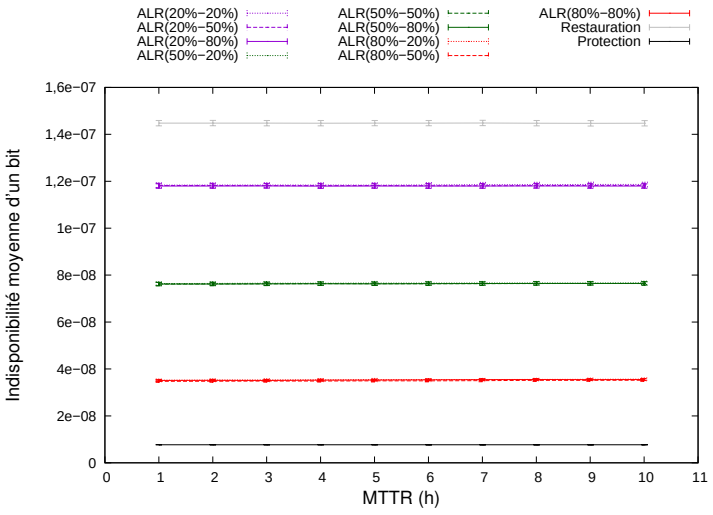
FIGURE C.27: Impact du MTBF sur la disponibilité.



(a) A



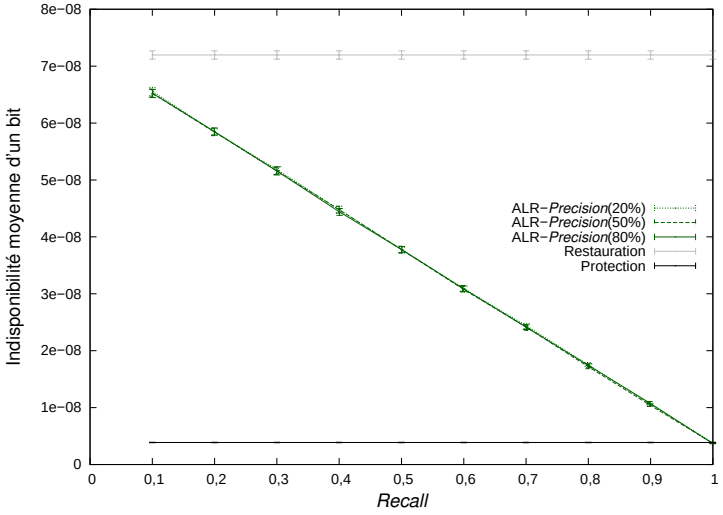
(b) US



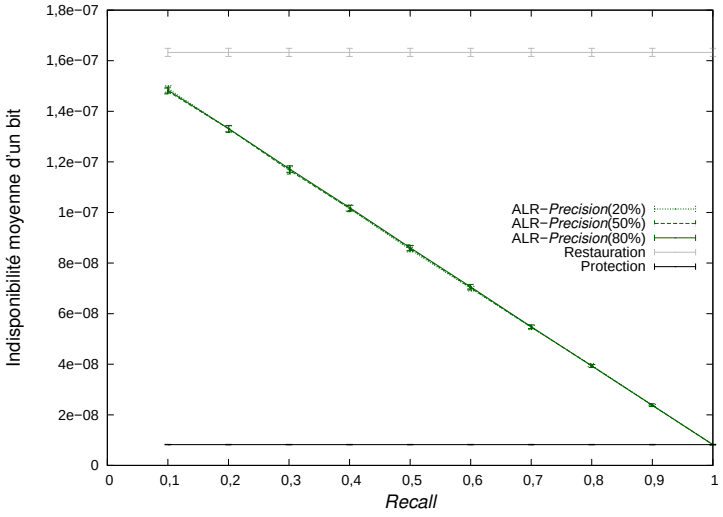
(c) EU

FIGURE C.28: Impact du MTTR sur la disponibilité.

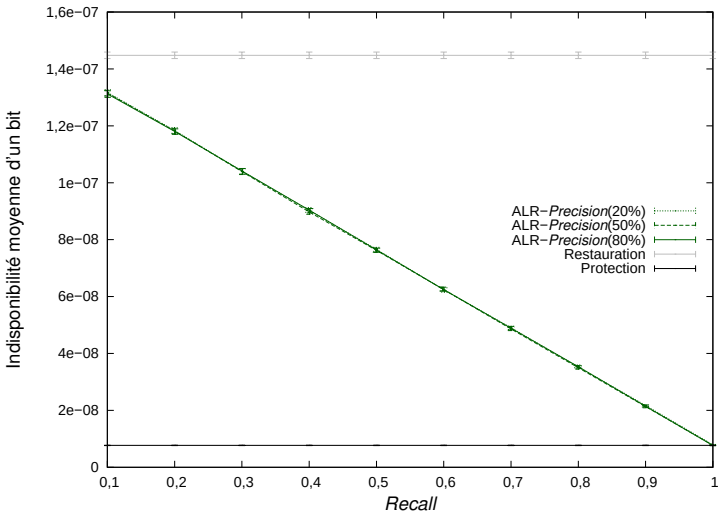
C.3.1.2 Les conséquences de la prédiction de pannes



(a) A

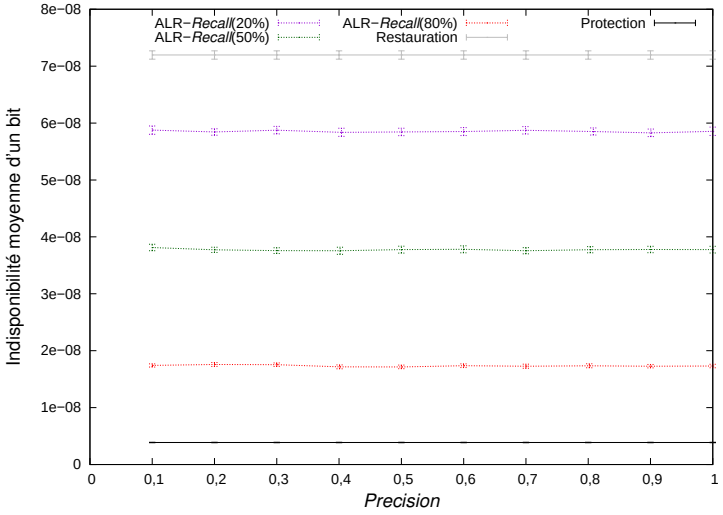


(b) US

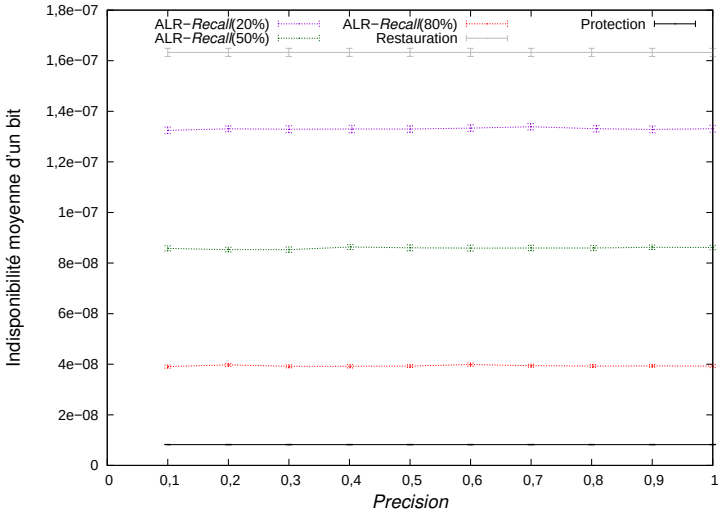


(c) EU

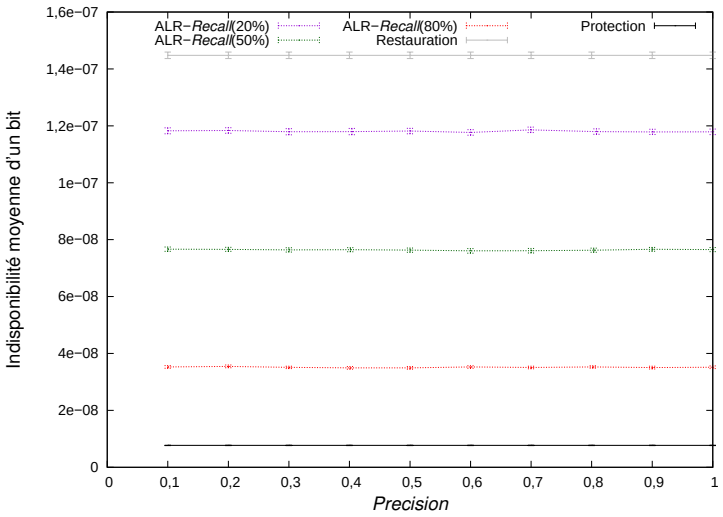
Figure C.29: Impact du *Recall* sur la disponibilité.



(a) A

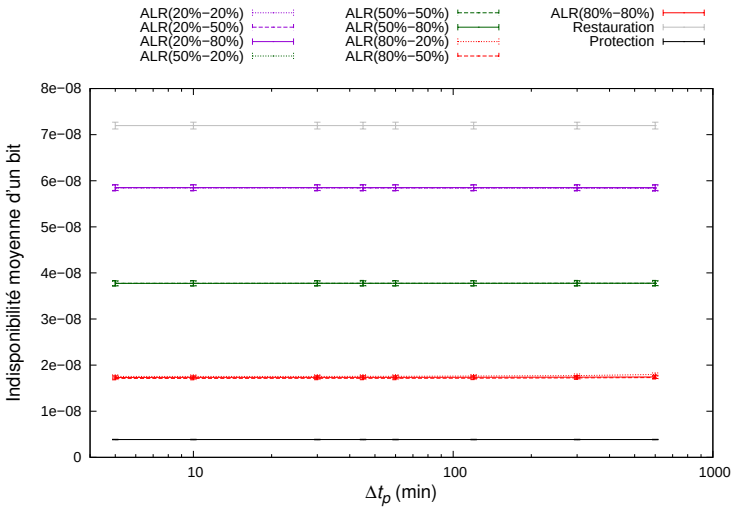


(b) US

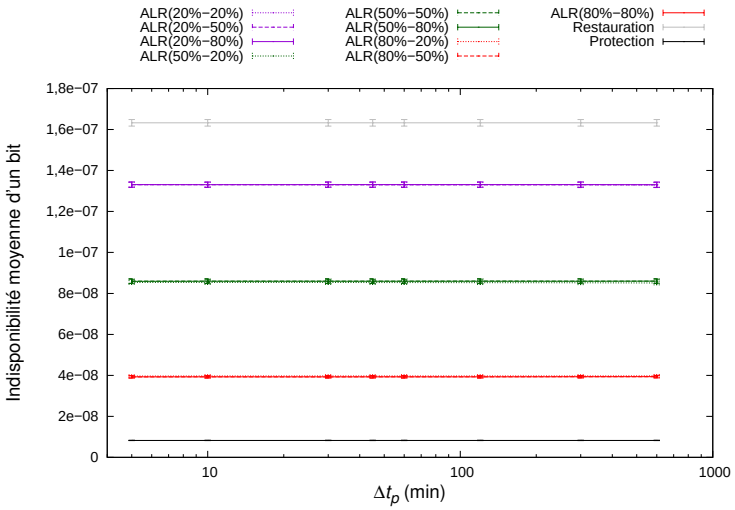


(c) EU

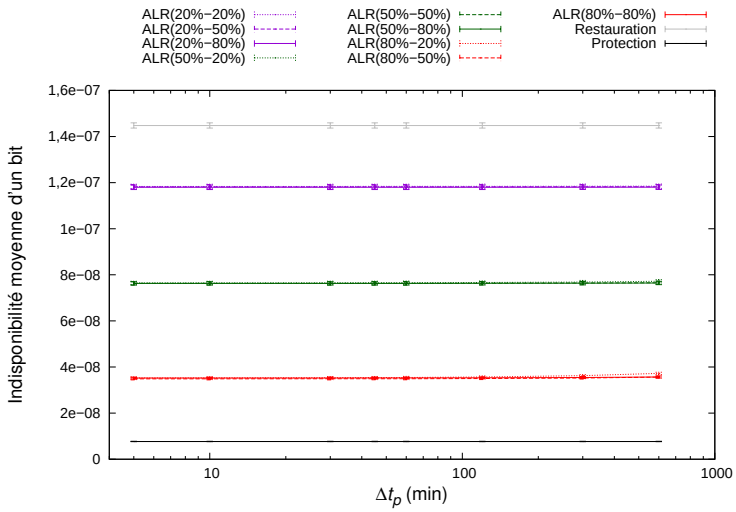
Figure C.30: Impact de la *Precision* sur la disponibilité.



(a) A



(b) US

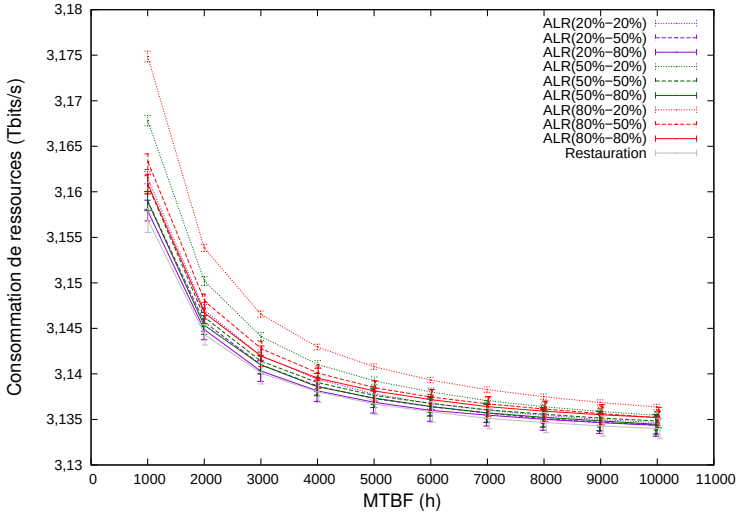


(c) EU

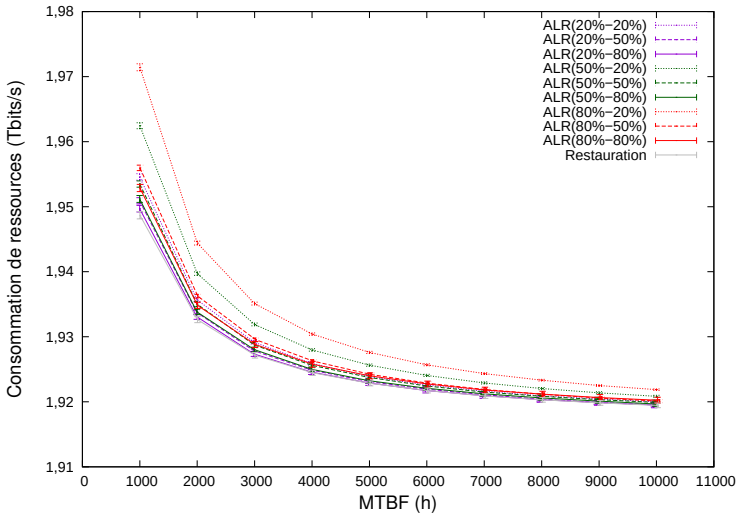
FIGURE C.31: Impact de Δt_p sur la disponibilité.

