



HAL
open science

Modélisation robuste du délai Internet et schémas de mesure intelligents pour l'automatisation des réseaux overlay

Maxime Mouchet

► **To cite this version:**

Maxime Mouchet. Modélisation robuste du délai Internet et schémas de mesure intelligents pour l'automatisation des réseaux overlay. Réseaux et télécommunications [cs.NI]. Ecole nationale supérieure Mines-Télécom Atlantique, 2020. Français. NNT : 2020IMTA0209 . tel-03666771

HAL Id: tel-03666771

<https://theses.hal.science/tel-03666771v1>

Submitted on 12 May 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THESE DE DOCTORAT DE

L'ÉCOLE NATIONALE SUPERIEURE MINES-TELECOM ATLANTIQUE
BRETAGNE PAYS DE LA LOIRE - IMT ATLANTIQUE

ECOLE DOCTORALE N° 601
*Mathématiques et Sciences et Technologies
de l'Information et de la Communication*
Spécialité : *Informatique*

Par

Maxime MOUCHET

Modélisation Robuste du Délai Internet et Schémas de Mesure Intelligents pour l'Automatisation des Réseaux Overlay

Thèse présentée et soutenue en visioconférence, le 27 novembre 2020

Unité de recherche : Lab-STICC

Thèse N° : 2020IMTA0209

Rapporteurs avant soutenance :

Mme. Hind CASTEL-TALEB

M. Kavé SALAMATIAN

Professeure

Professeur

Télécom SudParis

Université Savoie Mont Blanc

Composition du Jury :

Président : Mme. Cristel PELSSER

Examineurs : Mme. Hind CASTEL-TALEB

M. Kavé SALAMATIAN

M. Emmanuel HYON

Mme. Cristel PELSSER

M. Jean-Louis ROUGIER

Dir. de thèse : Mme. Sandrine VATON

Co-encadrant : M. Thierry CHONAVEL

Professeure

Professeure

Professeur

Maître de conférences

Professeure

Professeur

Professeure

Professeur

Université de Strasbourg

Télécom SudParis

Université Savoie Mont Blanc

Sorbonne Université

Université de Strasbourg

Télécom Paris

IMT Atlantique

IMT Atlantique

Invité :

M. Olivier BRUN

Directeur de recherche LAAS-CNRS

REMERCIEMENTS

Je me dois de remercier en premier mes encadrants, Sandrine et Thierry. Merci pour votre bienveillance, votre disponibilité et les connaissances que vous m'avez transmises. Je n'aurais pas imaginé faire de thèse dans de meilleures conditions.

Je remercie également les membres du jury, Mme Hind Castel-Taleb, professeure, M. Kavé Salamatian, professeur, M. Emmanuel Hyon, maître de conférences, Mme Cristel Pelsser, professeure, M. Jean-Louis Rougier, professeur, et M. Olivier Brun, directeur de recherche CNRS, d'avoir accepté d'évaluer mes travaux.

Il n'y aurait évidemment pas de thèse sans financement. Merci à l'Institut Mines-Télécom, à la Fondation Mines-Télécom et au Carnot Télécom & Société d'avoir financé ma thèse au travers du programme Futur & Ruptures. Merci également au RIPE NCC, à l'IFIP, au NL-ix et au GDR RSD d'avoir financé les déplacements aux conférences.

Durant ces trois années, j'ai eu la chance de pouvoir collaborer avec de nombreuses personnes, en France et à l'étranger : Bala, Emile, Isabel, Jasper, Marine, Martín, Olivier et Pablo. Merci à vous ! En particulier, merci à Pablo et Claudina de m'avoir accueilli un mois à l'*Universidad de la República* à Montévidéo, et à Robert et Emile pour m'avoir accueilli un mois au RIPE NCC à Amsterdam.

Si ma thèse a pu se dérouler sans encombre, c'est grâce à l'aide infaillible d'Armelle. Merci pour tout le temps passé sur les procédures administratives.

Merci à mes amis : Jean-Baptiste pour m'avoir convaincu de poursuivre en thèse ; Maxime pour tous les moments passés à refaire le monde ; Yann et Johann pour nos conversations sans fin ; David pour nos séances de running ; Stéphane, Noémie, Geoffrey et Vanessa pour nos week-ends marins bien mérités.

Merci à mes parents et à mes grands-parents de m'avoir intéressé à la science dès le plus jeune âge, et de m'avoir permis de poursuivre les études de mon choix dans les meilleures conditions.

Enfin, merci Fiona pour ta patience et ton soutien sans faille lors de cette dernière année de thèse.

Bien sûr, il m'est impossible de citer toutes les personnes m'ayant permis d'obtenir mon doctorat, tant nous sommes le produit de nos rencontres et de nos discussions tout au long de notre vie. Merci à toutes les personnes que j'ai pu rencontrer d'avoir partagé vos idées, vos connaissances et vos opinions et de m'avoir permis de devenir la personne que je suis.

C'est bien beau tout ça mais, Internet, ça se mange ?

— Kinder le chat

TABLE DES MATIÈRES

1	INTRODUCTION	1
1.1	Contexte	1
1.2	Contributions	1
1.3	Structure de la thèse	2
2	MODÉLISATION DU DÉLAI AVEC DES MODÈLES DE MARKOV CACHÉS NON PARAMÉTRIQUES	3
2.1	Problème	3
2.2	Modèles existants	4
2.2.1	Observations indépendantes	5
2.2.2	Modèles autorégressifs et réseaux de neurones	6
2.2.3	Modèles de Markov cachés	6
2.3	Modèles de mélange non paramétriques	7
2.3.1	Modèles de mélange	7
2.3.2	Processus de Dirichlet	14
2.3.3	Modèles de mélange à base de processus de Dirichlet	16
2.4	Modèles de Markov cachés non paramétriques	18
2.4.1	Modèles de Markov cachés	18
2.4.2	Processus de Dirichlet hiérarchiques	20
2.4.3	Modèles de Markov cachés à base de processus de Dirichlet hiérarchique	21
2.5	Modélisation du délai à l'aide du HDP-HMM	25
2.5.1	Un modèle d'observation non paramétrique	26
2.5.2	Implémentation du HDP-HMM à émissions DPMM	29
2.5.3	Considérations pratiques pour l'inférence	29
2.6	Validation de la modélisation HDP-HMM du délai	32
2.6.1	Validation sur un jeu de changements labélisés	32
2.6.2	Test du rapport de vraisemblance	35
2.6.3	Distribution des temps de séjour dans chaque état	37
2.7	Discussion	39
3	APPLICATION AU TRAITEMENT DE DONNÉES À GRANDE ÉCHELLE	40
3.1	La mesure de la performance d'Internet	40
3.1.1	Mesure du délai	41
3.1.2	Mesure des chemins	42
3.1.3	Infrastructures de mesure publiques	44
3.2	Caractérisation des données RIPE Atlas	45
3.2.1	Correspondance entre état caché et chemin réseau	46
3.2.2	Distribution du nombre d'états cachés	46
3.2.3	Durée moyenne et coefficient de variation	50
3.3	Segmentation intégrée à RIPE Atlas	51
3.3.1	Architecture	52

3.3.2	Terminaisons	53
3.4	Détection d'anomalies sur des infrastructures d'Internet	55
3.4.1	Couverture et choix des mesures	55
3.4.2	Incident du 13 mai 2015 à AMS-IX	57
3.4.3	Incident du 9 avril 2018 à DE-CIX	58
3.5	Autres types de mesures réseau	59
3.5.1	Mesures de congestion CAIDA MANIC	59
3.5.2	Mesures de temps de chargement Orange Web View	61
4	MÉTROLOGIE PARCIMONIEUSE AVEC DES PROCESSUS DE DÉCISION MARKOVIENS	62
4.1	Problème	62
4.2	Travaux connexes	64
4.3	Optimisation du routage	64
4.3.1	Filtrage et prédiction	65
4.4	Optimisation 1-pas en avant	67
4.4.1	Solution pour 1 chemin stochastique	68
4.4.2	Solution pour 2 chemins stochastiques	71
4.5	Optimisation à horizon de temps infini	74
4.5.1	Formulation MDP	74
4.5.2	Solution pour n chemins stochastiques	77
4.6	Résultats	78
4.6.1	1 chemin stochastique	78
4.6.2	2 chemins stochastiques	80
4.6.3	2 chemins entre des sondes RIPE Atlas	83
4.6.4	Simulation d'un overlay de routage	85
5	HEURISTIQUES POUR LA MÉTROLOGIE PARCIMONIEUSE	89
5.1	Structure du problème	89
5.1.1	Espace d'état	89
5.1.2	Transitions	91
5.1.3	Complexité	92
5.2	Optimisation à horizon de temps reculant	93
5.2.1	Complexité	94
5.3	Agrégation de chemins	95
5.4	Résultats	96
5.4.1	Données synthétiques	96
5.4.2	Données réelles	98
6	CONCLUSION	102
6.1	Travaux futurs	104
6.1.1	Apprentissage en ligne	104
6.1.2	Partitionnement des modèles	104
6.1.3	Détection d'incidents	105
	BIBLIOGRAPHIE	106
	PUBLICATIONS	118

TABLE DES FIGURES

Figure 2.1	Segmentation du RTT entre deux ancrs RIPE Atlas sous la modélisation HDP-HMM.	4
Figure 2.2	Segmentation du RTT entre deux ancrs RIPE Atlas avec un modèle de mélange gaussien : comparaison avec la modélisation HDP-HMM.	6
Figure 2.3	Représentation graphique d'un modèle de mélange bayésien.	8
Figure 2.4	Le processus <i>stick-breaking</i>	15
Figure 2.5	Représentation graphique d'un modèle de Markov caché bayésien.	18
Figure 2.6	Représentation graphique d'un modèle de Markov caché à base de processus de Dirichlet hiérarchique.	22
Figure 2.7	Segmentation du RTT entre deux ancrs RIPE Atlas sous la modélisation HDP-HMM, et densité des modèles de mélange inférés.	27
Figure 2.8	Segmentation d'une série temporelle de RTT avec des modèles de mélange et de HMMs paramétriques et non paramétriques.	28
Figure 2.9	Comparaison de la performance des implémentations du HDP-HMM.	30
Figure 2.10	Taux de rejet de H_0 en fonction de la probabilité de fausse alarme.	37
Figure 2.11	Diagramme quantile-quantile des temps de séjours normalisés observés contre la loi exponentielle de moyenne 1.	38
Figure 3.1	Composantes du délai dans les réseaux informatiques.	42
Figure 3.2	Exemple de traceroute de H_1 vers H_2 sans équilibrage de charge.	43
Figure 3.3	Exemple de traceroute de H_1 vers H_2 en présence d'équilibrage de charge.	43
Figure 3.4	Principe du système d'ancres Atlas avec 4 ancrs.	45
Figure 3.5	Exemple de correspondance entre état inféré et chemin réseau.	47
Figure 3.6	Correspondance entre état inféré et chemin IP dans des mesures RIPE Atlas.	47
Figure 3.7	Correspondance entre état inféré et chemin AS dans des mesures RIPE Atlas.	48
Figure 3.8	Exemples de segmentation de mesures vers les ancrs elle-même.	49