C.3.2 Étude de la consommation de ressources

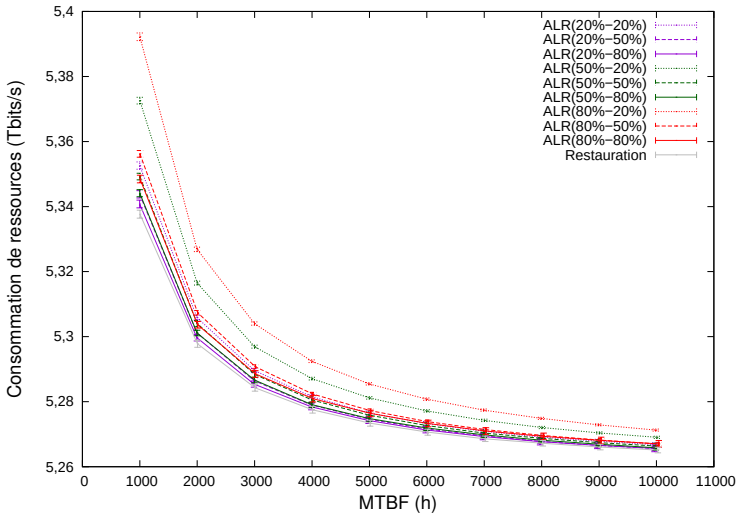
C.3.2.1 Influence de la probabilité de panne



(a) A

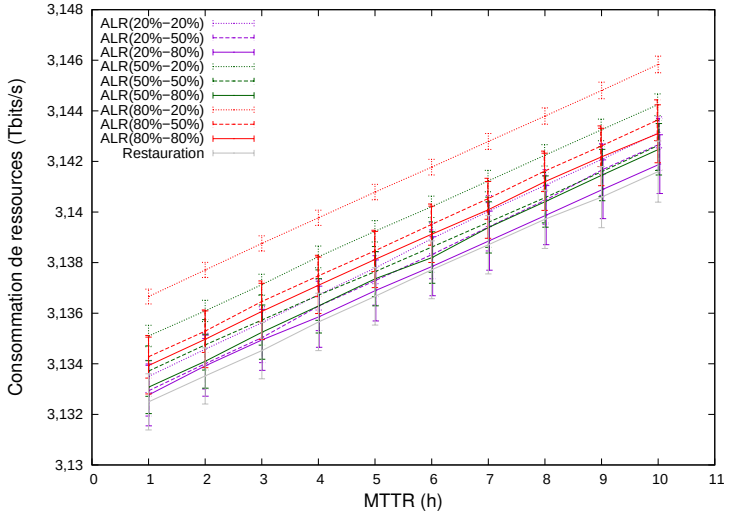


(b) US

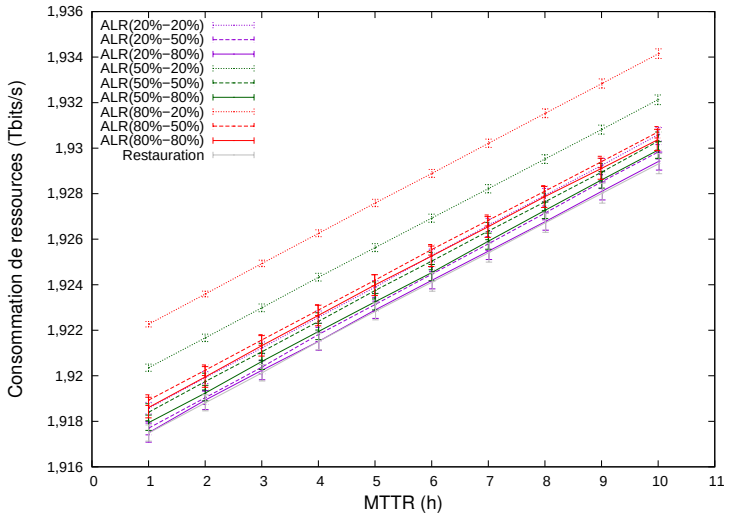


(c) EU

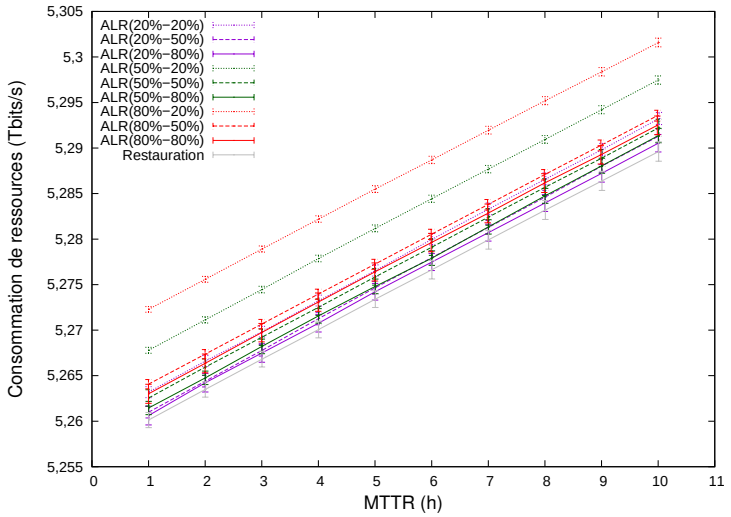
FIGURE C.32: Impact du MTBF sur l'utilisation des ressources.



(a) A



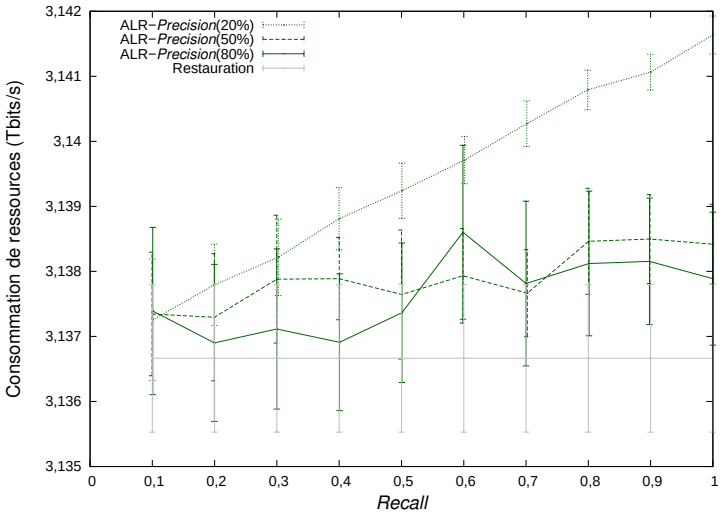
(b) US



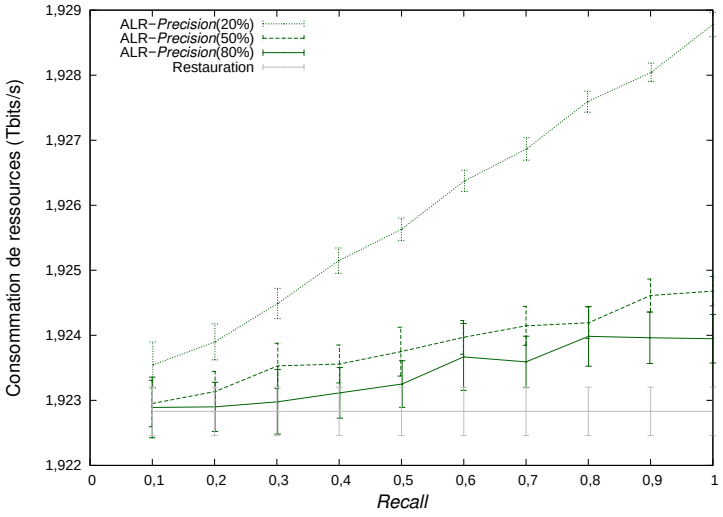
(c) EU

FIGURE C.33: Impact du MTTR sur l'utilisation des ressources.

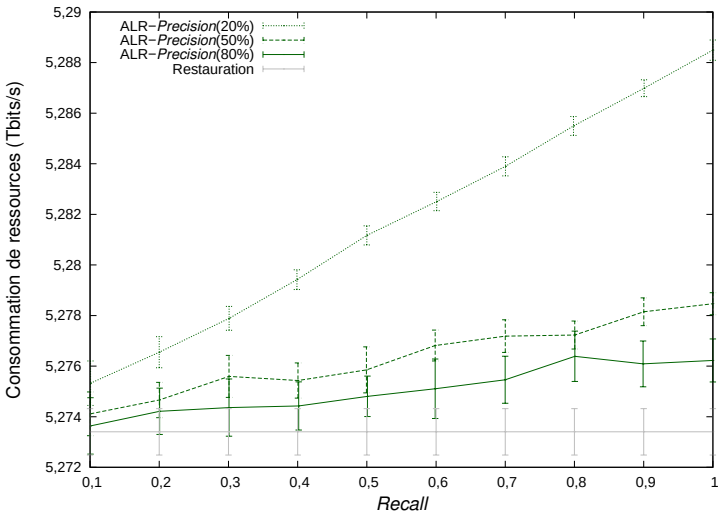
C.3.2.2 Les conséquences de la prédiction de pannes



(a) A

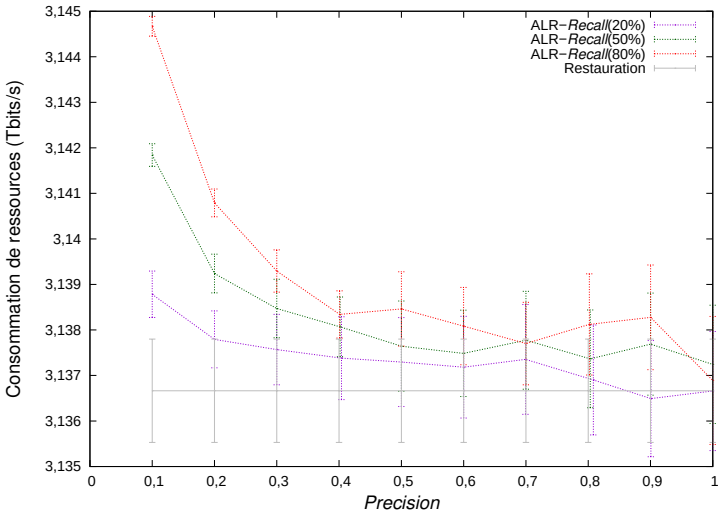


(b) US

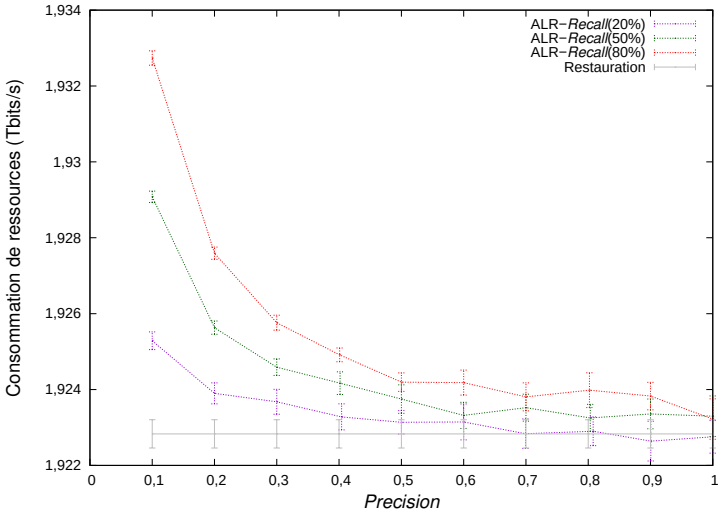


(c) EU

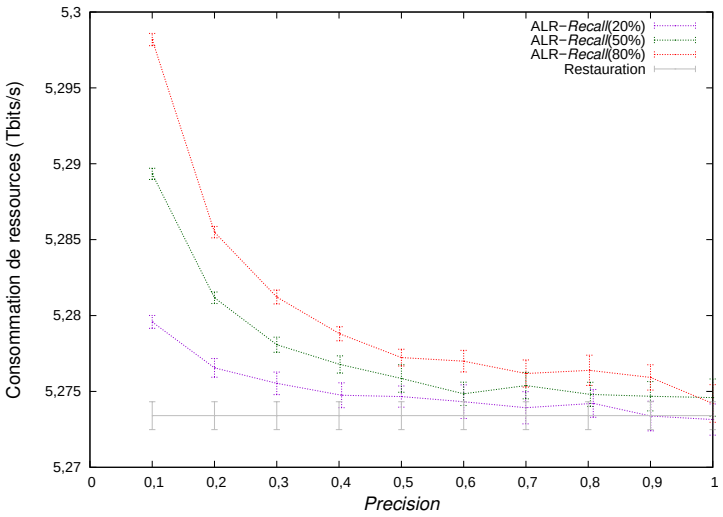
FIGURE C.34: Impact du *Recall* sur l'utilisation des ressources.



(a) A

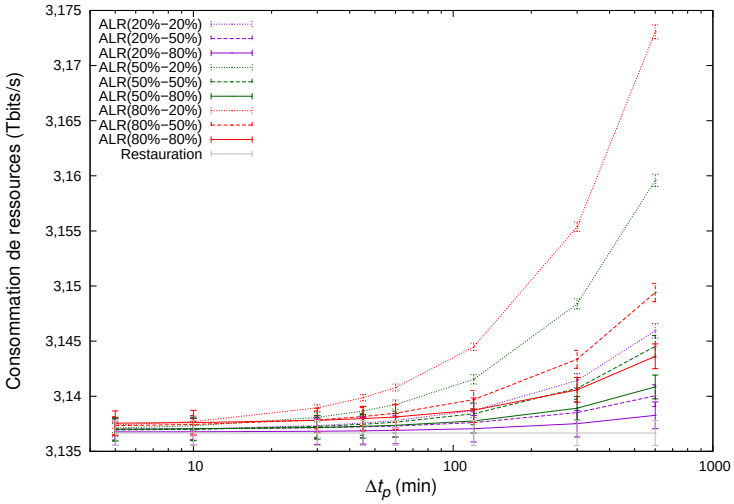


(b) US

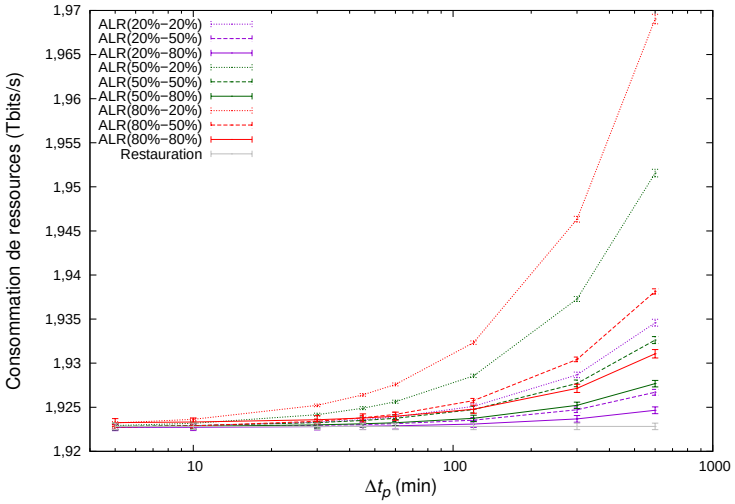


(c) EU

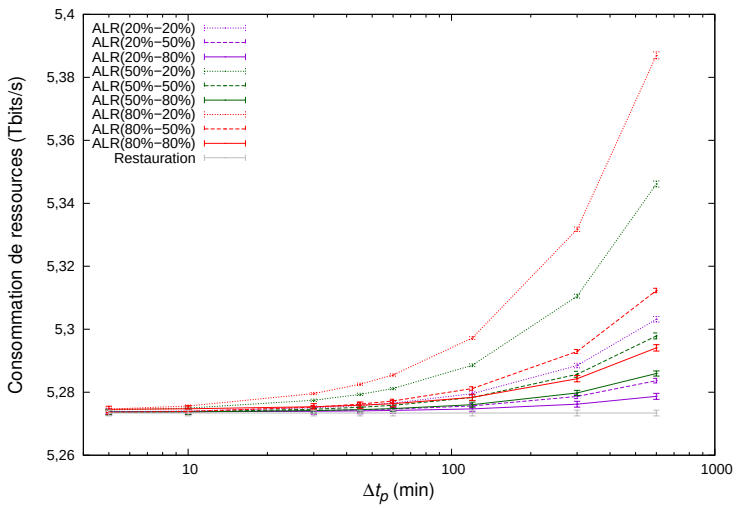
FIGURE C.35: Impact de la *Precision* sur l'utilisation des ressources.



(a) A



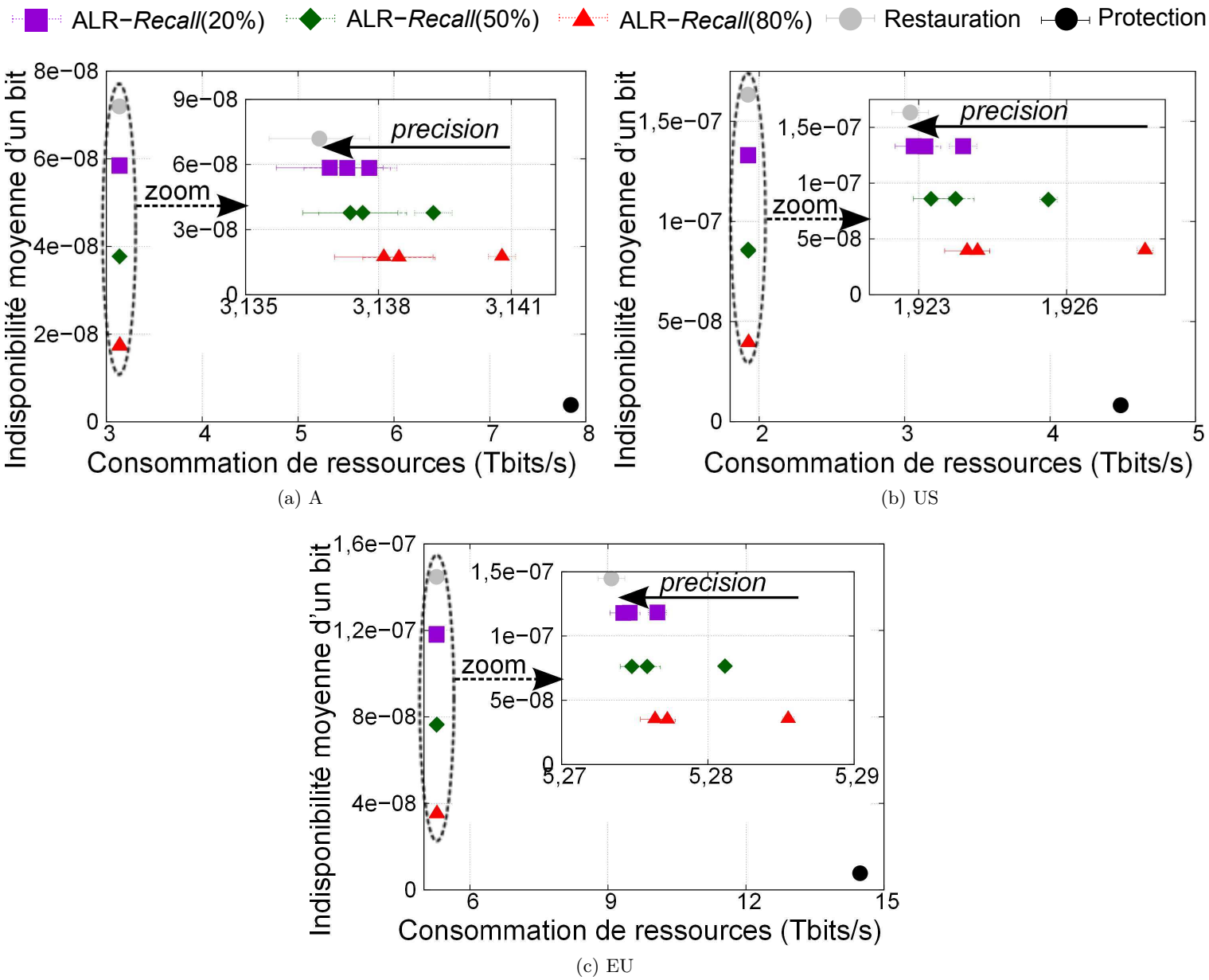
(b) US



(c) EU

Figure C.36: Impact de Δt_p sur l'utilisation des ressources.

C.3.3 Étude conjointe de la disponibilité et de la consommation de ressources



(a) A

(b) US

(c) EU

FIGURE C.37: Ratio indisponibilité / utilisation des ressources.

Annexe D

Démonstrateur

Une interface graphique visible sur la Fig. D.1 a été implémentée afin de contrôler et visualiser l'état d'un routeur du prototype implémentant le mécanisme de routage sensible au risque de pannes [KAB+11].

Cette interface dispose des fonctionnalités suivantes :

- activation de la fonction RAR par l'intermédiaire du bouton « *start risk-aware* » ;
- désactivation de la fonction RAR par l'intermédiaire du bouton « *stop risk-aware* » ;
- déclenchement d'une augmentation de température aboutissant à une panne avec par l'intermédiaire du bouton « *start overheating* », qui permet également de stopper cette augmentation ;
- déclenchement d'une baisse de température jusqu'à la température stationnaire par l'intermédiaire du bouton « *stop overheating* », qui permet également de stopper cette baisse ;
- affectation d'une température spécifique en utilisant la jauge située sous le thermomètre.

De plus les informations suivantes sont affichées (voir Fig. D.1) :

- état de la fonctionnalité de routage sensible au risque de pannes(*i.e.* activée ou non) ;
- risque de panne du routeur (*i.e.* détecté ou non) ;
- état du routeur (*i.e.* en fonctionnement ou non) ;
- température du routeur en temps réel ;
- historique de la température du routeur pendant les deux dernières minutes.

La Fig. D.1 affiche l'interface de contrôle d'un routeur lorsque le mécanisme RAR n'est pas activé et la Fig. D.2 affiche l'interface de contrôle d'un routeur lorsque celui-ci est en panne et que le mécanisme RAR n'est pas activé.

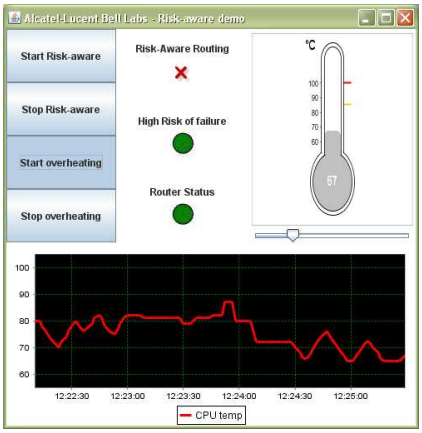


FIGURE D.1: Interface de contrôle du démonstrateur sans RAR.

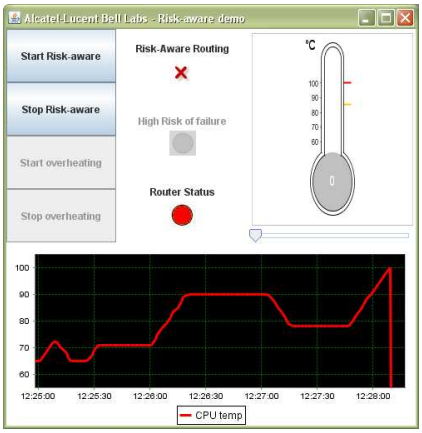


FIGURE D.2: Interface de contrôle du démonstrateur sans RAR lors d'une panne.

La Fig. D.3 affiche l'interface de contrôle d'un routeur lorsque celui-ci redémarre et que le mécanisme RAR n'est pas activé et la Fig. D.4 affiche l'interface de contrôle d'un routeur lorsque le mécanisme RAR n'est pas activé.

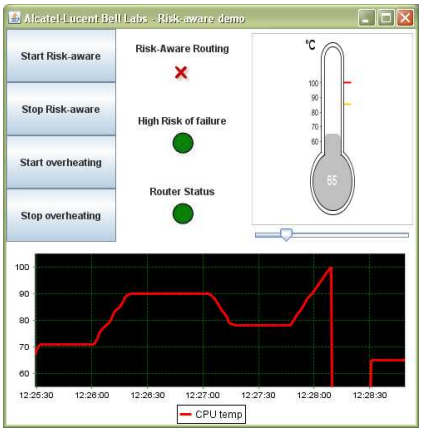


FIGURE D.3: Interface de contrôle du démonstrateur sans RAR après une panne.

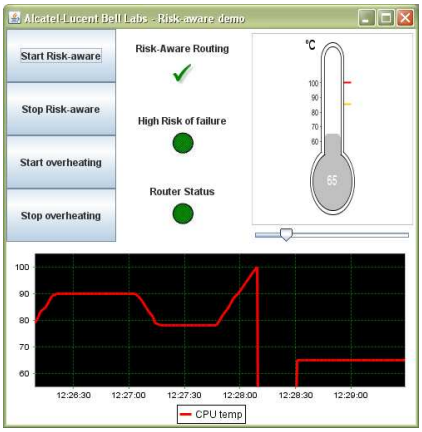


FIGURE D.4: Interface de contrôle du démonstrateur avec RAR.

La Fig. D.5 affiche l'interface de contrôle d'un routeur lorsque celui-ci utilise le mécanisme RAR et qu'un risque de panne est détecté et la Fig. D.6 affiche la gestion du risque de panne liée à la température en utilisant une hystérésis.

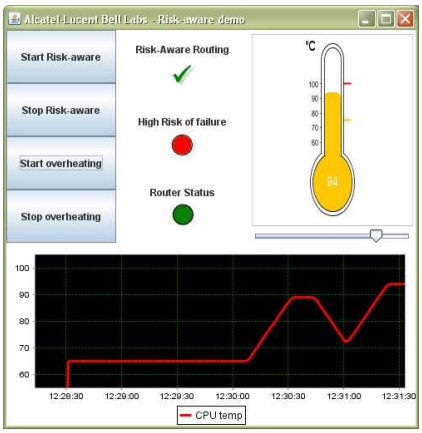


FIGURE D.5: Interface de contrôle du démonstrateur avec RAR lors de la détection d'un risque de panne.

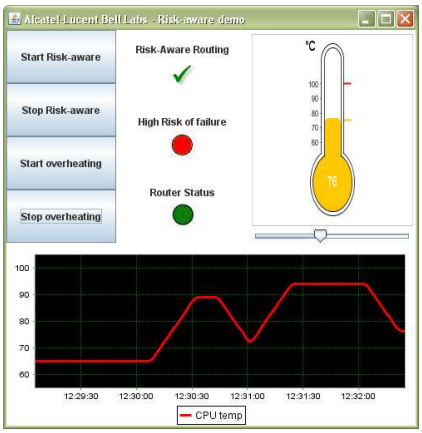


FIGURE D.6: Interface de contrôle du démonstrateur avec RAR illustrant la gestion de la température par Hystérésis.

La Fig. D.7 affiche l'interface de contrôle d'un routeur lorsque celui-ci est en panne et utilise le mécanisme RAR et la Fig. D.8 affiche l'interface de contrôle d'un routeur lorsque celui-ci redémarre et utilise le mécanisme RAR.