Figure 3.9	Distribution du nombre d'états inféré en fonction de la durée d'observations.	50
Figure 3.10	Distribution de la durée moyenne des états en fonction du coefficient de variation (écart type / moyenne).	52
Figure 3.11	Segmentation du délai sous la modélisation HDP-HMM entre deux ancres RIPE Atlas. Chaque couleur identifie un état.	54
Figure 3.12	Fréquence de changements le 13 mai 2015 pour les 20k paires qui ont vu AMS-IX dans leurs traceroutes la veille.	58
Figure 3.13	Fréquence de changements entre le 9 et le 10 avril 2018 pour les 60k paires qui ont vu DE-CIX Francfort dans leurs traceroutes la veille.	58
Figure 3.14	Méthode <i>Time Series Latency Probing</i> pour estimer le délai entre deux routeurs de bordure d'AS.	59
Figure 3.15	Segmentation d'une série temporelle de différence de RTT (lointain - proche) obtenue avec des sondes TSLP du projet CAIDA MANIC.	60
Figure 3.16	Densité de probabilité des états de précongestion et de congestion sur une série de différence temporelle de RTT segmentée.	60
Figure 3.17	Segmentation d'une série temporelle de temps de chargement de page obtenue à partir d'une sonde Web View.	61
Figure 4.1	Overlay de routage avec 6 nœuds et 15 arêtes (chemins directs). Un chemin alternatif est construit en transmettant les paquets par un ou plusieurs nœuds intermédiaires.	63
Figure 4.2	Configuration à deux nœuds.	67
Figure 4.3	Politique à seuil.	71
Figure 4.4	Schéma de principe de la métrologie parcimonieuse.	78
Figure 4.5	Simulation de quatre politiques : ne jamais mesurer, toujours mesurer, myope et MDP ($\rho = 0.99$).	79
Figure 4.6	Décisions de mesure pour la politique myope et MDP ($\rho = 0.01$) dans le cas de deux chemins stochastiques à deux états.	81
Figure 4.7	Simulation de la politique MDP ($\rho = 0.9$) entre deux sondes RIPE Atlas.	84
Figure 4.8	Nombre de mesures en fonction du coût de mesure.	88
Figure 4.9	Écart au délai de la politique toujours mesurer en fonction du nombre moyen de mesures (regroupés dans des intervalles de 0.1).	88

Figure 5.1	Espace d'état et entropie du filtre pour 2 chemins de 2 états. π représente la distribution stationnaire.	90
Figure 5.2	Espace d'état discret contenue dans l'espace d'état continu pour 2 chemins de 2 états. π représente la distribution stationnaire.	91
Figure 5.3	Comparaison de l'erreur relative moyenne des différentes politiques par rapport à la politique MDP optimale, pour deux chemins synthétiques et différentes valeurs de β (probabilité d'auto-transition). $c = 25$ et $\rho = 0.99$	97
Figure 5.4	Simulation de la politique à horizon de temps reculant sur un scénario à 4 chemins. En noir le délai après optimisation de la mesure et du routage.	101

LISTE DES TABLEAUX

Table 2.1	Méthodes de Monte-Carlo pour le HDP-HMM.	22
Table 2.2	Taxonomie des modèles à états cachés.	26
Table 2.3	Comparaison des implémentations du HDP-HMM.	30
Table 2.4	Comparaison du HDP-HMM avec des méthodes de détection de changement standards sur un jeu de données d'instant de changements labélisés. Méthodologie standard.	34
Table 2.5	Comparaison du HDP-HMM avec des méthodes de détection de changement standards sur un jeu de données d'instant de changements labélisés. Méthodologie alternative.	35
Table 3.1	Terminaisons de l'API RIPE Atlas <i>trends</i> . Les paramètres start et stop sont des dates au format ISO.	53
Table 3.2	Top 5 des systèmes autonomes et des points d'échanges présents dans les traceroutes du système d'ancres RIPE Atlas le 2 février 2020 entre ooh et ooh30.	56
Table 4.1	Décisions de routage et récompenses 1-pas en avant pour deux chemins markoviens, en fonction de la décision de mesure.	73

Table 4.2	Comparaison de la performance de quatre politiques : ne jamais mesurer, toujours mesurer, myope et MDP ($\rho = 0.01, 0.50, 0.99$). Statistiques calculées sur 100 trajectoires aléatoires.	80
Table 4.3	Conditions de mesure sur la « bordure » de l'espace d'état lorsque l'état du chemin 1 est parfaitement connu.	82
Table 4.4	Conditions de mesure sur la « bordure » de l'espace d'état lorsque l'état du chemin 2 est parfaitement connu.	82
Table 4.5	Évaluation de la politique MDP ($\rho = 0.9$) entre deux sondes RIPE Atlas. Comparaison avec un choix de chemin statique (direct ou indirect), et une mesure constante.	85
Table 4.6	Comparaison des politiques de routage statiques et optimisés. Valeurs médianes calculées sur 277 paires origine-destination.	87
Table 5.1	Comparaison de l'erreur relative moyenne des différentes politiques par rapport à la politique MDP optimale, pour trois chemins synthétiques et différentes valeurs de c et ρ	98
Table 5.2	Scénarios et temps de traitement pour l'heuristique d'aggrégations de chemins (H) et l'approche à horizon de temps reculant (RH, horizon = 3).	99
Table 5.3	Évaluation de l'heuristique d'aggrégations de chemins (H), de l'approche à horizon de temps reculant (RH, horizon = 3) et de la politique myope (MP) sur 4 scénarios avec des mesures NLNOG.	100

LISTE DES ALGORITHMES

ALGORITHME 2.1	Échantillonnage de Gibbs	10
ALGORITHME 2.2	Échantillonneur de Gibbs pour les modèles de mélange	13
ALGORITHME 2.3	Échantillonneur de Gibbs par blocs pour les modèles de Markov cachés finis	21
ALGORITHME 2.4	Échantillonneur de Gibbs par blocs pour les modèles de Markov cachés infinis	24

ACRONYMES

AMS-IX	Amsterdam Internet Exchange
API	Application Programming Interface
AS	Autonomous System
BGP	Border Gateway Protocol
DE-CIX	Deutscher Commercial Internet Exchange
CAIDA	Center for Applied Internet Data Analysis
DNS	Domain Name System
DP	Dirichlet Process
DPMM	Dirichlet Process Mixture Model
HMM	Hidden Markov Model
HDP-HMM	Hierarchical Dirichlet Process Hidden Markov Model
ICMP	Internet Control Message Protocol
IP	Internet Protocol
MDP	Markov Decision Process
NLNOG	Netherlands Network Operator Group
OWD	One-Way Delay
RH	Receding Horizon
RIPE	Réseaux IP Européens
RTT	Round Trip Time
SDN	Software-Defined Networking
TSLP	Time Series Latency Probing
TTL	Time To Live

NOTATIONS

En général

\bullet^T	la transposée d'un vecteur ou d'une matrice
e	le vecteur ligne $(1, \dots, 1)$
e_i	le vecteur ligne avec un 1 à la position i : $(0, \dots, \underset{i}{1}, \dots, 0)$
\mathbb{E}	une espérance
\mathbb{P}	une probabilité
p_θ	une densité de probabilité paramétrée par θ
A_n^k	le nombre d'arrangements de k éléments parmi n : $\frac{n!}{(n-k)!}$
C_n^k	le nombre de combinaisons de k éléments parmi n : $\frac{n!}{k!(n-k)!}$
$\{x : \phi(x)\}$	l'ensemble des valeurs de x pour lesquelles le prédicat $\phi(x)$ est vrai

Distributions de probabilité

$\text{Dir}(\boldsymbol{\pi} \mid \boldsymbol{\alpha})$	la loi de Dirichlet paramétrée par le vecteur $\boldsymbol{\alpha}$
$\text{DP}(\boldsymbol{\alpha}, H)$	le processus de Dirichlet de loi de base H et de concentration $\boldsymbol{\alpha}$
$\text{GEM}(\boldsymbol{\pi} \mid \boldsymbol{\alpha})$	la loi de Griffiths-Engen-McCloskey de concentration $\boldsymbol{\alpha}$
$\text{NIG}(\mu, \sigma^2 \mid \lambda)$	la loi normale-inverse-gamma avec $\lambda = (\mu_0, \nu, \alpha, \beta)$
$\mathcal{N}(y \mid \mu, \sigma^2)$	la loi normale de moyenne μ et de variance σ^2
$\text{Beta}(a, b)$	la loi bêta de paramètres de forme a et b
$\text{Gamma}(k, \theta)$	la loi gamma de paramètre de forme k et d'échelle θ
$B(\boldsymbol{\alpha})$	la fonction bêta : $B(\boldsymbol{\alpha}) = \Gamma(\alpha_1) \cdots \Gamma(\alpha_n) / \Gamma(\alpha_1 + \cdots + \alpha_n)$
$\Gamma(n)$	la fonction gamma : $\Gamma(n) = (n-1)!$

Modèles de mélange

K	nombre d'états cachés
T	nombre d'observations
y_t	observation au temps t
z_t	état caché associé à l'observation y_t
α	paramètre de concentration de la distribution ou du processus de Dirichlet
λ	paramètres de la loi a priori sur $\boldsymbol{\theta}_k$
$\boldsymbol{\theta}_k$	paramètres de la loi d'observation associée au k -ième état
$\boldsymbol{\pi}$	distribution de probabilité discrète sur les états cachés
n_k	nombre d'observations affectées à l'état k

$n_{-t,k}$ nombre d'observations affectées à l'état k en excluant y_t
 z_{-t} le vecteur z sans l'élément z_t

Modèles de Markov cachés

K nombre d'états cachés
 T nombre d'observations
 y_t observation au temps t
 z_t état caché associé à l'observation y_t
 α paramètre de concentration de la distribution ou du processus de Dirichlet
 β processus de Dirichlet racine
 γ paramètre de concentration du processus de Dirichlet racine
 κ paramètre *sticky* sur les auto transitions
 λ paramètres de la loi a priori sur θ_k
 θ_k paramètres de la loi d'observation associée au k -ième état
 π matrice de transition
 n_{ij} nombre de transitions de l'état i vers j dans la séquence \mathbf{z}

Métrologie parcimonieuse

$C(t)$ l'indice du chemin choisi au temps t
 $I^i(t)$ l'information sur le chemin i au temps t
 $L^i(t)$ le délai du chemin i au temps t
 l_j^i le délai associé à l'état j du chemin i
 $M^i(t)$ la décision de mesure pour le chemin i au temps t
 $P^{(i)}$ la matrice de transition du chemin i
 $\gamma_{t,t}^{(i)}$ le filtre du chemin i au temps t
 $\gamma_{t,t+u}^{(i)}$ le prédicteur u pas en avant du chemin i au temps t

INTRODUCTION

1.1 CONTEXTE

Cette thèse a pour motivation initiale des travaux sur l'optimisation de la mesure active (métrologie parcimonieuse) dans les overlays de routage sur Internet. Ces travaux ont été initiés avant le début de la thèse par une collaboration entre IMT Atlantique, le LAAS-CNRS, et l'*Universidad de la República* (UdelaR) en Uruguay. Une méthode basée sur les processus de décision markoviens est proposée pour réduire le nombre de mesures d'un chemin dont la longueur (le délai par exemple) peut être représentée par un modèle markovien. L'applicabilité de cette méthode sur des données réelles dépend de la capacité à apprendre un modèle markovien des chemins sur Internet. Les méthodes statistiques classiques d'estimation pour ces modèles sont compliquées à mettre en œuvre sur des données réelles. En effet, elles nécessitent la connaissance du nombre d'états cachés de la série temporelle, et elles sont peu robustes aux observations bruitées et aberrantes souvent présentes dans les données. De plus, la méthode proposée souffre d'une complexité exponentielle avec le nombre de chemins à mesurer, ce qui limite son applicabilité sur un grand nombre de chemins.

Cette thèse comporte donc deux aspects : la modélisation markovienne du délai sur Internet, et l'optimisation de la mesure du délai dans les overlays de routage.

1.2 CONTRIBUTIONS

Les contributions de cette thèse sont :

- La formulation et la résolution du problème de métrologie parcimonieuse dans les overlays de routage, en utilisant le formalisme des processus de décision markoviens. La méthode est validée sur des données synthétiques et réelles. Des algorithmes de résolution approximatifs sont proposés afin de supporter un grand nombre de chemins.
- L'application des modèles de Markov cachés infinis aux mesures de délai sur Internet. Ce modèle permet la prédiction et la segmentation des séries temporelles de délais sur Internet. Ce modèle est validé sur des données réelles, et on montre sa capacité à mettre en évidence des incidents sur Internet uniquement à partir de mesures du délai. De plus, ce modèle est accessible

au travers d'une API web publique, permettant la segmentation de séries temporelles de mesures de délai.

1.3 STRUCTURE DE LA THÈSE

Le [chapitre 2](#) traite de la modélisation du délai à l'aide de modèles de Markov cachés. Ce chapitre se concentre sur la présentation détaillée du formalisme mathématique du modèle et fournit des résultats de validation qui ne nécessitent pas d'introduire de concepts réseau. Le [chapitre 3](#) traite des applications du modèle pour l'analyse de grands jeux de données de mesure réseaux. C'est un chapitre plus appliqué qui se concentre sur les applications à l'analyse des données réseau. Le [chapitre 4](#) introduit et propose une solution au problème de surveillance parcimonieuse des overlay de routage. Il se base sur le modèle de Markov caché infini introduit précédemment. Enfin, le [chapitre 5](#) propose des algorithmes d'optimisation afin de résoudre le problème de surveillance parcimonieuse dans de grands réseaux.