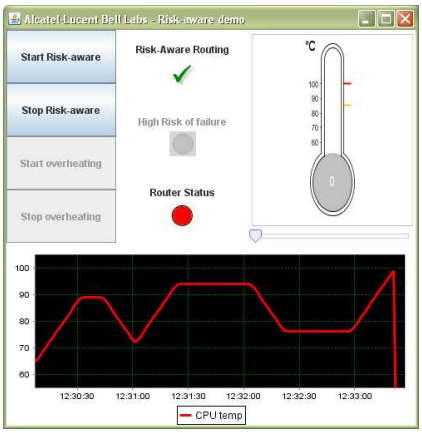


FIGURE D.7: Interface de contrôle du démonstrateur avec RAR lors d'une panne.

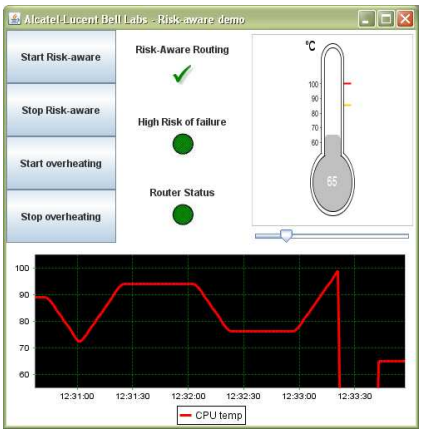


FIGURE D.8: Interface de contrôle du démonstrateur avec RAR après une panne.

Table des figures

Liste des tableaux

Bibliographie

[ABG+01] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan et G. Swallow : RSVP-TE : Extensions to RSVP for LSP Tunnels. RFC 3209 (Proposed Standard), décembre 2001. Updated by RFCs 3936, 4420, 4874, 5151, 5420, 5711.

[AC02] Cengiz Alaettinoglu et Stephen Casner : Detailed analysis of ISIS routing protocol on the qwest backbone, 2002.

[ADF+01] L. Andersson, P. Doolan, N. Feldman, A. Fredette et B. Thomas : LDP Specification. RFC 3036 (Proposed Standard), janvier 2001. Obsoleted by RFC 5036.

[AG05] A. Abraham et C. Grosan : Genetic programming approach for fault modeling of electronic hardware. *In Proceedings of the IEEE Congress on Evolutionary Computation*, volume 2, pages 1563–1569, septembre 2005.

[ALAS+02] J. Ash, Y. Lee, P. Ashwood-Smith, B. Jamoussi, D. Fedyk, D. Skalecki et L. Li : LSP Modification Using CR-LDP. RFC 3214 (Proposed Standard), janvier 2002.

[AS07] A. Andrzejak et L. Silva : Deterministic models of software aging and optimal rejuvenation schedules. *In Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Network Management, IM*, pages 159–168, 2007.

[Atl06] Alia Atlas : U-turn alternates for IP/LDP fast-reroute. http ://tools.ietf.org/html/draft-atlas-ip-local-protect-uturn-03, 2006.

[AYJ00] C. Alaettinoglu, H. Yu et V. Jacobson : Towards milli-second IGP convergence. *NANOG*, 2000.

[AZ08] A. Atlas et A. Zinin : Basic Specification for IP Fast Reroute : Loop-Free Alternates. RFC 5286 (Proposed Standard), septembre 2008.

[BAV03] H. R Berenji, J. Ametha et D. Vengerov : Inductive learning for fault diagnosis. *In Proceedings of the 12th IEEE International Conference on Fuzzy Systems, FUZZ*, volume 1, pages 726–731, mai 2003.

[BDFM+09] Luciano Baresi, Antonio Di Ferdin, Antonio Manzalini, Luciano Baresi, Antonio Di Ferdinando, Antonio Manzalini et Franco Zambonelli : The CASCADAS framework for autonomic communications. *Autonomic Communication*, 2009.

[BF07] Hitesh Ballani et Paul Francis : CONMan : a step towards network manageability. *In ACM SIGCOMM Computer Communication Review*, volume 37, pages 205–216, Kyoto, Japan, 2007.

[BF09] H. Ballani et P. Francis : Fault management using the CONMan abstraction. *In Proceedings of the 28th IEEE Conference on Computer Communications, INFOCOM*, pages 765–773, Rio De Janeiro, Brazil, avril 2009.

[BFB+05] P. Bodic, G. Friedman, L. Biewald, H. Levine, G. Candea, K. Patel, G. Tolle, J. Hui, A. Fox, M.I. Jordan et D. Patterson : Combining visualization and statistical analysis to improve operator confidence and efficiency for failure detection and localization. *In Proceedings of the Second International Conference on Autonomic Computing, ICAC*, pages 89–100, Seattle, WA, USA, 2005.

[BFPS07] Stewart Bryant, Clarence Filsfils, Stefano Previdi et Mike Shand : IP fast reroute using tunnels. http ://tools.ietf.org/html/draft-bryant-ipfrr-tunnels-03, 2007.

[BJT+10] Ghazi Bouabene, Christophe Jelger, Christian Tschudin, Stefan Schmid, Ariane Keller et Martin May : The autonomic network architecture (ANA). *IEEE Journal on Selected Areas in Communications*, 28(1):4–14, janvier 2010.

[BKH+08] Thomas Bullot, Rida Khatoun, Louis Hugues, Dominique Gaiti et Leila Merghem-Boulahia : A situatedness-based knowledge plane for autonomic networking. *International Journal of Network Management*, 18(2):171–193, mars 2008.

[BR01] Anindya Basu et Jon Riecke : Stability issues in OSPF routing. *In ACM SIGCOMM Computer Communication Review*, volume 31, page 225–236, New York, NY, USA, août 2001.

[BX02] Raouf Boutaba et Jin Xiao : Network management : State of the art. *In Proceedings of the 17th IFIP World Computer Congress*, page 127–146, 2002.

[BZG+06] Thomas Bullot, Hubert Zimmermann, Dominique Gaiti, Leila Merghem-Boulahia et Guy Pujolle : Autonomous agents for autonomic networks. *Annales des télécommunications*, pages 1017–1045, 2006.

[CAK+04] Mike Y. Chen, Anthony Accardi, Emre Kiciman, David A. Patterson, Armando Fox et Eric A. Brewer : Path-based failure and evolution management. *In Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation, NSDI*, pages 309–322, 2004.

[Cas00] R.P. Case : The commercial development of novel proactive sentient systems. *In Proceedings of the IEEE Engineering Management Society*, pages 523–527, 2000.

[CCW+09] Ranganai Chaparadza, Laurent Ciavaglia, Michał Wódczak, Chin-Chou Chen, Brian A. Lee, Athanassios Liakopoulos, Anastasios Zafeiropoulos, Estelle Mancini, Ultan Mulligan, Alan Davy, Kevin Quinn, Benoit Radier, Nancy Alonistioti, Apostolos Kousaridas, Panagiotis Demestichas, Kostas Tsagkaris, Martin Vigoureux, Laurent Vreck, Mick Wilson et Latif Ladid : ETSI industry specification group on autonomic network engineering for the self-managing future internet (ETSI ISG AFI). *In Proceedings of the 10th International Conference on Web Information Systems Engineering, WISE*, Lecture Notes in Computer Science, page 61–62, Poznan, Poland, 2009.

[CFSD90] J.D. Case, M. Fedor, M.L. Schoffstall et J. Davin : Simple Network Management Protocol (SNMP). RFC 1157 (Historic), mai 1990.

[CGM02] Karen J Cassidy, Kenny C Gross et Amir Malekpour : Advanced pattern recognition for detection of complex software aging phenomena in online transaction processing servers. *In Proceedings of the International Conference on Dependable Systems and Networks, DSN*, page 478–482, 2002.

[Cha08] Ranganai Chaparadza : Requirements for a generic autonomic network architecture (GANA), suitable for standardizable autonomic behavior specifications for diverse networking environments. *Annual Review of Communications*, 61, 2008.

[Cha09] Ranganai Chaparadza : UniFAFF : a unified framework for implementing autonomic fault management and failure detection for self-managing networks. *International Journal of Network Management*, 19(4):271–290, juillet 2009.

[CHH+01] V. Castelli, R. E Harper, P. Heidelberger, S. W Hunter, K. S Trivedi, K. Vaidyanathan et W. P Zeggert : Proactive management of software aging. *IBM Journal of Research and Development*, 45(2):311–332, mars 2001.

[CHK+08] T. C Cicic, A. F Hansen, A. Kvalbein, M. Hartman, R. Martin et M. Menth : Relaxed multiple routing configurations for IP fast reroute. *In Proceedings of the IEEE Network Operations and Management Symposium, NOMS*, pages 457–464, avril 2008.

[Cho99] Kathleen Cholewka : MCI outage has domino effect. http ://connection.ebscohost.com/c/articles/2226716/mci-outage-has-domino-effect, 1999.

[CKF+02] M. Y Chen, E. Kiciman, E. Fratkin, A. Fox et E. Brewer : Pinpoint : problem determination in large, dynamic internet services. *In Proceedings of the International Conference on Dependable Systems and Networks, DSN*, pages 595– 604, 2002.

[CL09] R. Cohen et A. Landau : "Not all at once !" - a generic scheme for estimating the number of affected nodes while avoiding feedback implosion. *In Proceeding of the 28th Conference on Computer Communications, INFOCOM*, pages 2641–2645, Rio De Janeiro, Brazil, avril 2009.

[CLSS02] G. Conte, M. Listanti, M. Settembre et R. Sabella : Strategy for protection and restoration of optical paths in WDM backbone networks for next-generation internet infrastructures. *Journal of Lightwave Technology*, 20(8):1264–1276, 2002.

[CMPS02] J. Case, R. Mundy, D. Partain et B. Stewart : Introduction and Applicability Statements for Internet-Standard Management Framework. RFC 3410 (Informational), décembre 2002.

[CMRW93] J. Case, K. McCloghrie, M. Rose et S. Waldbusser : Introduction to version 2 of the Internet-standard Network Management Framework. RFC 1441 (Historic), avril 1993.

[CMS01] G.L. Choudhury, A.S. Maunder et V.D. Sapozhnikova : Faster link-state IGP convergence and improved network scalability and stability. *In Proceedings of the 26th Annual IEEE Conference on Local Computer Networks, LCN*, pages 149–158, 2001.

[CPA+10] Ranganai Chaparadza, Symeon Papavassiliou, Giorgos Aristomenopoulos, Timotheos Kastrinogiannis, Mary Grammatikou, Christos Argyropoulos, Zhaojun Li, Michal Wodczak, Mariusz Fecko, Yuhong Li, Wang Wendong, Nikolay Tcholtchev, Arun Prakash, Razvan Petre, Kevin Quinn, Alan Davy, John Ronan, Juan Manuel Gonzalez, Tasos Zafeiropoulos, Athanassios Liakopoulos, Vassilios Kaldanis, Thomas Scherer, Rolland Vida, Felician Nemeth, Gabor Retvari, Slawomir Kuklinski, Martin Vigoureux, Pierre Peloso et Bruno Vidalenc : Final version of autonomic behaviours specifications (ABs) for the diverse networking environments. Rapport scientifique du projet FP7 EFIPSANS INFSO-ICT-215549/EFIPSANS/WP1/D1.7/Part1, 2010.

[CPK+09] Ranganai Chaparadza, Symeon Papavassiliou, Timotheos Kastrinogiannis, Andras Toth, Athanassios Liakopoulos et Mick Wilson : Creating a viable evolution path towards self-managing future internet via a standardizable reference model for autonomic network engineering. *In Towards the Future Internet - A European Research Perspective*, pages 136–147. IOSPress, 2009.

[CPP+10] Ranganai Chaparadza, Arun Prakash, Razvan Petre, Alexej Starschenko, Felician Nemeth, Michal Wodczak, Monika Grajzer, Tomasz zernicki, Slawomir Kuklinski, Jacek Wytrebowicz, Pawel Radziszewski, Bruno Vidalenc et Laurent Ciavaglia : Overall IPv6++ (EFIPSANS extended IPv6) feature combination scenarios for engineering autonomicity. Rapport scientifique du projet FP7 EFIPSANS INFSO-ICT-215549/EFIPSANS/WP2/D2.5, 2010.

[CPRW03] David D. Clark, Craig Partridge, J. Christopher Ramming et John T. Wroclawski : A knowledge plane for the internet. *In Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communications*, pages 3–10, Karlsruhe, Germany, 2003.

[CPT+10] Ranganai Chaparadza, Arun Prakash, Nikolay Tcholtchev, Razvan Petre, Symeon Papavassiliou, Giorgos Aristomenopoulos, Timotheos Kastrinogiannis, Mary Grammatikou, Christos Argyropoulos, Vassilis Merekoulias, Zhaojun Li, Michal Wodczak, Monika Grajzer, Tomasz zernicki, Yuhong Li, Wang Wendong, Xiangyang Gong, Yan Shi, Xin Li, Kevin Quinn, Aland Davy, Lei Shi, Juan Manuel Gonzalez, Vassilios Kaldanis, Slawomir Kuklinski, Krzysztof Cabaj, Szymon Lis, Krzysztof Szczypiorski, Laurent Ciavaglia et Bruno Vidalenc : Final version of the specification and description tables for decision elements (DEs). Rapport scientifique du projet FP7 EFIPSANS INFSO-ICT-215549/EFIPSANS/WP1/D1.7/Part3, 2010.

[CSC02] J. CROWELL, M. SHERESHEVSKY et C. CUKIC : Using fractal analysis to model software aging. Rapport technique, West Virginia University, Lane Department of CSEE, Morgantown, USA, 2002.

[Cse90] A. Csenki : Bayes predictive analysis of a fundamental software reliability model. *IEEE Transactions on Reliability*, 39(2):177–183, juin 1990.

[CTS03] Don Choi, D. Taylor et R. Seibel : Priority-based optical network protection and restoration with application to DOD networks. *In Proceedings of the IEEE Military Communications Conference, MILCOM*, volume 1, pages 298–303, octobre 2003.

[CTS09] R. Chaparadza, N. Tcholtchev et I. Schieferdecker : Implementation of the UniFAFF framework for autonomic fault-management in ANA networks. *In Proceedings of the Third International Conference on the Latest Advances in Networks, ICLAN*, pages 75–81, Toulouse, 2009.

[CWT+05] Fan-Tien Cheng, Shang-Lun Wu, Ping-Yen Tsai, Yun-Ta Chung et Haw-Ching Yang : Application cluster service scheme for near-zero-downtime services. *In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, pages 4062–4067, avril 2005.

[DBN+09] Dominique Dudkowski, Marcus Brunner, Giorgio Nunzi, Chiara Mingardi, Chris Foley, Miguel Ponce de Leon, Catalin Meirosu et Susanne Engberg : Architectural principles and elements of in-network management. *Proceedings of the 11th IFIP/IEEE International Symposium on Integrated Network Management, IM*, page 529–536, 2009.

[DDF+06] Simon Dobson, Spyros Denazis, Antonio Fernández, Dominique Gaïti, Erol Gelenbe, Fabio Massacci, Paddy Nixon, Fabrice Saffre, Nikita Schmidt et Franco Zambonelli : A survey of autonomic communications. *ACM Transaction on Autonomic and Adaptive System*, 1(2):223–259, 2006.

[DP07] Francesco De Pellegrini : BIONETS architecture : from networks to SerWorks. *In Proceedings of the 2nd International Conference on Bio-Inspired Models of Network, Information, and Computing Systems*, Budapest, Hungary, 2007.

[DPVM02] Carlotta Domeniconi, Chang-Shing Perng, Ricardo Vilalta et Sheng Ma : A classification approach for prediction of target events in temporal sequences. *In Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery, PKDD*, page 125–137. Lecture Notes in Computer Science, 2002.

[EKA03] Sebastian Elbaum, Satya Kanduri et Anneliese Amschler Andrews : Anomalies as precursors of field failures. *In Proceedings of the 14th International Symposium on Software Reliability Engineering, ISSRE*, page 108, 2003.

[Enn06] R. Enns : NETCONF Configuration Protocol. RFC 4741 (Proposed Standard), décembre 2006.

[ESRC09] G. Enyedi, P. Szilagyi, G. Retvari et A. Csaszar : IP fast ReRoute : lightweight not-via without additional addresses. *In Proceedings of the 28th IEEE International Conference on Computer Communications, INFOCOM*, pages 2771–2775, 2009.

[FB05] Pierre Francois et Olivier Bonaventure : An evaluation of IP-based fast reroute techniques. *In Proceedings of the ACM conference on Emerging network experiment and technology, CoNEXT*, page 244–245, New York, NY, USA, 2005.

[FB07] Pierre Francois et Olivier Bonaventure : Avoiding transient loops during the convergence of link-state routing protocols. *IEEE/ACM Transactions on Networking*, 15(6):1280–1292, 2007.

[FFEB05] Pierre Francois, Clarence Filsfils, John Evans et Olivier Bonaventure : Achieving sub-second IGP convergence in large IP networks. *ACM SIGCOMM Computer Communication Review*, 35(3):35–44, 2005.

[FMSN94] Daniele Frigioni, Alberto Marchetti-Spaccamela et Umberto Nanni : Incremental algorithms for the single-source shortest path problem. *In Proceedings of the 14th Conference on Foundations of Software Technology and Theoretical Computer Science*, page 113–124, London, UK, 1994. Lecture Notes in Computer Science.

[FT02] B. Fortz et M. Thorup : Optimizing OSPF/IS-IS weights in a changing world. *IEEE Journal on Selected Areas in Communications*, 20(4):756–767, 2002.

[FX07] Song Fu et Cheng-Zhong Xu : Quantifying temporal and spatial correlation of failure events for proactive management. *In Proceedings of the 26th IEEE International Symposium on Reliable Distributed Systems, SRDS*, page 175–184, 2007.

[GAVC05] A. Giorgetti, N. Andriolli, L. Valcarenghi et P. Castoldi : Failure-aware idle protection capacity reuse. *In Proceedings of the IEEE Global Telecommunications Conference, GLOBECOM*, volume 4, page 1899, 2005.

[GBB02] Kenny C Gross, Vatsal Bhardwaj et Randy Bickford : Proactive detection of software aging mechanisms in performance critical computers. *In Proceedings of the 27th Annual NASA Goddard Software Engineering Workshop, SEW*, page 17, 2002.

[GDB+09] Alex Galis, S Denazis, A Bassi, A Berl, A Fischer, H de Meer, J Srassner, S Davy, D Macedo, G Pujolle, JR Loyola, J Serrat, L Lefevre et A Cheniour : Management architecture and systems for future internet networks. *In Towards the Future Internet - A European Research Perspective*. IOSPress, Amsterdam, Netherlands, 2009.

[Ger10] Gerard Nguengang : *Une nouvelle approche de gestion de réseau : le pilotage autonomique*. Thèse de doctorat, Université Pierre et Marie Curie, Paris VI, Paris, France, 2010.

[GHM+05] Albert Greenberg, Gisli Hjalmtysson, David A. Maltz, Andy Myers, Jennifer Rexford, Geoffrey Xie, Hong Yan, Jibin Zhan et Hui Zhang : A clean slate 4D approach to network control and management. *In ACM SIGCOMM Computer Communication Review*, volume 35, pages 41–54, 2005.

[GLL+07] Prashasta Gujrati, Yawei Li, Zhiling Lan, Rajeev Thakur et John White : A meta-learning failure predictor for blue Gene/L systems. *In Proceedings of the International Conference on Parallel Processing*, page 40, 2007.

[GMT08] Michael Grottke, Rivalino Matias et Kishor S. Trivedi : The fundamentals of software aging. *In Proceedings of the IEEE International Conference on Software Reliability Engineering Workshops*, pages 1–6, Seattle, WA, USA, novembre 2008.

[GPSZ06] Dominique Gaiti, Guy Pujolle, Mikaël Salaun et Hubert Zimmermann : Autonomous network equipments. *In Autonomic Communication*, pages 177–185. 2006.

[GRcF03] M. Goyal, K.K. Ramakrishnan et Wu chi Feng : Achieving faster failure detection in OSPF networks. *In Proceedings of the IEEE International Conference on Communications, ICC*, volume 1, pages 296–300, 2003.

[Gro04] W. D Grover : The protected working capacity envelope concept : an alternate paradigm for automated service provisioning. *IEEE Communications Magazine*, 42(1):62– 69, janvier 2004.

[GSSM10] Lucas Guardalben, Susana Sargento, Paulo Salvador et Vitor Mirones : A cooperative hide and seek discovery over in network management. *In Proceedings of the IEEE/IFIP Network Operations and Management Symposium Workshops*, pages 217–224, Osaka, Japan, 2010.

[GVMVT98] S. Garg, A. Van Moorsel, K. Vaidyanathan et K. S Trivedi : A methodology for detection and estimation of software aging. *In Proceedings of the Ninth International Symposium on Software Reliability Engineering*, page 283, 1998.

[HE01] Greg Hamerly et Charles Elkan : Bayesian approaches to failure prediction for disk drives. *In Proceedings of the Eighteenth International Conference on Machine Learning, ICML*, page 202–209. Morgan Kaufmann Publishers Inc., 2001.

[HKM+96] K. Hatonen, M. Klemettinen, H. Mannila, P. Ronkainen et H. Toivonen : Knowledge discovery from telecommunication network alarm databases. *In Proceedings of the Twelfth International Conference on Data Engineering*, pages 115–122, 1996.

[HL97] M. Horowitz et S. Lunt : FTP Security Extensions. RFC 2228 (Proposed Standard), octobre 1997.

[HMKDE02] G. F HUGHES, J. F MURRAY, K. KREUTZ-DELGADO et C. ELKAN : Improved disk-drive failure warnings. *IEEE Transactions on Reliability*, 51(3):350– 357, septembre 2002.

[Hor01] Paul HORN : Autonomic computing : IBM's perspective on the state of information technology, 2001.

[HTM07] G. A HOFFMANN, K. S TRIVEDI et M. MALEK : A best practice guide to resource forecasting for computing systems. *IEEE Transactions on Reliability*, 56(4):615–628, décembre 2007.

[HZS99] J.L. HELLERSTEIN, Fan ZHANG et P. SHAHABUDDIN : An approach to predictive detection for service management. *In Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management, IM*, pages 309–322, Boston, MA, USA, 1999.

[ICM+02] Gianluca IANNACCONE, Chen-nee CHUAH, Richard MORTIER, Supratik BHATTACHARYYA et Christophe DIOT : Analysis of link failures in an IP backbone. *In ACM SIGCOMM Computer Communication Review*, pages 237–242, Marseille, France, 2002.

[Inf08] INFONETICS : Service provider capex, opex, ARPU, and subscribers. Rapport technique, 2008.

[ISO89] ISO : Open system interconnection (OSI) - basic reference model part 4, management framework technical specification. Standard ISO/ITU-T 7498-4, 1989.

[ISO98] ISO : Information technology -open system interconnection systems management overview. technical specification. Standard ISO/ITU-T 10040, 1998.

[Jad11] JADE : Java agent DEvelopment framework. http ://jade.tilab.com, 2011.

[JXLP09] Xuezhi JIANG, Mingwei XU, Qi LI et Lingtao PAN : Improving IGP convergence through distributed OSPF in scalable router. *In Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications*, pages 438–443, Los Alamitos, CA, USA, 2009.

[KAB+11] Vassilios KALDANIS, Domonkos ASZTALOS, Peter BENKO, Zoltan BACSKAY, Csaba SIMON, Felicián NÉMETH, Ferenc FICSOR, Wendong WANG, Xiangyang GONG, Xin LI, Shiduan CHENG, Yuhong LI, Alexej STARSCHENKO, Arun PRAKASH, Ranganai CHAPARADZA, Nikolay TCHOLTCHEV, Petre RAZVAN, Andras ZAHEMSZKY, Patrik TEPPO, Monika GRAJZER, Tomasz ZERNICKI, Mary GRAMMATIKOU, Timotheos KASTRINOGIANNIS, Vassilis MEREKOULIAS, Giorgos ARISTOMENOPOULOS, Constantinos MARINOS, Zhaojun LI, Mick WILSON, Krzysztof CABAJ, Sheila BECKER, Bruno VIDALENC, Anastasios ZAFEIROPOULOS, Thanassis LIAKOPOULOS, Slawomir KUKLINSKI, lukasz PODKALICKI, Pawel RADZISZEWSKI et Michal ULASKI : Final version of EFIPSANS demonstration results. Rapport scientifique du projet FP7 EFIPSANS INFSO-ICT-215549/EFIPSANS/WP5/D5.3, 2011.

[KC03] J.O. KEPHART et D.M. CHESS : The vision of autonomic computing. *Computer*, 36(1):41–50, 2003.

[KF05] Emre KICIMAN et Armando FOX : Detecting application-level failures in component-based internet services. *IEEE Transactions on Neural Networks*, 16(5):1027–1041, septembre 2005.

[KKY03] D. KATZ, K. KOMPELLA et D. YEUNG : Traffic Engineering (TE) Extensions to OSPF Version 2. RFC 3630 (Proposed Standard), septembre 2003. Updated by RFCs 4203, 5786.

[KW10] D. KATZ et D. WARD : Bidirectional Forwarding Detection (BFD). RFC 5880 (Proposed Standard), juin 2010.

[LAJ99] C. LABOVITZ, A. AHUJA et F. JAHANIAN : Experimental study of internet stability and backbone failures. *In Proceedings of the Twenty-Ninth Annual International Symposium on Fault-Tolerant Computing*, pages 278–285, 1999.

[LC98] R. LAL et G. CHOI : Error and failure analysis of a UNIX server. *In Proceedings of the third IEEE International High-Assurance Systems Engineering Symposium*, pages 232–239, novembre 1998.

[LC03] D. LEVY et R. CHILLAREGE : Early warning of failures through alarm analysis a case study in telecom voice mail systems. *In Proceedings of the 14th International Symposium on Software Reliability Engineering, ISSRE*, pages 271– 280, novembre 2003.

[LDRK10] Jingxian LU, Christophe DOUSSON, Benoit RADIER et Francine KRIEF : Towards an autonomic network architecture for self-healing in telecommunications networks. *In Mechanisms for Autonomous Management of Networks and Services*, pages 110–113. 2010.

[LFC07] CY. LEE, A. FARREL et S. De CNODDER : Exclude Routes - Extension to Resource ReserVation Protocol-Traffic Engineering (RSVP-TE). RFC 4874 (Proposed Standard), avril 2007. Updated by RFC 6001.

[LGLS07] Yawei LI, Prashasta GUJRATI, Zhiling LAN et Xian-he SUN : Fault-driven re-scheduling for improving system-level fault resilience. *In Proceedings of the International Conference on Parallel Processing*, page 39, 2007.

[LGS02] SuKyoung LEE, D. GRIFFITH et Nah-Oak SONG : An analytical approach to shared backup path provisioning in GMPLS networks. *In Proceedings of Military Communications Conference, MILCOM*, pages 75–80, Anaheim, CA, USA, 2002.

[Lin01] LINUX : Iperf. http ://iperf.sourceforge.net, 2001.

[Lin11a] LINUX : lm-sensors. http ://www.lm-sensors.org, 2011.

[Lin11b] LINUX : Traffic control. http ://tcng.sourceforge.net, 2011.

[Lin11c] LINUX : Ubuntu server. http ://www.ubuntu.com, 2011.

[LLR04] Chokchai LEANGSUKSUN, Tong LIU et Tirumala RAO : A failure predictive and policy-based high availability strategy for linux high performance computing cluster. *In Proceedings of the 5th LCI International Conference on Linux Clusters*, Austin, TX, USA, 2004.

[LQL08] Yonggang LI, Qun QIU et Longjiang LI : Availability-aware routing in optical networks with primary-backup sharing. *In Proceedings of the International Conference on High Performance Switching and Routing, HSPR*, pages 80–85, mai 2008.

[LRP07] J.P. LANG, Y. REKHTER et D. PAPADIMITRIOU : RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery. RFC 4872 (Proposed Standard), mai 2007. Updated by RFC 4873.

[LVT02] Lei LI, Kalyanaraman VAIDYANATHAN et Kishor S TRIVEDI : An approach for estimation of software aging in a web server. *In Proceedings of the International Symposium on Empirical Software Engineering*, page 91, 2002.

[LY07] Ji LI et K. L YEUNG : Most reliable routing in WDM mesh networks with arbitrary risk distribution. *In Proceedings of the IEEE International Conference on Communications, ICC*, pages 2198–2203, juin 2007.