MODÉLISATION DU DÉLAI AVEC DES MODÈLES DE MARKOV CACHÉS NON PARAMÉTRIQUES

2.1 PROBLÈME

L'administration des réseaux informatiques a traditionnellement été confiée aux humains. Il s'agit cependant d'une tâche coûteuse en ressources, sujette aux erreurs et difficile à adapter aux changements.

Les dix dernières années ont vu apparaître des technologies facilitant le déploiement et la gestion de réseaux à grande échelle, telles que les réseaux définis par logiciel (*software-defined networking*), ou la virtualisation des fonctions réseau (*network function virtualization*) [60]. Bien que ces technologies permettent d'automatiser le contrôle des réseaux, elles ne sont pas suffisantes pour les rendre autonomes.

Les réseaux autonomes (*driverless networks*) [19, 20, 59] exploitent les techniques récentes d'apprentissage automatique pour extraire les informations pertinentes dans les mesures réseau, et automatiser la prise de décision, sans intervention humaine. De nombreux problèmes sont à résoudre, tels que la modélisation statistique, la prédiction et la détection de changements et d'anomalies. Les méthodes proposées doivent être robustes à la diversité et à la qualité variable des données opérationnelles, et pouvoir passer à l'échelle en matière de ressources informatiques. De plus, les résultats doivent être fiables et précis afin d'automatiser la prise de décision pour la gestion, la sécurité et la résilience des réseaux.

Dans ce chapitre nous proposons d'utiliser les modèles de Markov cachés à base de processus de Dirichlet hiérarchique (*Hierarchical Dirichlet Process Hidden Markov Model* ou *HDP-HMM*) pour la modélisation du délai dans les réseaux informatiques, et en particulier sur Internet.

Le délai est une métrique de performance importante, car il est facile à mesurer et est étroitement corrélé avec la performance applicative. Des mesures actuelles et historiques du délai sont disponibles en accès libre sur des plateformes de mesure d'Internet telles que RIPE Atlas [98] ou CAIDA Ark [50].

La modélisation HDP-HMM offre de bonnes performances pour des tâches de détection de changements significatifs du délai, de détection d'anomalies et de prédiction du délai, comme nous allons le montrer par une analyse de données à grande échelle. Elle supporte une grande diversité de distributions du délai, et est robuste aux observations aberrantes et manquantes. Elle passe également à l'échelle, avec une complexité linéaire avec le nombre d'observations.

De plus, comme nous le verrons dans le chapitre suivant, ce modèle a été implémenté dans une infrastructure de mesure d’Internet à grande échelle, et est accessible publiquement à travers une API (*Application Programming Interface*) HTTP.

Dans la [section 2.2](#) nous commençons par une revue des modèles existants pour modéliser le délai, et nous justifions la nécessité d’un nouveau modèle. Dans la [section 2.3](#) nous introduisons les modèles de mélange, ainsi que les modèles bayésiens non paramétriques, qui sont un prérequis pour aborder le HDP-HMM. Dans la [section 2.4](#) nous introduisons les modèles de Markov cachés, et leur variante non paramétrique. Enfin, dans la [section 2.5](#) nous détaillons l’utilisation du HDP-HMM pour les mesures de délai, et nous validons le modèle sur un jeu de données d’instantés de changements labélisé, et avec un test statistique.

2.2 MODÈLES EXISTANTS

La modélisation du délai dans les réseaux informatiques est un problème abondamment étudié dans la littérature. Cependant les modèles obtenus sont fortement dépendants du réseau et du jeu de données considéré. Les travaux les plus anciens considèrent le plus souvent des réseaux locaux ou des réseaux universitaires, dont la taille et la complexité sont moindres. Sur Internet les paquets doivent traverser plusieurs systèmes autonomes, qui ont chacun leur politique de routage interne et peuvent mettre en place de l’équilibrage de charge.

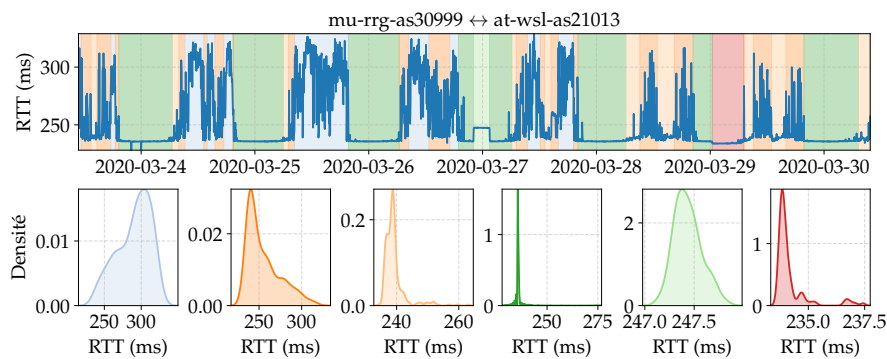


FIGURE 2.1 – Segmentation du RTT entre deux ancres RIPE Atlas sous la modélisation HDP-HMM. Chaque couleur représente un état caché.

La [figure 2.1](#) met en avant certains des motifs caractéristiques des mesures de délai sur Internet. La segmentation est réalisée avec la méthode décrite dans ce chapitre. Chaque couleur indique un « état réseau » inféré par notre modèle. Dans les deux états verts, le délai varie peu, et semble être distribué selon une loi normale de faible variance. Le passage entre ces deux états peut s’expliquer par un changement

de routage sur le chemin des paquets. Les états jaune, orange, et bleu correspondent peut-être à une augmentation journalière du niveau de trafic. Quand le trafic augmente, les files d'attente des routeurs se remplissent, et le délai augmente. Quand la file d'attente approche de sa capacité maximale, les paquets sont détruits. Ce phénomène s'observe souvent au niveau des liens d'appairage (*peering*) entre les opérateurs.

On remarque que la distribution du délai est différente suivant l'état. Dans certains cas le délai semble suivre une loi normale, dans d'autres une loi exponentielle, et parfois une loi plus complexe. On remarque également que les distributions du délai de plusieurs états peuvent se recouper, par exemple pour les états bleu et orange.

Nous recherchons un modèle qui soit capable de bien distinguer les différentes distributions de délai présentes dans les mesures. Cela nous permettra d'utiliser le modèle pour détecter les changements significatifs de délai dans des mesures, et donc des problèmes potentiels sur le réseau ([chapitre 3](#)). De plus nous recherchons un modèle qui prend en compte la dynamique de transition entre ces distributions, afin de pouvoir prédire le délai futur. Cela nous permettra de définir des méthodes de métrologie parcimonieuse qui limitent le coût de mesure ([chapitre 4](#)).

2.2.1 Observations indépendantes

Les modèles les plus simples supposent que les observations successives sont indépendantes entre elles. Ces modèles peuvent être utilisés pour la détection d'anomalies dans les valeurs du délai. Cependant, ils ne peuvent pas être utilisés pour prédire le délai ou identifier des motifs récurrents puisqu'ils ne prennent pas en compte les dépendances temporelles.

Dans [112] une loi de Pareto est utilisée pour modéliser le délai sur un réseau expérimental comportant 6 routeurs. Le trafic est généré artificiellement. Le modèle est validé en comparant les statistiques du premier et second ordre avec celles des données. Dans [47] un mélange de lois de Weibull est utilisé pour modéliser le délai observé sur Internet sur des périodes de 24 heures. Les mesures sont réalisées par le RIPE NCC à partir de 35 hôtes réparties dans le monde. Dans [85] un modèle de mélange gaussien est utilisé pour modéliser le délai observé sur Internet sur des périodes de 24 heures. Les mesures sont réalisées depuis une université vers des serveurs répartis dans le monde. Dans [36] un modèle de mélange log-normal non paramétrique est utilisé pour modéliser le délai d'un ensemble d'hôtes au niveau d'un lien d'interconnexion. Le trafic est mesuré pendant 15 minutes consécutives, tous les jours, sur plusieurs années.

Les modèles de mélange fournissent une segmentation (ou *clustering*) des données, mais à l'inverse des modèles de Markov cachés ils

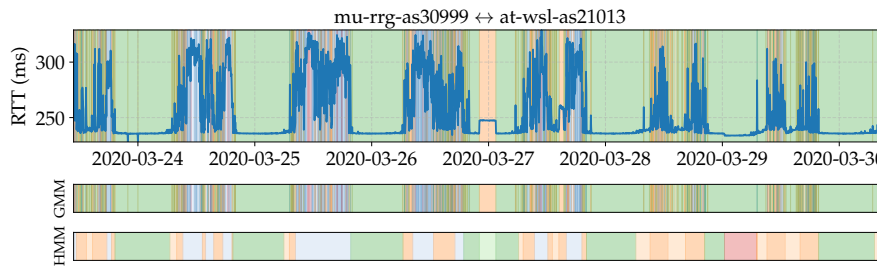


FIGURE 2.2 – Segmentation du RTT entre deux ancres RIPE Atlas avec un modèle de mélange gaussien (GMM) : comparaison avec la modélisation HDP-HMM. Chaque couleur représente un état caché.

ne prennent pas en compte la dépendance temporelle entre les états. Comme le montre la [figure 2.2](#), cela peut créer une « sur segmentation » dans les états bruités.

2.2.2 Modèles autorégressifs et réseaux de neurones

Les modèles autorégressifs et les réseaux de neurones permettent de modéliser des séries temporelles en prenant en compte la dépendance temporelle entre les observations. Ils peuvent être utilisés pour la prédiction de délai à court terme (de la seconde à plusieurs minutes). Prédire le délai permet d’améliorer la performance des applications temps-réel (*streaming* vidéo, voix sur IP, télérobotique, etc.).

Des modèles autorégressifs sont proposés dans [110] et [54]. Dans [110] le délai est mesuré sur une période de 24 heures depuis une université américaine, vers 2 destinations internes, et 6 destinations sur Internet. Dans [54] le délai est mesuré depuis l’Iran vers 3 destinations nationales, et une destination internationale.

Des réseaux de neurones récurrents sont proposés dans [14] et [30]. Dans [14] le délai est mesuré toutes les minutes sur une période de 24 heures, entre une université tunisienne et une université française. Le délai un pas en avant est prédit à partir des 3 dernières mesures. Dans [30] le délai est mesuré passivement sur une période de 1 mois, entre deux nœuds sur Internet. Le délai un pas en avant est prédit à partir de 20 dernières mesures. Le pas de temps entre les mesures n’est pas précisé.

Ces modèles ne fournissent pas de segmentation des données, et dans le cas des réseaux de neurones, leurs paramètres peuvent être difficiles à interpréter.

2.2.3 Modèles de Markov cachés

Les modèles de Markov cachés (*Hidden Markov Models* ou *HMM*) sont des modèles de séries temporelles à temps discret, qui peuvent

modéliser différentes distributions d'observations, et les transitions entre ces distributions. Ils généralisent les modèles de mélange, en introduisant une dépendance markovienne entre les états successifs.

Dans [84] un HMM à temps discret est utilisé pour modéliser la perte de paquets, tandis que dans [109] un HMM à temps continu est utilisé pour modéliser la perte de paquets et le délai. Dans [26], un HMM est utilisé pour modéliser les temps inter paquets, et la taille des paquets.

Les HMMs ont peu de paramètres et ils sont facilement interprétables (probabilités de transition entre les états, moyennes et variances). Comme nous le verrons dans la section suivante, il est difficile de déterminer le nombre d'états cachés, lorsque celui-ci est inconnu a priori, avec les HMMs standards. Pour remédier à ce problème, nous utilisons un HMM dit « bayésien non paramétrique » où le nombre d'états cachés est inféré directement à partir des données.

2.3 MODÈLES DE MÉLANGE NON PARAMÉTRIQUES

L'objectif de ce chapitre est d'introduire le HDP-HMM, ou « modèle de Markov caché infini ». Le HDP-HMM peut être vu comme une généralisation des modèles de Markov cachés, ces derniers étant eux-mêmes une généralisation des modèles de mélange. Nous proposons donc d'introduire le HDP-HMM progressivement, en commençant par introduire les modèles de mélange finis (MM) et infinis (DPMM) dans cette section, puis les modèles de Markov caché finis (HMM) et infinis (HDP-HMM) dans la section suivante (section 2.4).