[LZJ+06] Y. LIANG, Y. ZHANG, M. JETTE, Anand SIVASUBRAMANIAM et R. SAHOO : BlueGene/L failure analysis and prediction models. *In Proceedings of the International Conference on Dependable Systems and Networks, DSN*, pages 425–434, juin 2006.

[Man04] E. MANNIE : Generalized Multi-Protocol Label Switching (GMPLS) Architecture. RFC 3945 (Proposed Standard), octobre 2004. Updated by RFC 6002.

[Med10] Amelie MEDEM : *Conception de mécanismes d'amélioration de la gestion d'incidents dans les réseaux IP*. Thèse de doctorat, Université Pierre et Marie Curie, Paris VI, Paris, France, 2010.

[MFH08] Han MA, D. FAYEK et Pin-Han HO : Availability-constrained multipath protection in backbone networks with double-link failure. *In Proceedings of the IEEE International Conference on Communications, ICC*, pages 158–164, mai 2008.

[MHKD03] J. F. MURRAY, G. F. HUGUES et K. KREUTZ-DELGADO : Hard drive failure prediction using non-parametric statistical methods. *In Proceeding of the IEEE Congress on Evolutionary Computation*, Istanbull, Turkey, juin 2003.

[MIB+08] A. MARKOPOULOU, G. IANNACCONE, S. BHATTACHARYYA, Chen-Nee CHUAH, Y. GANJALI et C. DIOT : Characterization of failures in an operational IP backbone network. *IEEE/ACM Transactions on Networking*, 16(4):749–762, 2008.

[MMJ08] Carmen Mas MACHUCA, Oyvind MOE et Monika JAGER : Impact of protection schemes and network component's availability on operational expenditures. *Journal of Optical Networking*, 7(2):142–150, février 2008.

[MNBA09a] Andrej MIHAILOVIC, Gerard NGUENGANG, Julien BORGEL et Nancy ALONISTIOTI : Architectural principles for synergy of self-management and future internet evolution. *In Proceedings of the ICT-MobileSummit*, pages 3–8, 2009.

[MNBA09b] Andrej MIHAILOVIC, Gerard NGUENGANG, Julien BORGEL et Nancy ALONISTIOTI : Building knowledge lifecycle and situation awareness in self-managed cognitive future internet networks. *In Proceedings of the International Conference on Emerging Network Intelligence*, pages 3–8, Los Alamitos, CA, USA, 2009.

[Moy98] J. MOY : OSPF Version 2. RFC 2328 (Standard), avril 1998. Updated by RFC 5709.

[MP06] E. MANNIE et D. PAPADIMITRIOU : Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS). RFC 4427 (Informational), mars 2006.

[MTFM09] A. MEDEM, Renata TEIXEIRA, N. FEAMSTER et Michael MEULLE : Determining the causes of intradomain changes. Rapport technique, 2009.

[MTFM10] A. MEDEM, R. TEIXEIRA, N. FEAMSTER et M. MEULLE : Joint analysis of network incidents and intradomain routing changes. *In Proceedings of the International Conference on Network and Service Management, CNSM*, pages 198–205, 2010.

[MTM07] Amelie MEDEM, Renata TEIXEIRA et Michael MEULLE : Characterizing network events and their impact on routing. *In Proceedings of the ACM CoNEXT conference*, page 59 :1–59 :2, 2007.

[MZ06] Antonio MANZALINI et Franco ZAMBONELLI : Towards autonomic and situation-aware communication services : the CASCADAS vision. *In Proceedings of the IEEE Workshop on Distributed Intelligent Systems : Collective Intelligence and Its Applications*, page 383–388, 2006.

[NA85] Fares A. NASSAR et Dorothy M. ANDREWS : A methodology for analysis of failure prediction data. *In Proceedings of the IEEE Real-Time Systems Symposium*, pages 160–166, 1985.

[Nar00] Paolo NARVÁEZ : *Routing Reconfiguration in IP Networks*. Thèse de doctorat, Massachusetts Institute of Technology, MIT, Cambridge, MA, USA, 2000.

[NBTD07] Antonio NUCCI, Supratik BHATTACHARYYA, Nina TAFT et Christophe DIOT : IGP link weight assignment for operational tier-1 backbones. *IEEE/ACM Transactions on Networking*, 15(4):789–802, 2007.

[Net10] Nokia Siemens NETWORKS : Proactive care. http ://www.nokiasiemensnetworks.com/portfolio/services/care/proactive-care, 2010.

[NLY+07] Srihari NELAKUDITI, Sanghwan LEE, Yinzhe YU, Zhi-Li ZHANG et Chen-Nee CHUAH : Fast local rerouting for handling transient link failures. *IEEE/ACM Transactions on Networking*, 15:359–372, avril 2007.

[NSB+03] A. NUCCI, B. SCHROEDER, S. BHATTACHARYYA, N. TAFT et C. DIOT : IGP link weight assignment for transient link failures. *In Proceedings of International Teletraffic Congress*, 2003.

[nsm11] NSMAM : NS-3. http ://www.nsnam.org, 2011.

[NST01] Paolo NARVAEZ, Kai-Yeung SIU et Hong-Yi TZENG : New dynamic SPT algorithm based on a ball-and-string model. *IEEE/ACM Transactions on Networking*, 9(6):706–718, décembre 2001.

[OBOM03] Y. OHARA, M. BHATIA, N. OSAMU et J. MURAI : Route flapping effects on OSPF. *In Proceedings of the Symposium on Applications and the Internet Workshops*, pages 232–237, 2003.

[OL09] Milton OHRING et James R. LLOYD : *Reliability and Failure of Electronic Materials and Devices*. Academic Press, 2nd édition, 2009.

[Ora90] D. ORAN : OSI IS-IS Intra-domain Routing Protocol. RFC 1142 (Informational), février 1990.

[Ora08] ORACLE : Predictive self-healing in oracle solaris 10. http ://www.oracle.com/technetwork/systems/dtrace/zfs/index.html, 2008.

[PCF02] J. D PFEFFERMAN et B. CERNUSCHI-FRIAS : A nonparametric nonstationary procedure for failure prediction. *IEEE Transactions on Reliability*, 51(4):434– 442, décembre 2002.

[PE05] P. PILLAY-ESNAULT : OSPF Refresh and Flooding Reduction in Stable Topologies. RFC 4136 (Informational), juillet 2005.

[PHNP08] G. POST, J.M. HOUSSIN, L. NOIRIE et P. PELOSO : Networks for high-bandwith services combining photonic circuit cross-connects with packet switches. *In Proceedings of the 5th International Conference on Broadband Communications, Networks and Systems, BROADNETS*, pages 274–281, 2008.

[PIKF04] S. PASQUALINI, A. ISELT, A. KIRSTÄDTER et A. FROT : MPLS protection switching vs. OSPF rerouting - a simulative comparison. *In Proceedings of the 5th International Workshop on Quality of future Internet Services, QofIS*, septembre 2004.

[PIM11] Scott PORETSKY, Brent IMHOFF et Kris MICHIELSEN : Terminology for benchmarking link-state IGP data plane route convergence. http ://tools.ietf.org/html/draft-ietf-bmwg-igp-dataplane-conv-term-23, 2011.

[PKA+10] Symeon PAPAVASSILIOU, Timotheos KASTRINOGIANNIS, Giorgos ARISTOMENOPOULOS, Eirini-Eleni TSIROPOULOU, Juan Manuel GONZALEZ, Zhaojun LI, Nikolay TCHOLTCHEV, Michal WODCZAK, Monika GRAJZER, Yuhong LI, Wang WENDONG, John RONAN, Vassilios KALDANIS, Rolland VIDA, Gabor VINCZE, Peter SCHAFFER, Thorsten RIES et Bruno VIDALENC : Advanced network services in autonomic IPv6 networking : Architectures' performance analysis and evaluation. Rapport scientifique du projet FP7 EFIPSANS INFSO-ICT-215549/EFIPSANS/WP3/D3.6, 2010.

[PKM+09] Symeon PAPAVASSILIOU, Timotheos KASTRINOGIANNIS, Juan M., Zhaojun LI, Ranganai CHAPARADZA, Petre RAZVAN, Nikolay TCHOLTCHEV, Michal WODCZAK, Monika GRAJZER, Yuhong LI, Sheila BECKER, Thorsten RIES, Xiangyang GONG, Yan SHI, Xin LI, Wang WENDONG, John RONAN, Tasos ZAFEIROPOULOS, Athanassios LIAKOPOULOS, Vassilios KALDANIS, Rolland VIDA, Gabor VINCZE, Bruno VIDALENC, Giorgos ARISTOMENOPOULOS et Eleni TSIROPOULOU : Advanced network services in autonomic IPv6 networking : Performance analysis and evaluation. Rapport scientifique du projet FP7 EFIPSANS INFSO-ICT-215549/EFIPSANS/WP3/D3.2, 2009.

[PLR10] J. PESIC et E. LE ROUZIC : Fault prediction and proactive restoration in optical network links using events recognition. *BONE-TIGER2 Summer School*, septembre 2010.

[Pre98] AT&T PRESS : AT&T announces cause of frame-relay network outage. http ://www.networkworld.com/news/0420frameout.html, 1998.

[PSA05] P. PAN, G. SWALLOW et A. ATLAS : Fast Reroute Extensions to RSVP-TE for LSP Tunnels. RFC 4090 (Proposed Standard), mai 2005.

[PSBDG98] M. PIZZA, L. STRIGINI, A. BONDAVALLI et F. DI GIANDOMENICO : Optimal discrimination between transient and permanent faults. *In Proceedings of the Third IEEE International High-Assurance Systems Engineering Symposium*, pages 214–223, novembre 1998.

[Qua09] Sun QUAN : On parameter settings of network keep-alive protocol for failure detection. *In Proceedings of the 2nd IEEE International Conference on Broadband Network & Multimedia Technology, IC-BNMT*, pages 19–23, 2009.

[Qua11] QUAGGA : Software routing suite. http ://www.quagga.net, 2011.

[RBGK03] Rajeev RASTOGI, Yuri BREITBART, Minos GAROFALAKIS et Amit KUMAR : Optimal configuration of OSPF aggregates. *IEEE/ACM Transactions on Networking*, 11(2):181–194, avril 2003.

[RCT+11] G. RETVARI, L. CSIKOR, J. TAPOLCAI, G. ENYEDI et A. CSASZAR : Optimizing IGP link costs for improving IP-level resilience. *In Proceeding of the International Workshop on Design Of Reliable Communication Networks, DRCN*, 2011.

[Rea01] Light READING : In qwest outage, ATM takes some heat. http ://www.lightreading.com/document.asp ?doc_id=4624, 2001.

[RI07] Alex RAJ et Oliver C. IBE : A survey of IP and multiprotocol label switching fast reroute schemes. *Computer Networks : The International Journal of Computer and Telecommunications Networking*, 51(8):1882–1907, 2007.

[RM90] M.T. ROSE et K. MCCLOGHRIE : Structure and identification of management information for TCP/IP-based internets. RFC 1155 (Standard), mai 1990.

[RM91] M.T. ROSE et K. MCCLOGHRIE : Concise MIB definitions. RFC 1212 (Standard), mars 1991.

[RMD05] S. RAI, B. MUKHERJEE et O. DESHPANDE : IP resilience within an autonomous system : current approaches, challenges, and future directions. *IEEE Communications Magazine*, 43(10):142–149, 2005.

[RNCS09] Gabor RETVARI, Felician NEMETH, Ranganai CHAPARADZA et Robert SZABO : OSPF for implementing self-adaptive routing in autonomic networks : A case study. *In Proceedings of the 4th IEEE International Workshop on Modelling Autonomic Communications Environments, MACE*, page 72–85, Berlin, Germany, 2009. Lecture Notes in Computer Science.

[RNP+11] Gábor RÉTVÁRI, Felicián NÉMETH, Arun PRAKASH, Ranganai CHAPARADZA, Ibrahim HOKELEK, Mariusz FECKO, Michal WÓDCZAK et Bruno VIDALENC : A guideline for realizing the vision of autonomic networking : Implementing self-adaptive routing on top of OSPF. *In Formal and Practical Aspects of Autonomic Computing and Networking*. IGI Global, mai 2011.

[RVC01] E. ROSEN, A. VISWANATHAN et R. CALLON : Multiprotocol Label Switching Architecture. RFC 3031 (Proposed Standard), janvier 2001. Updated by RFC 6178.

[SAL06] John STRASSNER, N AGOULMINE et E LEHTIHET : FOCALE : a novel autonomic networking architecture. *In Proceedings of the Latin American Autonomic Computing Symposium, LAACS*, 2006.

[SB10] M. SHAND et S. BRYANT : IP Fast Reroute Framework. RFC 5714 (Informational), janvier 2010.

[SBP11] Mike SHAND, Stewart BRYANT et Stefano PREVIDI : IP fast reroute using not-via addresses. http ://tools.ietf.org/html/draft-ietf-rtgwg-ipfrr-notvia-addresses-07, 2011.

[SCC+03] M. SHEREshevsky, J. CROWELL, B. CUKIC, V. GANDIKOTA et Yan LIU : Software aging and multifractality of memory resources. *In Proceedings of the International Conference on Dependable Systems and Networks*, pages 721–730, San Francisco, CA, USA, 2003.

[SFL+08] John STRASSNER, Joel FLECK, David LEWIS, Manish PARASHAR et Willie DONNELLY : The role of standardization in future autonomic communication systems. *In Proceedings of the Fifth IEEE Workshop on Engineering of Autonomic and Autonomous Systems, EASE*, pages 165–173, 2008.

[SFT+06] Shoichiro SENO, Teruko FUJII, Motofumi TANABE, Eiichi HORIUCHI, Yoshimasa BABA et Tetsuo IDEGUCHI : Optical path protection with fast extra path preemption. *IEICE Transactions*, pages 3032–3039, 2006.

[SG01] Aman SHAIKH et Albert GREENBERG : Experience in black-box OSPF measurement. *In Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, pages 113–125, San Francisco, California, USA, 2001.

[SHc+10] James P. G. STERBENZ, David HUTCHISON, Egemen K. ÇETINKAYA, Abdul JABBAR, Justin P. ROHRER, Marcus SCHÖLLER et Paul SMITH : Resilience and survivability in communication networks : Strategies, principles, and survey of disciplines. *Computer Networks, COMNET, Special Issue on Resilient and Survivable Networks*, 54(8):1245–1265, juin 2010.

[SHS+11] P. SMITH, D. HUTCHISON, J.P.G. STERBENZ, M. SCHÖLLER, A. FESSI, M. KARALIOPOULOS, C. LAC et B. PLATTNER : Network resilience : a systematic approach. *IEEE Communications Magazine*, 49(7):88 –97, juillet 2011.

[SL04] H. SMIT et T. LI : Intermediate System to Intermediate System (IS-IS) Extensions for Traffic Engineering (TE). RFC 3784 (Informational), juin 2004. Obsoleted by RFC 5305, updated by RFC 4205.

[SM07] Felix SALFNER et Miroslaw MALEK : Using hidden semi-markov models for effective online failure prediction. *In Proceedings of the 26th IEEE International Symposium on Reliable Distributed Systems*, pages 161–174, 2007.

[SN05] Amir SIDDIQI et Biswajit NANDY : Improving network convergence time and network stability of an OSPF-Routed IP network. *In Proceedings of the 4th IFIP-TC6 international conference on Networking Technologies, Services, and Protocols, NETWORKING*, page 469–485, Berlin, Heidelberg, 2005. Lecture Notes in Computer Science.

[Sof11] VMware Virtualization SOFTWARE : VMware server. http ://www.vmware.com, 2011.

[SOR+03] R. K SAHOO, A. J OLINER, I. RISH, M. GUPTA, J. E MOREIRA, S. MA, R. VILALTA et A. SIVASUBRAMANIAM : Critical event prediction for proactive management in large-scale computer clusters. *In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, page 426–435, Washington, D.C., 2003.

[SPW09] Bianca SCHROEDER, Eduardo PINHEIRO et Wolf-Dietrich WEBER : DRAM errors in the wild : a large-scale field study. *In Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems, SIGMETRICS*, page 193–204, Seattle, WA, USA, 2009.

[SRM02] L. SAHASRABUDDHE, S. RAMAMURTHY et B. MUKHERJEE : Fault management in IP-over-WDM networks : WDM protection versus IP restoration. *IEEE Journal on Selected Areas in Communications*, 20(1):21–33, 2002.

[SSJ11] SSJ : Simulation stochastique en java. http ://www.iro.umontreal.ca/~simardr/ssj, 2011.

[SSM06] F. SALFNER, M. SCHIESCHKE et M. MALEK : Predicting failures of computer systems : a case study for a telecommunication system. *In 20th International Parallel and Distributed Processing Symposium, IPDPS*, avril 2006.

[ST08] Felix SALFNER et Steffen TSCHIRPKE : Error log processing for accurate failure prediction. *In Proceedings of the First USENIX Workshop on the Analysis of System Logs, WASL*, 2008.

[Sur02] Yankee Group SURVEY : Network downtime survey. Rapport technique, 2002.

[SVKD00] Aman SHAIKH, Anujan VARMA, Lampros KALAMPOUKAS et Rohit DUBE : Routing stability in congested networks : experimentation and analysis. *In ACM SIGCOMM Computer Communication Review*, volume 30, page 163–174, New York, NY, USA, août 2000.

[SWL+03] A. SHAIKH, Dongmei WANG, Guangzhi LI, J. YATES et C. KALMANEK : An efficient algorithm for OSPF subnet aggregation. *In Proceedings of the 11th IEEE International Conference on Network Protocols*, pages 200– 209, novembre 2003.

[Sys02] Cisco SYSTEMS : OSPF support for fast hellos packets. http ://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/fasthelo.html, 2002.

[TA03] D. TURNBULL et N ALLDRIN : Failure prediction in hardware systems. Rapport technique, University of California, San Diego, USA, 2003.

[TCC10] Nikolay TCHOLTCHEV, Agnieszka Betkowska CAVALCANTE et Ranganai CHAPARADZA : Scalable markov chain based algorithm for fault-isolation in autonomic networks. *In Proceedings of the IEEE Global Telecommunications Conference, GLOBECOM*, pages 1–6, Miami, FL, USA, décembre 2010.

[TDQ+09] Nikolay TCHOLTCHEV, Alan DAVY, Kevin QUINN, John ROHNAN, Bruno VIDALENC, Bela BERDE, Monika GRAJZER, Agnieszka CAVALCANTE, Ranganai CHAPARADZA, Constantinos MARINOS, Christos ARGYROPOULOS et Cynthia WAGNER : Components and mechanisms for autonomic fault-management. Rapport scientifique du projet FP7 EFIPSANS INFSO-ICT-215549/EFIPSANS/WP4/D4.5, 2009.

[TGV09] Nikolay TCHOLTCHEV, Monika GRAJZER et Bruno VIDALENC : Towards a unified architecture for resilience, survivability and autonomic fault-management for self-managing networks. *In Proceedings of the International Conference on Service-oriented computing, ICSOC/ServiceWave, Mona+*, page 335–344, Stockholm, Sweden, 2009. Lecture Notes in Computer Science.

[Tia03] H. TIANFIELD : Multi-agent based autonomic architecture for network management. *In Proceedings of the IEEE International Conference on Industrial Informatics, INDIN*, pages 462–469, 2003.

[TMN96] TMN : Recommendation m. 3010, principles for a telecommunications management network. 1996.

[TWFV06] M. TACCA, Kai WU, A. FUMAGALLI et J.-P. VASSEUR : Local detection and recovery from multi-failure patterns in MPLS-TE networks. *In IEEE International Conference on Communications, ICC*, volume 2, pages 658–663, 2006.

[VAH+02] R. VILALTA, C. V APTE, J. L HELLERSTEIN, S. MA et S. M WEISS : Predictive algorithms in the management of computer systems. *IBM System Journal*, 41(3):461–474, juillet 2002.

[VC10] B. VIDALENC et L. CIAVAGLIA : Proactive fault management based on risk-augmented routing. *In Proceedings of the IEEE Global Telecommunications Conference, GLOBECOM Workshops*, pages 481–485, Miami, USA, décembre 2010.

[VCL+11] Bruno VIDALENC, Laurent CIAVAGLIA, Yuhong LI, Xin LI, Lei ZHOU, Nikolay Vassilev TCHOLTCHEV, Ranganai CHAPARADZA et Michal WODCZAK : Resilience, survivability and/for autonomicity in IPv6 networks. Rapport scientifique du projet FP7 EFIPSANS INFSO-ICT-215549/EFIPSANS/WP3/D3.4, 2011.

[VGD11] Bruno VIDALENC et Samir GHAMRI-DOUDANE : Method for detecting failure of a router. Brevet européen EP2363982, Alcatel-Lucent, septembre 2011.

[VGDNR13] Bruno VIDALENC, Samir GHAMRI-DOUDANE, Ludovic NOIRIE et Eric RENAULT : Adaptive failure detection timers for IGP networks. *IFIP/IEEE International Symposium on Integrated Network Management, IM*, mai 2013.

[VLC11] VLC : VideoLAN. http ://www.videolan.org, 2011.

[VM02] Ricardo VILALTA et Sheng MA : Predicting rare events in temporal domains. *In Proceedings of the IEEE International Conference on Data Mining, ICDM*, page 474, 2002.

[VNC12] Bruno VIDALENC, Ludovic NOIRIE et Laurent CIAVAGLIA : GMPLS adaptive level of recovery. *In Proceedings of the IEEE International Conference on Communications, ICC*, pages 2768–2773, Ottawa, Canada, juin 2012.

[VNCR13] Bruno VIDALENC, Ludovic NOIRIE, Laurent CIAVAGLIA et Eric RENAULT : Dynamic risk-aware routing for OSPF networks. *IFIP/IEEE International Symposium on Integrated Network Management, IM*, mai 2013.

[VNR+10] Bruno VIDALENC, Felicián NÉMETH, Gábor RÉTVÁRI, Zoltan THEISZ, Zhaojun LI, Ranganai CHAPARADZA, Nikolay TCHOLTCHEV, Giorgos ARISTOMENOPOULOS, Mary GRAMMATIKOU, Timotheos KASTRINOGIANNIS, Symeon PAPAVASILEIOU, Mariusz FECKO, Michal WODCZAK, Monika GRAJZER, Juan M. GONZÁLEZ, Thomas SCHERER, Vassilios KALDANIS, Alan DAVY, Kevin QUINN, John RONAN et Slawomir KUKLINSKI : Final version of requirements specifications (RQs) regarding features, required in IPv6 protocols, network architectures and paradigms, in order to implement the specified autonomic behaviours. Rapport scientifique du projet FP7 EFIPSANS INFSO-ICT-215549/EFIPSANS/WP1/D1.8, 2010.

[VPD04] Jean-Philippe VASSEUR, Mario PICKAVET et Piet DEMEESTER : *Network Recovery : Protection and Restoration of Optical, SONET-SDH, IP, and MPLS*. Morgan Kaufmann, 1 édition, août 2004.

[VT99] K. VAIDYANATHAN et K. S TRIVEDI : A measurement-based model for estimation of resource exhaustion in operational software systems. *In Proceedings of the 10th International Symposium on Software Reliability Engineering*, pages 84–93, 1999.

[WCKR03] S. WALDBUSSER, R. COLE, C. KALBFLEISCH et D. ROMASCANU : Introduction to the Remote Monitoring (RMON) Family of MIB Modules. RFC 3577 (Informational), août 2003.

[WCLL08] Fang WANG, Shanzhi CHEN, Xin LI et Yuhong LI : A route flap suppression mechanism based on dynamic timers in OSPF network. *In Proceedings of the 9th International Conference for Young Computer Scientists, ICYCS*, pages 2154–2159, 2008.

[WDS08] Fetahi WUHIB, Mads DAM et Rolf STADLER : Decentralized detection of global threshold crossings using aggregation trees. *Computer Networks, COMNET*, 52(9):1745–1761, juin 2008.

[WDSC07] Fetahi WUHIB, Mads DAM, Rolf STADLER et Alexander CLEMM : Robust monitoring of network-wide aggregates through gossiping. *In Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management, IM*, pages 226–235, Munich, Germany, mai 2007.

[Wei99] Gary M. WEISS : Timeweaver : a genetic algorithm for identifying predictive patterns in sequences of events. *In Proceedings of the Genetic and Evolutionary Computation Conference*, page 718–725. Morgan Kaufmann, 1999.

[WGR98] Amy WARD, Peter GLYNN et Kathy RICHARDSON : Internet service performance failure detection. *In ACM SIGMETRICS Performance and Evaluation Review*, volume 26, page 38–43, 1998.

[WL09] Stefan WALLIN et Viktor LEIJON : Telecom network and service management : An operator survey. *In Proceedings of the 12th IFIP/IEEE International Conference on Management of Multimedia and Mobile Networks and Services : Wired-Wireless Multimedia Networks and Services Management, MMNS*, page 15–26, Berlin, Germany, 2009. Lecture Notes in Computer Science.

[WN07] Junling WANG et Srihari NELAKUDITI : IP fast reroute with failure inferencing. *In Proceedings of the SIGCOMM workshop on Internet network management*, pages 268–273, Kyoto, Japan, 2007.

[Wor01] Network WORLD : AT&T, customers grapple with ATM net outage. http ://www.networkworld.com/news/2001/0226attatm.html, 2001.

[WTVL13] Michal WODCZAK, Nikolay TCHOLTCHEV, Bruno VIDALENC et Yuhong LI : Design and evaluation of techniques for resilience and survivability of the routing node. *International Journal of Adaptive, Resilient and Autonomic Systems, IJARAS*, 4(3), 2013.

[WWM+10] Ye WANG, Hao WANG, Ajay MAHIMKAR, Richard ALIMI, Yin ZHANG, Lili QIU et Yang Richard YANG : R3 : resilient routing reconfiguration. *ACM SIGCOMM Computer Communication Review*, 40(4):291–302, 2010.

[XN98] Xipeng XIAO et Lionel M. NI : Parallel routing table computation for scalable IP routers. *In Proceedings of the Second International Workshop on Network-Based Parallel Computing : Communication, Architecture, and Applications*, page 144–158, London, UK, 1998. Lecture Notes in Computer Science.

[XTMM10] Ming XIA, Massimo TORNATORE, Charles U. MARTEL et Biswanath MUKHERJEE : Risk-aware provisioning for optical WDM mesh networks. *IEEE/ACM Transactions on Networking*, 2010.

[YVJ05] S. YUAN, S. VARMA et J. P JUE : Minimum-color path problems for reliability in mesh networks. *In Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM*, volume 4, pages 2658– 2669, mars 2005.

[ZXW07] Yuanbo ZHU, Mingwei XU et Qian WU : Parallel dynamic SPT update algorithm in OSPF. *In* Victor MALYSHKIN, éditeur : *Parallel Computing Technologies*, volume 4671, pages 346–359. Lecture Notes in Computer Science, 2007.

[ZZZ+07] Jing ZHANG, Keyao ZHU, Hui ZANG, N. S MATLOFF et B. MUKHERJEE : Availability-aware provisioning strategies for differentiated protection services in wavelength-convertible WDM mesh networks. *IEEE/ACM Transactions on Networking*, 15(5):1177–1190, octobre 2007.