Dans cette section nous commençons par introduire les modèles de mélange et l'estimation de leurs paramètres, de manière générale, puis dans le cadre bayésien. Nous introduisons ensuite les processus de Dirichlet, et leur utilisation dans des modèles de mélange infinis. Les modèles infinis (DPMM et HDP-HMM) étant des modèles dits « bayésiens », nous proposons de suivre dès le début un traitement bayésien des modèles finis (MM et HMM) pour permettre une construction progressive du HDP-HMM.

2.3.1 Modèles de mélange

Considérons le modèle graphique de la figure 2.3. Les nœuds de ce graphe représentent les variables aléatoires et les hyperparamètres de notre modèle, tandis que les arêtes représentent les dépendances entre ces variables. Les nœuds colorés en gris représentent les variables observées, en opposition aux variables cachées (ou *latentes*). Enfin, les plaques encadrant les nœuds indiquent le nombre de fois que ces nœuds sont répétés.

Ce modèle graphique représente un modèle de mélange : les observations y_t sont distribuées selon une loi de probabilité qui dépend

Un modèle graphique représenté par un graphe orienté acyclique est aussi appelé réseau bayésien. Dans le cas d'un graphe non orienté, on parle de champ aléatoire de Markov.

d'un état caché $z_t \in \{1, \dots, K\}$, et d'un vecteur de paramètres θ_k . Une distribution de probabilité discrète π assigne une probabilité à chaque état caché. On remarque que les états cachés successifs et les observations successives sont indépendants entre eux : connaître y_t ou z_t n'apporte aucune information sur y_{t+1} ou z_{t+1} .

Ce modèle est également dit *bayésien* car le vecteur de probabilité π et les paramètres θ_k sont eux-mêmes des variables aléatoires. Les distributions de probabilité qui leur sont associées sont appelées distributions *a priori*, et leurs paramètres (ici α et λ) sont appelés *hyperparamètres*.

Remarquons que les paramètres d'un modèle statistique ne sont pas nécessairement des variables aléatoires. Cependant, l'introduction des notions de distribution a priori et d'hyperparamètre dans cette section nous permettra de faciliter la présentation des modèles plus complexes dans les sections suivantes.

À partir d'un modèle graphique, nous pouvons réaliser deux tâches : générer des observations à partir des paramètres du modèle, et inférer les paramètres du modèle à partir des observations.

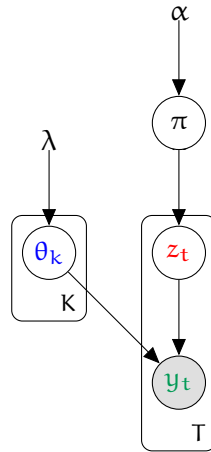


FIGURE 2.3 – Représentation graphique d'un modèle de mélange bayésien.

2.3.1.1 Échantillonnage

Nous pouvons générer des échantillons de notre modèle grâce à l'échantillonnage ancestral [17, p. 365], qui consiste à parcourir les variables par ordre topologique : une variable j est échantillonnée après une variable i si i est un parent de j .

Supposons que nous ne connaissions que les hyperparamètres α et λ . Nous commençons par échantillonner π et $(\theta_k)_{k=1:K}$, qui n'ont aucune variable pour parent, d'après leurs distributions a priori $p(\pi)$ et $p(\theta)$. Nous échantillonnons ensuite z_t à partir de sa loi conditionnelle $p(z_t | \pi)$, puis y_t à partir de $p(y_t | z_t, \theta_{z_t})$. Cette opération est répétée pour t variant de 1 à T afin d'obtenir un vecteur d'observations $\mathbf{y} = (y_1, \dots, y_T)$.

On remarque qu'un tel modèle graphique spécifie le processus par lequel les observations sont générées : il s'agit d'un modèle *génératif*. Dans certains cas cela permet d'attribuer un sens physique aux variables cachées, comme nous le verrons dans la [section 2.5](#).

2.3.1.2 Inférence

L'inférence peut être vue comme le processus inverse de l'échantillonnage : on cherche à estimer (ou *inférer*) les paramètres du modèle à partir des observations. L'approche standard en statistique consiste à formuler le problème comme un problème d'optimisation, où l'on cherche les paramètres qui maximisent une fonction objectif, typiquement, la vraisemblance des paramètres au vu des observations. Dans notre cas, nous chercherions les $(\boldsymbol{\pi}, \boldsymbol{\theta})$ qui maximisent $p(\mathbf{y} \mid \boldsymbol{\pi}, \boldsymbol{\theta})$. On parle alors d'estimation *au sens du maximum de vraisemblance*.

Dans certains cas il est possible de calculer directement les paramètres qui maximisent globalement la fonction de vraisemblance. Par exemple, en cherchant les valeurs pour lesquelles la dérivée de la fonction de vraisemblance est nulle. Dans le cas de modèles comportant des variables cachées, tel que les modèles de mélange, le calcul de la fonction de vraisemblance requiert de calculer une somme sur toutes les combinaisons d'états cachés possibles : $p(\mathbf{y} \mid \boldsymbol{\pi}, \boldsymbol{\theta}) = \sum_{\mathbf{z}} p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\pi}, \boldsymbol{\theta})$. Lorsque le nombre d'observations est grand, le nombre de combinaisons pour le vecteur \mathbf{z} explose et le calcul de cette somme devient impossible. Dans ce cas, l'algorithme EM (*Expectation-Maximization*) [28] permet d'obtenir un maximum local de la fonction de vraisemblance avec un coût de calcul bien plus faible.

L'algorithme EM est un algorithme itératif qui comporte deux étapes : l'étape E (*Expectation*) et l'étape M (*Maximization*). Dans l'étape E, on calcule l'espérance de la vraisemblance, conditionnellement à la dernière estimation des paramètres et aux observations. Dans l'étape M, on cherche les paramètres qui maximisent l'espérance calculée précédemment. La vraisemblance des observations augmente à chaque itération de l'algorithme EM, jusqu'à la convergence vers un maximum local de la fonction de vraisemblance. Il n'y a pas de garantie de converger vers un maximum global de la fonction de vraisemblance. L'algorithme EM est donc dépendant de la valeur initiale des paramètres, et il est souvent nécessaire de réaliser plusieurs estimations avec des valeurs initiales différentes pour avoir plus de chances de trouver un maximum proche du maximum global de la fonction de vraisemblance.

Dans la suite de ce chapitre, nous suivons une approche alternative pour l'inférence, dite *inférence bayésienne*. Celle-ci nous permettra en particulier de dériver des modèles avec un espace de paramètres de dimension infinie, où le nombre de paramètres n'est pas fixé a priori, mais inféré directement à partir des données.

Dans l'inférence bayésienne, on ne cherche pas à estimer directement une valeur particulière pour les paramètres du modèle, mais plutôt une distribution de probabilité sur ces paramètres. Nous pouvons exprimer cela avec le théorème de Bayes, en notant Φ les paramètres du modèle, et \mathbf{y} les observations :

$$p(\Phi | \mathbf{y}) = \frac{p(\mathbf{y} | \Phi)p(\Phi)}{p(\mathbf{y})}. \quad (2.1)$$

$p(\Phi | \mathbf{y})$ est la distribution a posteriori des paramètres, $p(\Phi)$ la distribution a priori, $p(\mathbf{y} | \Phi)$ la fonction de vraisemblance, et $p(\mathbf{y})$ une constante de normalisation.

Il existe parfois une forme close de cette expression. C'est notamment le cas lorsque la distribution a priori est *conjuguée*. Une distribution a priori est conjuguée si la distribution a posteriori est de la même famille que la distribution a priori. Lorsqu'il n'existe pas de forme close, ou que son évaluation est trop complexe, des méthodes de simulation stochastique telles que les méthodes de Monte-Carlo par chaînes de Markov (*Markov Chain Monte Carlo* ou *MCMC*) [81], ou des méthodes d'approximation telles que l'inférence variationnelle [17] peuvent être utilisées.

Dans ce chapitre, nous considérons l'échantillonnage de Gibbs [81, ch. 10], une méthode MCMC, qui permet de générer facilement des échantillons selon une loi multidimensionnelle. Cela nous est utile puisque, dans le cas d'un modèle de mélange, nous souhaitons estimer la distribution a posteriori $p(\Phi | \mathbf{y})$ avec $\Phi = (\theta, \pi, z)$. L'échantillonnage de Gibbs consiste à simuler séquentiellement les valeurs de ϕ_1, \dots, ϕ_M conditionnellement aux observations $y_{1:T}$ et aux valeurs simulées courantes pour les autres paramètres ([algorithme 2.1](#)).

L'échantillonnage de Gibbs demande seulement d'évaluer la loi de chaque paramètre conditionnellement aux autres paramètres, sans avoir besoin d'évaluer la loi jointe de tous les paramètres.

Algorithme 2.1 : Échantillonnage de Gibbs

```

initialiser  $\phi_1^{(0)}, \dots, \phi_M^{(0)}$ ;
pour  $n \leftarrow 1$  à  $N$ 
    échantillonner  $\phi_1^{(n)} \sim p(\phi_1 | \phi_2^{(n-1)}, \phi_3^{(n-1)}, \dots, \phi_M^{(n-1)}, \mathbf{y})$ ;
    échantillonner  $\phi_2^{(n)} \sim p(\phi_2 | \phi_1^{(n)}, \phi_3^{(n-1)}, \dots, \phi_M^{(n-1)}, \mathbf{y})$ ;
    ...;
    échantillonner  $\phi_M^{(n)} \sim p(\phi_M | \phi_1^{(n)}, \phi_2^{(n)}, \dots, \phi_{M-1}^{(n)}, \mathbf{y})$ ;
fin

```

2.3.1.3 Exemple : mélange gaussien

Considérons le cas d'un modèle de mélange gaussien, où les observations, conditionnellement aux états cachés, suivent une loi normale.

Nous souhaitons estimer les paramètres du modèle à partir d'un ensemble d'observations. Comme il s'agit d'un modèle avec plusieurs paramètres, nous proposons d'utiliser l'échantillonnage de Gibbs afin d'échantillonner séquentiellement les paramètres. Nous utiliserons la même méthode pour les modèles plus complexes des sections suivantes. Il sera ainsi plus facile de comparer les algorithmes d'échantillonnage et les équations associées entre les modèles.

Pour utiliser l'échantillonnage de Gibbs, nous devons définir pour chaque paramètre sa loi a priori, et sa loi a posteriori conditionnellement aux observations et aux autres paramètres. Nous commençons par les paramètres de la loi d'observation, puis nous traitons le vecteur de probabilité sur les états cachés. Finalement, nous donnons l'algorithme d'échantillonnage complet.

Dans un modèle de mélange gaussien, on associe à chaque état caché k une loi normale de moyenne μ_k et de variance σ_k^2 . On note $\theta_k = (\mu_k, \sigma_k^2)$ et $\theta = (\theta_k)_k$. L'expression de la loi conditionnelle des observations est donnée par :

$$p(y_t | z_t = k, \theta_k) = \mathcal{N}(y_t | \mu_k, \sigma_k^2) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left\{-\frac{1}{2} \frac{(y_t - \mu_k)^2}{\sigma_k^2}\right\} \quad (2.2)$$

Afin de construire un modèle bayésien, nous devons placer une loi a priori sur les paramètres $(\mu_k, \sigma_k^2)_k$. L'a priori conjugué pour la moyenne et la variance d'une loi normale est la loi normale-inverse-gamma :

$$p(\theta_k) = \frac{\sqrt{\nu}}{\sqrt{2\pi\sigma_k^2}} \frac{\beta^\alpha}{\Gamma(\alpha)} \left(\frac{1}{\sigma_k^2}\right)^{\alpha+1} \exp\left(-\frac{2\beta + \nu(\mu_k - \mu_0)^2}{2\sigma_k^2}\right) \quad (2.3)$$

où $\Gamma(n) = (n-1)!$. Elle est paramétrée par $\lambda = (\mu_0, \nu, \alpha, \beta)$, où $\mu_0 \in \mathbb{R}$ et $\nu, \alpha, \beta \in \mathbb{R}^+$. La moyenne et la variance de μ_k et σ_k^2 sont données par :

$$\begin{aligned} \mathbb{E}[\mu_k] &= \mu_0 & \text{Var}[\mu_k] &= \frac{\beta}{(\alpha-1)\nu} \\ \mathbb{E}[\sigma_k^2] &= \frac{\beta}{\alpha+1} & \text{Var}[\sigma_k^2] &= \frac{\beta^2}{(\alpha-1)^2(\alpha-2)} \end{aligned}$$

On remarque que μ_0 contrôle μ_k , et β/α contrôle σ_k^2 . α et ν représentent un nombre d'observations virtuelles, plus ce nombre est grand, plus la variance des paramètres μ_k et σ_k^2 est faible. En l'absence de connaissance a priori sur la valeur des paramètres, on choisira une loi a priori peu informative, avec une grande variance, afin de ne pas biaiser la loi a posteriori. Les hyperparamètres peuvent être définis à partir des observations. Par exemple, μ_0 et β peuvent être définis à partir de la moyenne et de la variance des observations [79].