Glossaire

E Ensemble des liens du réseau ($G = (N, E)$). 76, 104, 141

F Ensemble des flux de trafic du réseau G. 76, 104, 141, 143

FN Faux négatif (*False Negative*) *i.e.* Panne n'étant pas précédé par une prédiction. 57, 143

FP Faux positif (*False Positive*) *i.e.* Prédiction de panne à tort. 57, 79, 107, 143

G Graphe représentant le réseau ($G = (N, E)$). 76, 77, 104, 105, 141, 143

$In(f)$ Nœud source du flux f ($In(f) \in N$). 76, 86, 104, 116, 141

$InitialCost(link_i)$ Valeur initiale de la métrique du lien i.. 101

$MTBF(n)$ Temps moyen entre deux pannes du nœud n. 76, 104, 141

$MTTR(n)$ Temps de réparation moyen pour le nœud n. 76, 105, 141

$MaxInitialCost$ La grande métrique initiale du réseau.. 101

$MaxPossibleCost$ Valeur positive maximum d'un métrique dans OSPF égale à $2^{16} - 2$.. 100, 101

$Min(CostR)$ La plus petite métrique de lien risqué du réseau.. 101

N Ensemble des nœuds du réseau ($G = (N, E)$). 68, 76, 104, 141

$Out(f)$ Nœud destination du flux f ($Out(f) \in N$). 76, 86, 104, 116, 141

$P_{node}(n)$ Probabilité de panne du nœud n ($n \in N$). 76, 105, 141, 143

Precision Proportion de bonnes prédictions par rapport à toutes les prédictions. 57, 62–64, 76, 79, 81–93, 105, 107–123, 128, 141–143, 145–153, 155, 182, 187, 193, 198, 202, 205, 209, 212, 217–220

Precision(n) Proportion de bonnes prédictions par rapport à toutes les prédictions pour le nœud $n \in N$. 105, 142

Recall Proportion de pannes prédites par rapport aux pannes totales. 57, 62–64, 74, 76, 77, 79, 81–87, 89–93, 105–123, 126, 141–143, 145–153, 155, 173, 175, 180, 182, 185, 187, 193, 197, 202, 205, 209, 212, 217–220

Recall(n) Proportion de pannes prédites par rapport aux pannes totales pour le nœud $n \in N$. 76, 105, 142

$RiskyMetric(link_i)$ Valeur de la métrique lorsque le lien i est risqué.. 99, 101

TN Vrai négatif (*True Negative*) *i.e.* Cas où le risque de panne est nul et qu'aucune panne n'apparaît. 57

TP Vrai positif (*True Positive*) *i.e.* Panne prédite à raison. 57, 79, 107, 143

T_P Temps d'indisponibilité moyen lors de l'utilisation du mécanisme de protection (ms). 141, 145

$T_P(f, n)$ Temps d'indisponibilité moyen pour le flux f utilisant un mécanisme de protection lorsque que le nœud n tombe en panne (ms). 141

T_R Temps d'indisponibilité moyen lors de l'utilisation du mécanisme de restauration (ms). 141, 145

$T_R(f, n)$ Temps d'indisponibilité moyen pour le flux f utilisant un mécanisme de restauration lorsque que le nœud n tombe en panne (ms). 141

$\Delta CostR$ Différence maximum dans le réseau entre deux métriques de lien ayant un risque de panne important.. 101

Δt_d Fenêtre de temps utilisée par une prédiction pour la conservation des paramètres observés. 57, 62, 64

Δt_l Délai minimum entre une prédiction et le moment supposé d'occurrence de la panne. 57, 62–64, 125

Δt_p Le temps de validité d'une prédiction. 57, 62, 64, 72, 74, 76, 79, 81, 84–87, 91, 93, 106–110, 115, 116, 121–123, 136, 141–143, 145, 147–150, 153, 155, 158

Δt_w Délai maximum pour mettre en place un dispositif de gestion proactif. 57

$\mu(f)$ Débit du flux f (bit/s). 76, 104, 141

$d^+(n)$ degré sortant du nœud $n \in N$, *i.e.* nombre de liens sortant du routeur n. 79

$d^-(n)$ degré entrant du nœud $n \in N$, *i.e.* nombre de liens entrant du routeur n. 79

f un flux de trafic du réseau ($f \in F$). 76, 104, 106, 141

n un nœuds du réseau ($n \in N$). 76, 105, 141

$p(f)$ Ensemble de nœuds composant le chemin de protection du flux f ($p(f) \subseteq N$). 141

$r(f, n)$ Ensemble de nœuds composant le chemin de restauration du flux f lors de la panne du nœud n ($r(f, n) \subseteq N - \{n\}$). 141

$sp(f)$ Ensemble de nœuds composant le plus court chemin du flux f ($sp(f) \subseteq N$). 76, 104, 106

$sp_{RAR}(f)$ Ensemble de nœuds composant le nouveau plus court chemin du flux f ($sp(f) \subseteq N$ lorsque le mécanisme RAR à modifié une des métriques de l'ancien plus court chemin $sp(f)$ suite à une prédiction de panne.). 104, 106

t Instant donné dans le temps. 105, 106

t_C Temps de convergence moyen lors d'une panne sur un nœud $n \in N$. 76, 96, 105, 109, 128

t_D Temps de détection moyen d'une panne sur un nœud. 68, 76, 96, 105, 125

t_F Temps de diffusion moyen d'une information de panne. 68, 76, 81, 96, 105, 109

t_U Temps de mise à jour des tables de routage (RIB) et de *forwarding* (FIB). 68, 76, 81, 97, 105, 109

t_{FC} Temps de convergence moyen lors d'une panne sur un nœud $n \in N$ avec l'utilisation du protocole *Hello* à une fréquence rapide. 77

t_{FHI} Valeur rapide du *Hello Interval*. 75, 77

t_{FRDI} Valeur du *Router Dead Interval* correspondant à la période d'envoi de message *Hello* t_{FHI}. 75, 77

t_{HI} Valeur du *Hello Interval*. 75–77, 79, 81, 105, 109

t_{RDI} Valeur du *Router Dead Interval*. 76, 77, 105

t_{SC} Temps de convergence moyen lors d'une panne sur un nœud $n \in N$ avec l'utilisation du protocole *Hello* à une fréquence lente. 77

t_{SHI} Valeur lente du *Hello Interval*. 75, 77, 81, 109

t_{SP} Temps de recalcule des plus court chemins. 76, 96, 105

t_{SRDI} Valeur du *Router Dead Interval* correspondant à la période d'envoi de message *Hello* t_{SHI}. 75, 77

$w(f)$ Ensemble de nœuds composant le chemin principal du flux f ($w(f) \subseteq N$). 141

Hello Interval Période d'envoi des messages *Hello*. 68–78, 81, 84, 85, 91, 92, 105, 125

Hello Message envoyé pour la détection de pannes. 13, 15, 65–92, 97, 102, 105, 109, 116, 158, 159, 174–177, 195–200, 217–219

Ingress router Routeur d'entrée dans un domaine (G)MPLS. 130, 136

Router Dead Interval Compteur déclenchant la suppression d'un routeur voisin s'il est atteind avant la réception d'un message *Hello*. 68, 70, 73–78, 105, 125

Acronymes

ABR *Area Border Routers*. 67, 160

AFDT *Adaptive Failure Detection Timers*. 53, 63, 71–73, 75–93, 104, 128, 158, 159, 217

AFI *Autonomic network engineering for the self-managing Future Internet*. 42

ALR *Adaptive Level of Recovery*. 52, 53, 135, 136, 138, 139, 141–147, 149–155, 158, 159, 218

ANA *Autonomic Network Architecture*. 31, 38

API *Application Programming Interface*. 124

AS *Autonomous System*. 66

ASBR *Autonomous System Border Router*. 67, 160

ASF *Action Synchronization Functions*. 50–52

ATM *Asynchronous Transfer Mode*. 19

AutoI *Autonomic Internet*. 33

BFD *Bidirectional Forwarding Detection*. 71, 73

BGP *Border Gateway Protocol*. 44, 53, 160

BIOS *Basic Input Output System*. 124

CAPEX Dépenses d'investissement de capital (*CApital EXpenditure*). 12, 21, 37, 52, 130, 134, 143, 144, 149

CLI *Command Line Interface*. 124

CMR *Causality Model Repository*. 50, 51

CR-LDP *Constraint-Based Routing Label Distribution Protocol*. 131, 133

DE *Decision Element*. 41

DMR *Dependability Model Repository*. 50, 51

DMTF *Distributed Management Task Force*. 21

DWDM *Dense Wavelength-Division Multiplexing*. 131

ECMP *Equal-cost multi-path routing*. 98

EFIPSANS *Exposing the features in IP version six protocols that can be exploited/extended for the purposes of designing/building autonomic networks and services*. 12, 13, 41, 157

ER-LSP *Explicitly Routed Label Switched Path*. 131

ERO *Explicit Route Object*. 131

ETSI *European Telecommunications Standards Institute*. 42

FAB *Forwarding Adjacency Builder*. 143, 144, 150

FCAPS *Fault, Configuration, Accounting, Performance, Security*. 21, 37

FDLI *Fault-Diagnosis/Localization/Isolation functions*. 49, 51

FIB *Forwarding Information Base*. 67, 68, 96, 98

FM_DE *Fault-Management Decision Element*. 44, 48–52, 157

FMF *Fault Masking Function*. 50–53, 63, 136, 138, 139

FOCALE *Foundation, Observation,Compare, Act, Learn, et rEason*. 31

FRAF *Fault-Removal Assessment Functions*. 50, 51

FRF *Fault-Removal Functions*. 50, 51

GANA *Generic Autonomic Network Architecture*. 12, 15, 30, 37, 41–49, 64, 157, 217

GMPLS *Generalized MultiProtocol Label Switching*. 11–13, 15, 52, 53, 95, 97, 104, 128–138, 140, 144, 149, 150, 155, 158, 159, 162, 164, 168, 218, 220

HD Haute Définition. 127

HSS *Home Subscriber Server*. 160

HTTP *Hypertext Transfer Protocol*. 23

I-CSF *Interrogating-Call Session Control Function*. 160

ICMP *Internet Control Message Protocol*. 43

IDE *Incident Information Dissemination Engine*. 50

IETF *Internet Engineering Task Force*. 19, 21, 23, 71, 130, 131, 159

IF InterFace réseau. 75, 80, 103, 139

IGP *Interior Gateway Protocol*. 11, 13, 15, 66–68, 70, 71, 73, 74, 77, 79, 81, 82, 85, 87, 91, 93, 96–102, 104, 106–115, 117, 118, 131, 135, 218

IMS *IP Multimedia Subsystem*. 37, 160

INM *In Network Management*. 32

IP *Internet Protocol*. 11–13, 15, 19, 38, 54, 66, 67, 69–71, 73–75, 93, 96–100, 103, 104, 125, 128, 130, 131, 135, 157–160, 217, 229

IP FRR *IP Fast ReRoute*. 98, 99, 135

IPC *Inter-Process Communication*. 124

IS-IS *Intermediate system to intermediate system*. 11, 44, 53, 66–68, 71, 73, 76, 77, 96, 99, 104, 106, 130

IS-IS-TE *Intermediate System to Intermediate System - Traffic Engineering*. 131

ISO *International Organization for Standardization*. 21

ITU-T *International Telecommunication Union - Telecommunication Standardization Sector*. 21, 22

LDP *Label Distribution Protocol*. 130, 131

LFA *Loop Free Alternate*. 98, 135

LSA *Link State Advertisements*. 67–70, 96, 97, 124, 134

LSDB *Link State DataBase*. 68, 96, 97

LSP *Label Switched Path*. 71, 104, 130–139, 150, 155

LSR *Label Switched Router*. 130, 131

MAPE-K *Monotoring, Analyse, Planification, Execution - Knowledge*. 25

MBB *Make-Before-Break*. 104, 133, 136–138

ME *Managed Entity*. 41

MGCF *Media Gateway Controller Function*. 160

MGW *Media Gateway*. 160

MIB *Management Information Base*. 21, 23

Mon_DE *Monitoring Decision Element*. 51

MPLS *MultiProtocol Label Switching*. 11, 19, 66, 98, 130–132

MPLS FRR *MPLS Fast ReRoute*. 98, 135

MTBF *Mean Time Between Failure*. 52, 62, 81–83, 86–89, 108–110, 112–119, 122, 123, 141, 145–147, 150–152, 172, 174, 178, 179, 181, 184, 186, 192, 195, 196, 201, 204, 208, 211, 217–220

MTTR *Mean Time To Repair*. 81–83, 85–89, 108, 109, 112, 113, 115–119, 128, 141, 145–147, 150–152, 172, 174, 179, 181, 184, 186, 192, 195, 201, 204, 208, 211, 217–220

NMS *Network Management System*. 19, 26, 62

OPEX Dépenses d'exploitation (*OPerational EXpenditure*). 12, 20, 37, 85, 130, 134, 149

OSPF *Open Shortest Path First*. 11, 12, 30, 44, 52, 53, 63, 66–69, 71, 73, 74, 76, 77, 80, 96, 98–101, 104, 106–108, 110, 117, 123–125, 130

OSPT-TE *Open Shortest Path First-Traffic Engineering*. 131, 136, 141

OSS *Operating Suport System*. 62

P Protection (G)MPLS. 132, 141

P-CSF *Proxy-Call Session Control Function*. 160

PBM *Policy Based Network Management*. 27

QoE *Quality of Experience*. 127, 131

QoS Qualité de service (*Quality of Service*). 11, 66, 69, 97, 99, 130–132, 134, 157

R Restauration (G)MPLS. 132, 141

R&S_DE *Resilience and Survivability Decision Element*. 44, 48–53, 63, 124, 136, 138, 149, 157

RAM *Risk Assessment Module*. 50, 52–54, 62–64, 71, 73, 75, 81–84, 86, 93, 96, 99, 102, 103, 108, 111, 113, 116, 122, 124, 128, 135, 136, 138, 139, 146, 147, 149, 157, 217

RAR *Risk-Aware Routing*. 53, 63, 95, 96, 99–115, 117, 118, 121–128, 158, 215, 216, 218, 220, 228

RFC *Request For Comments*. 19, 130

RIP *Routing Information Protocol*. 66

RM_DE *Routing Management Decision Element*. 48, 53, 63, 71, 73–75, 99, 102, 103, 124, 157

RSVP *Resource ReSerVation Protocol*. 43, 130, 131

RSVP-TE *Resource Reservation Protocol - Traffic Engineering*. 131, 133, 136

S-CSF *Serving-Call Session Control Function*. 160

SDH *Synchronous optical networking*. 19, 132

SLA Contrat de niveau de service (*Service Level Agreement*). 11, 99, 130

SNMP *Simple Network Management Protocol*. 23

SOA *Service Oriented Architecture*. 32

SOAP *Simple Object Access Protocol*. 124

SPF *Shortest Path First*. 67, 68

TBF *Time Between Failure*. 88, 117

TMN *Telecommunications Management Network*. 22

TTR *Time To Repair*. 88, 117

UAFAReS *Unified Architecture for Autonomic Fault-Management, Resilience and Survivability*. 12, 15, 47, 49–51, 64, 157, 159, 217

UDP *User Datagram Protocol*. 125, 127

VoIP *Voice over IP*. 19, 37, 97

WBEM *Web-Based Enterprise Management*. 23

WDM *Wavelength-division multiplexing*. 19

XML *Extensible Markup Language*. 24

XRO *Exclude Route Object*. 104, 136–138