Comme $p(\theta_k)$ est un a priori conjugué, la loi a posteriori $p(\theta_k | z, y)$ est à nouveau une loi normale-inverse-gamma :

$$p(\theta_k | z, y) \propto p(\{y_t : z_t = k\} | \theta_k) p(\theta_k) = \mathbf{NIG}(\theta_k | \mu'_0, \nu', \alpha', \beta') \quad (2.4)$$

où le symbole \propto représente la proportionnalité. Conditionnellement aux états cachés, les hyperparamètres a posteriori $(\mu'_0, \nu', \alpha', \beta')$ ne dépendent que des observations associées à l'état k :

$$\begin{aligned} \mu'_0 &= \frac{\nu\mu_0 + n_k\bar{y}}{\nu + n_k} & \alpha' &= \alpha + \frac{n_k}{2} \\ \nu' &= \nu + n_k & \beta' &= \beta + \frac{1}{2}\tilde{y} + \frac{\nu n_k}{\nu + n_k} \frac{(\bar{y} - \mu_0)^2}{2} \\ \bar{y} &= \frac{1}{n_k} \sum_{t:z_t=k} y_t & \tilde{y} &= \sum_{t:z_t=k} (y_t - \bar{y})^2 \end{aligned}$$

où $n_k = \#\{z_t : z_t = k\}$ est le nombre d'observations affectées à l'état k .

Nous venons de déterminer la loi a posteriori des paramètres θ_k . Nous devons maintenant déterminer la loi a posteriori des états cachés z_t . Les états cachés sont indépendants entre eux, et distribués selon le vecteur π :

$$p(z_t = k | \pi) = \pi_k \quad (2.5)$$

et

$$p(z | \pi) = \prod_{t=1}^T \pi_{z_t} = \prod_{k=1}^K \pi_k^{n_k}. \quad (2.6)$$

L'a priori conjugué pour un vecteur de probabilité de dimension K , est la loi de Dirichlet d'ordre K :

$$p(\pi) = \mathbf{Dir}(\pi | \alpha) = \frac{\Gamma(\alpha_0)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K \pi_k^{\alpha_k - 1} \quad (2.7)$$

avec $\alpha_0 = \sum_{k=1}^K \alpha_k$. Elle est paramétrée par un vecteur de poids positifs α . La moyenne et la variance de π_k sont données par :

$$\mathbb{E}[\pi_k] = \frac{\alpha_k}{\alpha_0} \quad \text{Var}[\pi_k] = \frac{\alpha_k(\alpha_0 - \alpha_k)}{\alpha_0^2(\alpha_0 + 1)}$$

Ainsi, plus α_k est grand par rapport aux autres poids, plus un poids important sera affecté à π_k . En l'absence de connaissances a priori, on peut choisir un a priori dit « symétrique » avec $\alpha_1 = \dots = \alpha_K$. Dans ce cas, une valeur de α inférieure à 1 privilégiera des vecteurs π_k parcimonieux, avec beaucoup d'éléments proches de 0. À l'inverse, une valeur de α supérieure à 1 privilégiera des vecteurs homogènes, avec une valeur semblable pour tous les éléments.

Il est possible d'exprimer $p(z)$ directement en fonction de α en intégrant le vecteur π dans l'équation 2.6 [79] :

$$\begin{aligned} p(z) &= \int_{\pi} p(z | \pi) p(\pi) d\pi \\ &= \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_0 + T)} \prod_{k=1}^K \frac{\Gamma(n_k + \alpha_k)}{\Gamma(\alpha_k)}. \end{aligned} \quad (2.8)$$

On en déduit la loi a priori d'un état caché conditionnellement aux autres états [79] :

$$p(z_t = k | \mathbf{z}_{-t}) = \frac{p(\mathbf{z})}{p(\mathbf{z}_{-t})} = \frac{n_{-t,k} + \alpha_k}{\alpha_0 + T - 1} \quad (2.9)$$

où $\mathbf{z}_{-t} = (z_1, \dots, z_{t-1}, z_{t+1}, \dots, z_T)$ et $n_{-t,k} = \#\{z_\tau : z_\tau = k, \tau \neq t\}$ est le nombre d'observations affectées à l'état k en excluant y_t .

Finalement, la loi a posteriori d'un état caché conditionnellement aux autres paramètres et aux observations est donnée par :

$$\begin{aligned} p(z_t = k | \mathbf{y}, \mathbf{z}_{-t}, \boldsymbol{\theta}) &= p(z_t = k | y_t, \mathbf{y}_{\neq t}, \mathbf{z}_{-t}, \boldsymbol{\theta}) \\ &\propto p(z_t = k, y_t, \mathbf{z}_{-t}, \boldsymbol{\theta}) \\ &\propto p(y_t | z_t = k, \mathbf{z}_{\neq t}, \boldsymbol{\theta}_k, \boldsymbol{\theta}_{\neq k}) p(z_t = k, \mathbf{z}_{-t}, \boldsymbol{\theta}) \\ &\propto \underbrace{p(y_t | z_t = k, \boldsymbol{\theta}_k)}_{\text{équation 2.2}} \underbrace{p(z_t = k | \mathbf{z}_{-t}, \boldsymbol{\theta})}_{\text{équation 2.9}}. \end{aligned} \quad (2.10)$$

La constante de normalisation est donnée par $\sum_{k=1}^K p(z_t = k | \cdot) = 1$.

Algorithme 2.2 : Échantillonneur de Gibbs pour les modèles de mélange

Entrées : $\boldsymbol{\theta}^{(n-1)}$ et $\mathbf{z}^{(n-1)}$

pour $t \leftarrow 1$ à T

échantillonner $z_t^{(n)} \sim p(z_t | \mathbf{y}, z_1^{(n)}, \dots, z_{t-1}^{(n)}, z_{t+1}^{(n-1)}, \dots, z_T^{(n-1)}, \boldsymbol{\theta}^{(n-1)})$ (équation 2.10) ;

fin

pour $k \leftarrow 1$ à K

échantillonner $\boldsymbol{\theta}_k^{(n)} \sim p(\boldsymbol{\theta}_k | \mathbf{y}, \mathbf{z}^{(n)})$ (équation 2.4) ;

fin

L'algorithme 2.2 résume la procédure d'échantillonnage. Remarquons qu'il n'est pas nécessaire d'échantillonner $\boldsymbol{\pi}$ puisque nous l'avons intégré dans \mathbf{z} dans l'équation 2.8. Dans ce cas, on parle d'échantillonneur de Gibbs « intégré » (*collapsed Gibbs sampler*). Une procédure similaire est possible pour $\boldsymbol{\theta}$ lorsque $p(y_t | z_t = k, \boldsymbol{\theta}_k)$ et $p(\boldsymbol{\theta}_k)$ sont conjugués. Il suffit alors d'échantillonner \mathbf{z} uniquement.

Dans cet exemple, nous avons considéré un modèle où les hyperparamètres (α et λ) sont fixés en avance. Dans le cadre bayésien, il est tout à fait possible de placer une loi a priori sur ces paramètres, et de les échantillonner dans l'échantillonneur de Gibbs. Cela permet de créer des modèles très souples capables de s'adapter à une grande variété de données.

L'échantillonnage de Gibbs, au contraire de l'algorithme EM, ne souffre pas de problème de convergence vers des minima non globaux

de la fonction objectif, en fonction des valeurs initiales. Cependant, afin d'accélérer la convergence vers la vraie distribution a posteriori, il est souhaitable d'initialiser les paramètres à des valeurs proches de cette distribution.

Une stratégie d'initialisation possible pour les modèles de mélange gaussien, consiste à utiliser l'algorithme k-means++ [8] pour obtenir un ensemble de points distribués aléatoirement dans le jeu de données, puis d'utiliser cet ensemble de points comme initialisation de l'algorithme k-means [17, ch. 9]. L'algorithme k-means réalise une segmentation des données en minimisant la variance à l'intérieur des groupes. Les groupes ainsi obtenus peuvent être utilisés comme initialisation pour z , et θ peut être estimé à partir de z .

2.3.2 Processus de Dirichlet

Les modèles de mélange considérés jusqu'à présent sont des modèles paramétriques, où le nombre de paramètres est fixé a priori. Cela rend leur utilisation difficile pour les mesures réseau puisque, souvent, le nombre d'états cachés n'est pas connu en avance.

Une approche classique pour choisir l'ordre d'un modèle (par exemple, le nombre d'états cachés dans un mélange) consiste à privilégier celui qui offre le meilleur compromis entre l'adéquation aux données et la complexité du modèle. Le critère d'information d'Akaike [4] et le critère d'information bayésien [89] sont souvent utilisés pour mesurer l'adéquation du modèle aux données, tout en pénalisant les modèles avec beaucoup de paramètres.

Le cadre bayésien offre une approche alternative, avec les modèles bayésiens non paramétriques dont l'espace des paramètres est de dimension infinie. L'ordre du modèle décrivant les observations est alors inféré directement à partir de celles-ci.

L'élément central des modèles bayésiens non paramétriques est le processus de Dirichlet (*Dirichlet Process* ou *DP*). Les processus de Dirichlet ont été introduits par Ferguson en 1973 [35] et ont d'abord été appliqués aux modèles de mélange avec un nombre inconnu de composantes par Antoniak en 1974 [7]. Une première extension aux modèles de Markov cachés est proposée par Beal et al. en 2002 [13] puis formalisée dans le cadre des processus de Dirichlet hiérarchiques (*Hierarchical Dirichlet Process* ou *HDP*) par Teh et al. en 2006 [102]. Une introduction générale aux modèles bayésiens non paramétriques est offerte dans [40].

Un processus de Dirichlet est un processus stochastique $\mathbf{G} \sim \mathbf{DP}(\alpha, H)$. Il est paramétré par un paramètre de concentration α et une distribution de base H . Il est défini comme un processus indexé par les partitions (A_1, \dots, A_n) ($n > 0$) de l'espace E sur lequel H est définie, avec pour réalisation des variables de Dirichlet n -dimensionnelles :

$$(G(A_1), \dots, G(A_n)) \sim \mathbf{Dir}(\alpha H(A_1), \dots, \alpha H(A_n)). \quad (2.11)$$

Nous allons voir que les réalisations d'un processus de Dirichlet sont des lois de probabilité discrètes.

Il existe plusieurs représentations des processus de Dirichlet, qui peuvent être utiles pour leur compréhension et leur simulation. Nous rappelons, en particulier, la représentation *stick-breaking* et la représentation par urne de Pólya.

2.3.2.1 Représentation *stick-breaking*

La représentation *stick-breaking* [91] permet de construire simplement des réalisations d'un processus de Dirichlet.

Considérons un processus de Dirichlet $G \sim \mathbf{DP}(\alpha, H)$ de concentration α et de distribution de base H . L'idée est de construire une distribution de probabilité discrète en assignant des probabilités π_k à des échantillons θ_k tirés indépendamment selon H . Un bâton de longueur unitaire représente le vecteur de probabilité π (figure 2.4).

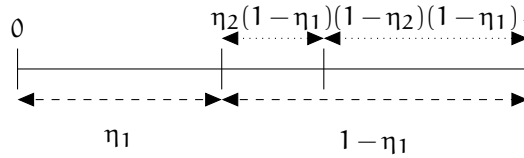


FIGURE 2.4 – Le processus *stick-breaking*.

Le bâton est d'abord cassé en deux morceaux de proportions η_1 et $1 - \eta_1$. Ensuite le second morceau est cassé en deux parties de proportions η_2 et $1 - \eta_2$. On obtient alors trois morceaux de longueurs η_1 , $\eta_2(1 - \eta_1)$ et $(1 - \eta_2)(1 - \eta_1)$. Le processus continue ainsi indéfiniment.

La longueur des morceaux obtenus correspond aux poids π_k . Ainsi, $\pi_1 = \eta_1, \pi_2 = \eta_2(1 - \eta_1), \pi_3 = \eta_3(1 - \eta_2)(1 - \eta_1)$, et en général

$$\pi_k = \eta_k \prod_{l=1}^{k-1} (1 - \eta_l). \tag{2.12}$$

Les proportions η_k sont échantillonnées indépendamment à partir d'une loi $\text{Beta}(1, \alpha)$ où la densité de $\text{Beta}(a, b)$ est donnée par

$$p(x) = \frac{\Gamma(a + b)}{\Gamma(a)\Gamma(b)} x^{a-1} (1 - x)^{b-1} \mathbb{1}_{[0,1]}(x) \tag{2.13}$$

où $\mathbb{1}_A(x) = 1$ si $x \in A$, 0 sinon. La distribution des poids $\pi = (\pi_1, \pi_2, \dots)$ est appelée distribution de Griffiths-Engen-McCloskey, et est notée $\mathbf{GEM}(\alpha)$.

On obtient alors la représentation *stick-breaking* du processus de Dirichlet G :

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k} \text{ avec } \pi \sim \mathbf{GEM}(\alpha) \text{ et } \theta_k \sim H. \tag{2.14}$$

L'équation 2.14 montre que les réalisations d'un processus de Dirichlet sont des mélanges infinis de lois paramétriques.

On remarque que les π_k s tendent vers zéro de façon géométrique, en effet :

$$\mathbb{E}[\pi_k] = \mathbb{E}[\eta_k] \prod_{l=1}^{k-1} (1 - \mathbb{E}[\eta_l]) = \frac{1}{\alpha + 1} \left(\frac{\alpha}{\alpha + 1} \right)^{k+1}. \quad (2.15)$$

Ainsi, plus le paramètre de concentration α est faible, plus les réalisations de G seront concentrées autour d'un faible nombre de valeurs. Lorsque $\alpha \rightarrow 0$, les réalisations tendent à être concentrées sur une seule valeur ($\mathbb{E}[\pi_1] \rightarrow 1$).

2.3.2.2 Représentation par urne de Pólya

La représentation par urne de Pólya [18] est utile pour aborder l'inférence dans les modèles à base de processus de Dirichlet.

Considérons une urne contenant des boules noires et des boules de couleur. À l'initialisation l'urne contient uniquement α boules noires. Lorsque l'on tire une boule de l'urne, si la boule est noire alors une nouvelle boule colorée est tirée selon une distribution de base H et la boule noire ainsi que la boule colorée sont remises dans l'urne ($\bullet \rightarrow \bullet\bullet$). Au contraire, si la boule tirée de l'urne est de couleur, alors elle est remise dans l'urne avec une nouvelle boule de la même couleur ($\bullet \rightarrow \bullet\bullet$). On peut établir que les valeurs (couleurs) de la séquence infinie de tirages, et leurs proportions, définissent une réalisation d'un processus de Dirichlet $DP(\alpha, H)$.

Plus précisément, considérons un modèle de mélange, où z_t dénote l'état caché et y_t l'observation. L'urne de Pólya peut-être décrite comme suit. On note $\theta'_t = \theta_{z_t}$ la valeur de θ associée avec z_t . Si $z_t = k$ alors $\theta'_t = \theta_k$ et y_t est distribué selon p_{θ_k} . Étant donnée une séquence de variables aléatoires $(\theta'_t)_{t>0}$ avec

$$\mathbb{P}(\theta'_1 \in B) = H(B), \quad \text{et} \\ \mathbb{P}(\theta'_{t+1} \in B \mid \theta'_{1:t}) = \frac{1}{\alpha + t} \left(\sum_{\tau=1}^t \delta_{\theta'_\tau}(B) + \alpha H(B) \right) \quad (2.16)$$

il a été montré que la distribution des θ'_t converge presque sûrement vers un $DP(\alpha, H)$ quand $t \rightarrow \infty$ [18].

Remarquons que ce modèle montre une préférence pour les boules de couleur déjà existantes. Plus une couleur est présente dans l'urne, plus il y a de chances qu'une boule de cette couleur soit tirée et dupliquée.

2.3.3 Modèles de mélange à base de processus de Dirichlet

Supposons maintenant que l'on souhaite estimer un modèle de mélange pour des observations $y_{1:T} = (y_1, \dots, y_T)$. On note $p_\theta(y)$ la

densité de la composante paramétrée par θ et on place un a priori $\theta \sim H$. On peut construire un modèle bayésien non paramétrique des observations sous la forme d'un modèle de mélange à base de processus de Dirichlet (*Dirichlet Process Mixture Model* ou *DPMM*). Dans ce modèle la distribution des observations est un mélange avec un nombre infini de composantes :

$$p(\mathbf{y}) = \sum_{k=1}^{\infty} \pi_k p_{\theta_k}(\mathbf{y}) \quad (2.17)$$

et les poids π_k et les paramètres θ_k des différentes composantes suivent un processus de Dirichlet :

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k} \sim \text{DP}(\alpha, H). \quad (2.18)$$

L'estimation des paramètres et des états d'un modèle de mélange non paramétrique à partir de la distribution a posteriori $p(z_{1:T}, \theta_{1:K_T} | y_{1:T})$ peut à nouveau être réalisée avec l'échantillonnage de Gibbs [72].

Commençons par déterminer la loi a priori d'un état caché, conditionnellement aux autres états (équation 2.9 dans le cas fini).

En reprenant le modèle de l'urne de Pólya, nous indiquons par $1, \dots, K_t$ les couleurs distinctes des boules présentes dans l'urne au temps t et nous notons z_t l'indice de la couleur de la nouvelle boule.

Comme le rôle des boules peut être échangé, en définissant $\mathbf{z}_{-t} = \{z_{\tau} : \tau \neq t\}$ et $n_{-t,k} = \#\{z_{\tau} : z_{\tau} = k, \tau \neq t\}$ le nombre d'occurrences de la valeur k dans \mathbf{z}_{-t} , il peut être montré [72] que :

$$\mathbb{P}(z_t = k | \mathbf{z}_{-t}) = \begin{cases} \frac{n_{-t,k}}{\alpha + T - 1} & \text{si } k \text{ est une composante existante} \\ & (n_{-t,k} > 0) \\ \frac{\alpha}{\alpha + T - 1} & \text{si } k \text{ est une nouvelle composante} \\ & (k = K_t + 1) \end{cases} \quad (2.19)$$

Afin d'échantillonner les états cachés, nous devons définir la loi a posteriori des états cachés, conditionnellement aux autres états cachés, et aux observations (équation 2.10 dans le cas fini) :

$$p(z_t = k | \mathbf{y}, \mathbf{z}_{-t}, \boldsymbol{\theta}) \propto \begin{cases} p(z_t = k | \mathbf{z}_{-t}) p(y_t | z_t = k, \boldsymbol{\theta}_k) & \text{si } n_{-t,k} > 0 \\ p(z_t = k | \mathbf{z}_{-t}) \mathcal{J}(y_t) & \text{sinon} \end{cases} \quad (2.20)$$

où la constante de normalisation est donnée par $\sum_{k=1}^K p(z_t = k | \cdot) = 1$. $\mathcal{J}(y_t)$ est la loi marginale a priori des observations, c'est à dire, la

loi conditionnelle d’une observation, intégrée pour toutes les valeurs possibles de la loi a priori θ :

$$\mathcal{J}(y_t) = p(y_t | z_t = K_t + 1) = \int_{\theta} p(y_t | \theta)H(\theta)d\theta. \quad (2.21)$$

En effet, dans le cas où z_t correspond à une nouvelle composante, on ne peut pas utiliser $p(y_t | z_t = K + 1, \theta_{K+1})$, puisque θ_{K+1} n’existe pas encore. Une forme close peut-être obtenue facilement pour $\mathcal{J}(y_t)$ si p_{θ} et H sont des distributions conjuguées. Sinon, une simulation de Metropolis-Hastings peut être utilisée avec la distribution a priori de z_t (équation 2.19) comme distribution auxiliaire [72].

Finalement, nous pouvons échantillonner les états cachés z et les paramètres θ des lois d’observations à l’aide de l’algorithme 2.2, en remplaçant l’équation 2.10 par l’équation 2.20.

2.4 MODÈLES DE MARKOV CACHÉS NON PARAMÉTRIQUES

2.4.1 Modèles de Markov cachés

Puisque les modèles de mélange ne prennent pas en compte la dépendance temporelle entre les états cachés, leur capacité de prédiction est limitée. En effet, connaître l’état z_t qui a généré l’observation y_t n’apporte pas d’information sur le prochain état z_{t+1} .

Les modèles de Markov cachés (figure 2.5) sont une généralisation des modèles de mélange. Dans un modèle de Markov caché l’état z_{t+1} dépend de l’état z_t . Cette dépendance s’exprime au travers de la propriété de Markov : $\mathbb{P}(z_{t+1} | z_{1:t}) = \mathbb{P}(z_{t+1} | z_t)$.

Les probabilités de transition entre deux états successifs sont définies par la matrice π de dimension $K \times K$ avec pour entrées $\pi_{ij} = \mathbb{P}(z_{t+1} = j | z_t = i)$.

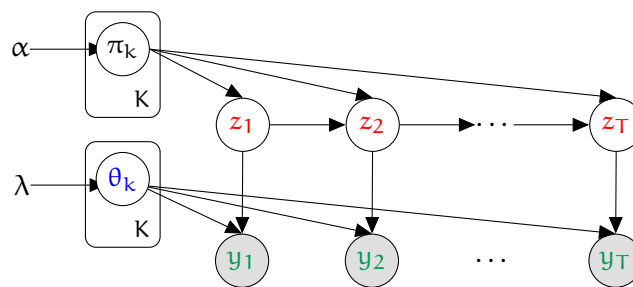


FIGURE 2.5 – Représentation graphique d’un modèle de Markov caché bayésien.

L’estimation au sens du maximum de vraisemblance des paramètres d’un HMM peut être obtenue avec une variante de l’algorithme EM connue sous le nom d’algorithme de Baum-Welch [77]. Cet algorithme est facile à implémenter et est largement étudié, mais il est sensible au surapprentissage sur les données bruitées ou avec peu d’échantillons.

Il demande également une attention particulière à l'initialisation des paramètres, au risque de converger vers un maximum local et non global de la vraisemblance. Enfin, il requiert la connaissance du nombre d'états cachés K , ce qui n'est en général pas le cas avec les séries temporelles de délais.

2.4.1.1 Inférence

Dans cette section nous décrivons l'estimation des paramètres d'un modèle de Markov caché « paramétrique » à l'aide de l'échantillonnage de Gibbs. Cela nous permettra d'introduire le HDP-HMM, ou modèle de Markov caché infini, et l'estimation de ses paramètres dans la section suivante. Tout comme pour le cas d'un modèle de mélange (section 2.3.1.3), nous souhaitons inférer les paramètres du modèle à l'aide de l'échantillonnage de Gibbs. Pour cela nous devons définir les lois a priori sur les paramètres, puis calculer les lois a posteriori qui prennent en compte les observations.

Commençons par l'inférence des lignes de la matrice de transition π . Afin de déterminer la loi a posteriori $p(\pi | z)$ des lignes de la matrice de transition, nous devons déterminer la vraisemblance $p(z | \pi)$ et la loi a priori $p(\pi)$. Pour déterminer $p(z | \pi)$, remarquons que

$$p(z_t = j | z_{t-1} = i, \pi) = \pi_{ij} \quad (2.22)$$

et

$$p(z | \pi) = \prod_{t=2}^T \pi_{z_{t-1}, z_t} = \prod_{i=1}^K \prod_{j=1}^K \pi_{ij}^{n_{ij}} \quad (2.23)$$

où $n_{ij} = \#\{t : z_{t-1} = i, z_t = j\}$ est le nombre de transitions de l'état i vers j dans la séquence z .

De façon analogue au modèle de mélange (équation 2.7), on place un a-priori Dirichlet sur les lignes π_i de la matrice de transition π , les lignes de la matrice de transition étant indépendantes entre-elles :

$$p(\pi) = \prod_{i=1}^K \text{Dir}(\pi_i | \alpha) \propto \prod_{i=1}^K \prod_{j=1}^K \pi_{ij}^{\alpha_j - 1}. \quad (2.24)$$

Finalement, la loi a posteriori sur les lignes de la matrice de transition est obtenue en multipliant l'équation 2.23 par l'équation 2.24 [83] :

$$\begin{aligned} p(\pi | z) &\propto p(\pi)p(z | \pi) \\ &\propto \prod_{i=1}^K \prod_{j=1}^K \pi_{ij}^{n_{ij}} \prod_{i=1}^K \prod_{j=1}^K \pi_{ij}^{\alpha_j - 1} = \prod_{i=1}^K \prod_{j=1}^K \pi_{ij}^{n_{ij} + \alpha_j - 1} \\ &= \prod_{i=1}^K \text{Dir}(n_{i1} + \alpha_1 - 1, \dots, n_{iK} + \alpha_K - 1). \end{aligned} \quad (2.25)$$

On souhaite désormais inférer la séquence des états cachés \mathbf{z} . Pour cela deux approches sont possibles [90]. L'approche classique, appelée *direct assignment sampling*, consiste à échantillonner chaque état caché selon la loi conditionnelle a posteriori $p(z_t | \mathbf{z}_{-t}, \mathbf{y}, \boldsymbol{\pi}, \boldsymbol{\theta})$. Nous avons utilisé une approche similaire dans le cas des modèles de mélange (équation 2.10). Une approche alternative, consiste à échantillonner la séquence (z_1, \dots, z_T) en un seul « bloc » selon la loi conditionnelle a posteriori $p(\mathbf{z} | \mathbf{y}, \boldsymbol{\pi}, \boldsymbol{\theta})$. On parle alors d'échantillonneur de Gibbs par blocs (*blocked Gibbs sampling*). Cette méthode permet à l'échantillonneur de Gibbs de converger plus rapidement vers la loi a posteriori des paramètres [90].

Nous souhaitons donc déterminer la loi a posteriori de la séquence des états cachés $p(\mathbf{z} | \mathbf{y}, \boldsymbol{\pi}, \boldsymbol{\theta})$. Remarquons que la loi se factorise ainsi grâce à la propriété de Markov :

$$p(\mathbf{z} | \mathbf{y}, \boldsymbol{\pi}, \boldsymbol{\theta}) = p(z_T | z_{T-1}, \mathbf{y}, \boldsymbol{\pi}, \boldsymbol{\theta}) p(z_{T-1} | z_{T-2}, \mathbf{y}, \boldsymbol{\pi}, \boldsymbol{\theta}) \cdots p(z_2 | z_1, \mathbf{y}, \boldsymbol{\pi}, \boldsymbol{\theta}) p(z_1 | \mathbf{y}, \boldsymbol{\pi}, \boldsymbol{\theta}). \quad (2.26)$$

Nous pouvons ainsi échantillonner z_1 selon $p(z_1 | \mathbf{y}, \boldsymbol{\pi}, \boldsymbol{\theta})$, puis z_2 selon $p(z_2 | z_1, \mathbf{y}, \boldsymbol{\pi}, \boldsymbol{\theta})$ et ainsi de suite. Ces lois sont données par les équations suivantes [39] :

$$p(z_1 = i | \mathbf{y}, \boldsymbol{\pi}, \boldsymbol{\theta}) \propto p(z_1 = i) p(y_1 | \boldsymbol{\theta}_i) p(y_{2:T} | z_1 = i, \boldsymbol{\pi}, \boldsymbol{\theta}) \\ p(z_t = j | z_{t-1} = i, \mathbf{y}, \boldsymbol{\pi}, \boldsymbol{\theta}) \propto p(z_t = j | z_{t-1} = i) p(y_t | \boldsymbol{\theta}_j) p(y_{t+1:T} | z_t = j, \boldsymbol{\pi}, \boldsymbol{\theta}). \quad (2.27)$$

Le coefficient de proportionalité se calcule facilement à l'aide de la condition de normalisation $\sum_j p(z_t = j | z_{t-1} = i) = 1$. La quantité $p(z_t = j | z_{t-1} = i)$ correspond aux probabilités de transition, et $p(y_t | \boldsymbol{\theta}_j)$ correspond à la loi des observations conditionnellement à l'état caché. La quantité $p(y_{t+1:T} | z_t = i, \boldsymbol{\pi}, \boldsymbol{\theta})$, aussi notée $\beta_t(i)$ et appelée « filtre arrière », se calcule récursivement :

$$\beta_t(i) = \sum_j \beta_{t+1}(j) p(z_{t+1} = j | z_t = i) p(y_{t+1} | z_{t+1} = j) \quad (2.28)$$

lorsque $t < T$, sinon $\beta_t(i) = 1$ lorsque $t = T$.

L'inférence des paramètres $(\boldsymbol{\theta}_k)_1^K$ des lois d'observations est identique au cas d'un modèle de mélange. On utilisera par exemple les équations 2.2 à 2.4 dans le cas de lois d'observations gaussiennes.

L'algorithme 2.3 résume la procédure d'échantillonnage. L'algorithme a une complexité temporelle de $\mathcal{O}(TK^2)$, puisque le calcul du filtre arrière requiert $\mathcal{O}(K^2)$ opérations pour chaque instant t .

2.4.2 Processus de Dirichlet hiérarchiques

Tout comme il est possible de définir des modèles de mélange avec un nombre inconnu d'états, il est possible de définir des modèles de

Algorithme 2.3 : Échantillonneur de Gibbs par blocs pour les modèles de Markov cachés finis

Entrées : $\theta^{(n-1)}$ et $\pi^{(n-1)}$
pour $t \leftarrow T$ à 1 **et** $k \leftarrow 1$ à K
 | calculer $p(\mathbf{y}_{t+1:T} | z_t = k, \pi, \theta)$ (équation 2.28) ;
fin
pour $t \leftarrow 1$ à T
 | échantillonner $z_t \sim p(z_t | z_{t-1}, \mathbf{y}, \pi^{(n-1)}, \theta^{(n-1)})$
 | (équation 2.27) ;
fin
pour $k \leftarrow 1$ à K
 | échantillonner $\pi_k^{(n)} \sim p(\pi_k | \mathbf{z})$ (équation 2.25) ;
 | échantillonner $\theta_k^{(n)} \sim p(\theta_k | \mathbf{y}, \mathbf{z})$ (équation 2.4) ;
fin

Markov cachés avec un nombre inconnu d'états, à l'aide des processus de Dirichlet. Un tel modèle est appelé « modèle de Markov caché à base de processus de Dirichlet hiérarchique » (HDP-HMM) [102].

Dans le HDP-HMM, un processus de Dirichlet \mathbf{G}_i est utilisé comme a priori sur chaque ligne $\pi_i = (\pi_{i1}, \pi_{i2}, \dots)$ de la matrice de transition. Il faut cependant s'assurer que les probabilités de transition π_{ik} , pour toutes les lignes i , correspondent aux mêmes lois d'émission p_{θ_k} . Pour cela, chaque DP \mathbf{G}_i est paramétré par la même distribution de base à valeur discrète \mathbf{G}_0 :

$$\mathbf{G}_i = \sum_k \pi_{ik} \delta_{\theta_k} \sim \mathbf{DP}(\alpha, \mathbf{G}_0) \quad (2.29)$$

où \mathbf{G}_0 suit un processus de Dirichlet de distribution de base H :

$$\mathbf{G}_0 = \sum_k \beta_k \delta_{\theta_k} \sim \mathbf{DP}(\gamma, H). \quad (2.30)$$

On parle alors de processus de Dirichlet *hiérarchique*.

2.4.3 Modèles de Markov cachés à base de processus de Dirichlet hiérarchique

Cette hiérarchie de processus de Dirichlet produit le HDP-HMM, représenté dans la figure 2.6. γ est le paramètre de concentration du processus de Dirichlet racine \mathbf{G}_0 (aussi noté β). Une petite valeur pour γ correspond à un petit nombre d'états a priori (équation 2.15). α est le paramètre de concentration des processus de Dirichlet associés à chaque ligne de la matrice de transition. Une petite valeur pour α indique que chaque ligne de la matrice contient a priori peu de transitions. κ est le paramètre dit *sticky* qui contrôle l'a priori sur les auto

transitions $\pi_{i,i}$. Une grande valeur pour κ indique une grande probabilité a priori de rester dans un état donné. Enfin, λ correspond aux paramètres de la loi a priori sur les paramètres des lois d'observation.

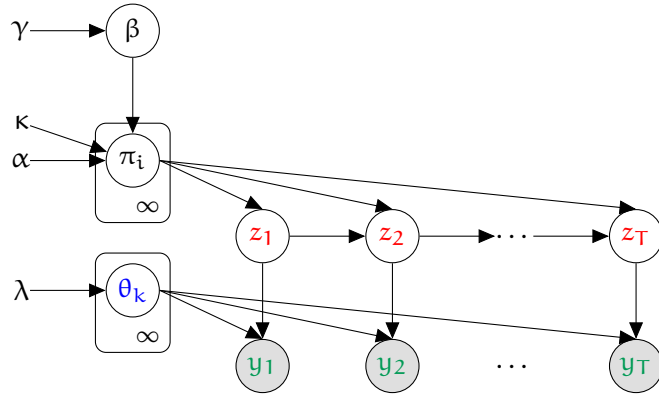


FIGURE 2.6 – Représentation graphique d’un modèle de Markov caché à base de processus de Dirichlet hiérarchique.

2.4.3.1 Échantillonneurs

Plusieurs échantillonneurs ont été dérivés pour l’inférence dans les HDP-HMMs (tableau 2.1). L’échantillonneur direct [39, 102] est analogue à l’échantillonneur présenté précédemment pour les modèles de mélange : les états cachés sont échantillonnés un à un, conditionnellement au reste des paramètres. L’échantillonneur par blocs [39] échantillonne les états $z_{1:T}$ conjointement, de façon similaire à l’échantillonneur présenté pour les HMMs standards. Cet échantillonneur converge plus rapidement vers la distribution a posteriori des paramètres [39]. L’échantillonneur de Gibbs particulière [104] se place dans le cadre des méthodes de Monte Carlo séquentielles. Enfin le *beam sampler* [105] est similaire à l’échantillonneur par blocs mais il échantillonne les états en plusieurs blocs de taille fixe.

Algorithme	Références	Complexité
Gibbs direct	Teh et al., 2006 [102] Fox et al., 2011 [39]	$\mathcal{O}(TK)$
Gibbs par blocs	Fox et al., 2011 [39]	$\mathcal{O}(TL^2)$
Gibbs particulière	Tripuraneni et al., 2015 [104]	$\mathcal{O}(TKN)$
<i>Beam sampler</i>	Van Gael et al., 2008 [105]	$\mathcal{O}(TK^2)$

TABLE 2.1 – Méthodes de Monte-Carlo pour le HDP-HMM.

T : nombre d’observations, K : nombre d’états instanciés,
L : borne supérieure sur le nombre d’états, N : nombre de particules.

Comme pour le cas d’un modèle de Markov caché fini (section 2.4.1.1), nous avons choisi d’utiliser l’échantillonneur par blocs dans nos tra-

vaux. Celui-ci converge plus rapidement vers la distribution a posteriori que l'échantillonneur direct [39], et son implémentation est plus aisée.

2.4.3.2 Échantillonneur de Gibbs par blocs

L'échantillonneur de Gibbs par blocs du HDP-HMM est similaire à celui présenté pour le HMM fini [section 2.4.1.1](#). La principale différence concerne l'inférence de la matrice de transition π . Le détail des calculs menant aux équations finales étant compliqué, nous résumons ici les principaux résultats, et le lecteur intéressé pourra trouver plus de détails dans [39, 102].

Dans le HDP-HMM, l'a priori sur la distribution de base β est un processus stick-breaking de concentration γ :

$$p(\beta) = \mathbf{GEM}(\gamma). \quad (2.31)$$

L'a priori sur les lignes π_i de la matrice de transition est un processus de Dirichlet de concentration α et de distribution de base β :

$$p(\pi_i | \beta) = \mathbf{DP} \left(\alpha + \kappa, \frac{\alpha\beta + \kappa\delta_j}{\alpha + \kappa} \right). \quad (2.32)$$

Ici $\alpha\beta + \kappa\delta_j$ indique que l'on ajoute la quantité κ à la j -ième composante du vecteur $\alpha\beta$. Ainsi une grande valeur pour κ indique une forte probabilité d'auto-transition π_{ii} a priori pour chaque état i .

Dans l'échantillonneur par blocs, le processus stick-breaking \mathbf{GEM} est approximé par une distribution de Dirichlet d'ordre L :

$$\mathbf{GEM}_L(\gamma) = \mathbf{Dir}(\gamma/L, \dots, \gamma/L) \quad (2.33)$$

où L est une borne supérieure sur le nombre d'états qui pourra être inféré. Cette approximation, dite *weak limit*, est introduite par Ishwaran et al. dans [51]. Elle permet de simplifier l'implémentation de l'échantillonneur, puisque la dimension L des vecteurs de probabilité est fixe au cours des itérations, et ne varie pas avec le nombre d'états actuellement instanciés.

Sous cette approximation, la loi a priori sur la distribution de base β est une distribution de Dirichlet d'ordre L :

$$p(\beta) = \mathbf{Dir}(\gamma/L, \dots, \gamma/L). \quad (2.34)$$

et la loi a posteriori, prenant en compte les observations, est donnée par :

$$p(\beta | \bar{\mathbf{m}}) = \mathbf{Dir}(\gamma/L + \bar{\mathbf{m}}_{.1}, \dots, \gamma/L + \bar{\mathbf{m}}_{.L}) \quad (2.35)$$

où $\bar{\mathbf{m}}_{.j} = \sum_i \bar{\mathbf{m}}_{ij}$. Intuitivement, $\bar{\mathbf{m}}_{ij}$ est une quantité proportionnelle au nombre de transitions de l'état i à j dans la séquence d'états \mathbf{z} . Le calcul des $\bar{\mathbf{m}}_{.j}$ à partir des observations est détaillé dans [39]. Étant

donné β , et sous l'approximation *weak limit*, la loi a priori sur les lignes de la matrice de transition est donnée par

$$p(\pi_i | \beta) = \text{Dir}(\alpha\beta_1, \dots, \alpha\beta_i + \kappa, \dots, \alpha\beta_L) \quad (2.36)$$

et la loi a posteriori est donnée par

$$p(\pi_i | z, \beta) = \text{Dir}(\alpha\beta_1 + n_{i1}, \dots, \alpha\beta_i + n_{ii} + \kappa, \dots, \alpha\beta_L + n_{iL}) \quad (2.37)$$

où $n_{ij} = \#\{t : z_{t-1} = i, z_t = j\}$ est le nombre de transitions de l'état i vers j dans la séquence z .

Une fois β et la matrice de transition π inférés, l'inférence de la séquence des états cachés z et des paramètres $(\theta_k)_1^K$ des lois d'observations est identique au cas d'un HMM fini. L'échantillonneur est décrit dans l'[algorithme 2.4](#).

Algorithme 2.4 : Échantillonneur de Gibbs par blocs pour les modèles de Markov cachés infinis

Entrées : $\theta^{(n-1)}$, $\beta^{(n-1)}$ et $\pi^{(n-1)}$
pour $t \leftarrow T$ à 1 **et** $k \leftarrow 1$ à K
 | calculer $p(y_{t+1:T} | z_t = k, \pi, \theta)$ ([équation 2.28](#)) ;
fin
pour $t \leftarrow 1$ à T
 | échantillonner $z_t \sim p(z_t | z_{t-1}, \mathbf{y}, \pi^{(n-1)}, \theta^{(n-1)})$
 | ([équation 2.27](#)) ;
fin
pour $k \leftarrow 1$ à K
 | échantillonner $\bar{\mathbf{m}}$ à partir de $\beta^{(n-1)}$ et z (voir [39]) ;
 | échantillonner $\beta^{(n)} \sim p(\beta | \bar{\mathbf{m}})$ ([équation 2.35](#)) ;
 | échantillonner $\pi_k^{(n)} \sim p(\pi_k | z, \beta)$ ([équation 2.37](#)) ;
 | échantillonner $\theta_k^{(n)} \sim p(\theta_k | \mathbf{y}, z)$ ([équation 2.4](#)) ;
fin

2.4.3.3 Intégration du DPMM dans le HDP-HMM

Il est possible de combiner les modèles de mélange infinis, avec les modèles de Markov cachés infinis, en utilisant un DPMM comme loi d'observation $p(y_t | z_t = j)$ du HDP-HMM. Cela permet d'obtenir un modèle flexible, avec peu d'a priori sur la distribution des observations dans un état donné. Ce modèle est décrit [39] et consiste à redéfinir l'état caché comme le couple (z_t, s_t) , où z_t correspond à l'état caché du HMM, et s_t à l'état caché du DPMM.

On associe à chaque état j du HDP-HMM, un modèle de mélange dont les poids ψ_j suivent un processus stick-breaking de concentration σ :

$$p(\psi_j) = \text{GEM}(\sigma) \quad (2.38)$$

et

$$p(s_t = k \mid z_t = j, \psi_j) = (\psi_j)_k. \quad (2.39)$$

Sous l'approximation *weak limit* de l'échantillonneur par blocs, ψ_j suit une distribution de Dirichlet d'ordre L' , où L' définit le nombre maximal d'états cachés pour le modèle de mélange :

$$p(\psi_j) = \mathbf{Dir}(\sigma/L', \dots, \sigma/L'). \quad (2.40)$$

Dans ce modèle, la loi des observations dépend de l'état du HDP-HMM et de l'état du DPMM. Par exemple, dans le cas d'un DPMM avec des composantes gaussiennes :

$$p(y_t \mid z_t = j, s_t = k, \theta_{j,k}) = \mathcal{N}(y_t \mid \mu_{j,k}, \sigma_{j,k}^2) \quad (2.41)$$

où $\theta_{j,k} = (\mu_{j,k}, \sigma_{j,k}^2)$.

La seule différence avec le HDP-HMM classique pour l'inférence des paramètres avec l'échantillonneur par blocs, se trouve dans l'inférence de la séquence des états cachés, puisque nous avons redéfini l'état caché pour inclure la composante du modèle de mélange.

Nous modifions la loi a posteriori des états cachés (équation 2.27) pour prendre en compte la nouvelle définition :

$$\begin{aligned} p(z_t = j, s_t = k \mid z_{t-1} = i, \mathbf{y}, \boldsymbol{\pi}, \boldsymbol{\psi}, \boldsymbol{\theta}) \\ \propto \underbrace{p(z_t = j \mid z_{t-1} = i)}_{\text{probabilités de transition}} \underbrace{p(s_t = k \mid \psi_j)}_{\text{poids du mélange}} \underbrace{p(y_t \mid \theta_{j,k})}_{\text{vraisemblance}} \underbrace{p(y_{t+1:T} \mid z_t = j, \boldsymbol{\pi}, \boldsymbol{\psi}, \boldsymbol{\theta})}_{\text{filtre arrière}}. \end{aligned} \quad (2.42)$$

Le coefficient de proportionalité se calcule avec la condition de normalisation $\sum_j \sum_k p(z_t = j, s_t = k \mid z_{t-1} = i, \cdot) = 1$. Le filtre arrière $\beta'_t(i) = p(y_{t+1:T} \mid z_t = j, \boldsymbol{\pi}, \boldsymbol{\psi}, \boldsymbol{\theta})$ se calcule de façon similaire au cas fini (équation 2.28), mais avec une double somme sur les états cachés du HMM et du DPMM :

$$\beta'_t(i) = \sum_{j=1}^L \sum_{k=1}^{L'} \beta'_{t+1}(j) p(z_{t+1} = j \mid \boldsymbol{\pi}_i) p(s_{t+1} = k \mid \psi_j) f_{jk}(y_{t+1}) \quad (2.43)$$

où $f_{jk}(y_{t+1}) = p(y_{t+1} \mid z_{t+1} = j, s_{t+1} = k)$ et lorsque $t < T$, sinon $\beta_t(i) = 1$ lorsque $t = T$.

Pour réaliser l'échantillonnage il suffit de remplacer l'équation 2.27 par l'équation 2.42 dans l'algorithme 2.4.

2.5 MODÉLISATION DU DÉLAI À L'AIDE DU HDP-HMM

Nous venons d'introduire dans les sections précédentes les modèles de mélange et les modèles de Markov cachés, ainsi que leurs variantes « bayésiennes non paramétriques ». Ces modèles sont tous des modèles à états cachés, ce qui leur permet de fournir une segmentation

des données. Comme résumé dans le [tableau 2.2](#), les modèles de mélange ne prennent pas en compte la dépendance temporelle entre les observations, et les modèles non paramétriques ne nécessitent pas de connaissances a priori du nombre d'états. Ainsi, les modèles de Markov cachés non paramétriques se prêtent bien au problème de la segmentation de séries temporelles de RTT, où le nombre d'états cachés n'est pas connu a priori.

Modèle	Nombre d'états	Dépendance temporelle
Modèle de mélange	Fixé	Non
Modèle de Markov caché	Fixé	Oui
Modèle de mélange à base de processus de Dirichlet	Infini	Non
Modèle de Markov caché à base de processus de Dirichlet hiérarchique	Infini	Oui

TABLE 2.2 – Taxonomie des modèles à états cachés.

Dans la [section 2.2](#) nous avons mis en avant, visuellement, que les séries temporelles de délais exhibent une dépendance temporelle entre les observations. Intuitivement, en l'absence de changement du niveau de trafic et du routage, le délai au temps $t + 1$ doit être similaire au délai au temps t . Nous avons également mis en avant que, dans certains cas, la distribution du délai peut suivre différentes lois de probabilités (normale, exponentielle ou une autre loi paramétrique).

Dans cette section nous proposons d'utiliser le HDP-HMM avec un DPMM comme modèle d'observation, permettant ainsi de modéliser une grande variété de séries. Nous discutons ensuite de l'implémentation de ce modèle et de quelques considérations pratiques liées à l'inférence du modèle sur des données réelles.

2.5.1 Un modèle d'observation non paramétrique

Comme on peut le voir dans la [figure 2.7](#), la distribution du délai varie au cours du temps. Dans certains cas le délai est relativement constant (état vert clair par exemple) et semble suivre une loi normale. Dans d'autres cas, le délai subit des variations importantes (état bleu par exemple), potentiellement causées par des variations du niveau de trafic, et semble suivre une loi plus complexe.

Nous avons également vu dans la [section 2.2](#) que différentes distributions paramétriques avaient été proposées pour le délai. Ici, plutôt que de choisir une distribution en particulier, nous proposons d'utiliser un modèle de mélange à base de processus de Dirichlet, avec des com-

On utilisera les termes loi d'observation et modèle d'émission de façon interchangeable.

posantes gaussiennes, comme loi d'observation du HDP-HMM. En effet, un modèle de mélange gaussien permet d'approximer n'importe quelle distribution régulière étant donné un nombre suffisamment grand de composantes [44, p. 67].

Remarquons qu'un tel modèle implique une double incertitude sur le nombre de paramètres : le nombre d'états du modèle de Markov caché et le nombre de composantes du modèle de mélange à l'intérieur de chaque état. Ce modèle aurait été compliqué à formuler en dehors du cadre bayésien non paramétrique, puisqu'une approche de sélection de modèles classique nécessiterait de choisir entre des HMMs avec différents nombres d'états cachés, et différents nombres de composantes pour chacun des ces états cachés. Si on note K le nombre d'états considérés pour le HMM, et L l'ordre maximal considéré pour chaque modèle de mélange, alors une approche naïve nécessiterait d'estimer L^K modèles HMM pour chaque ordre K . Le HDP-HMM offre une alternative moins coûteuse qui ne nécessite que de l'ordre de LK^2 opérations où K est une borne supérieure sur le nombre d'états (section 2.4.2).

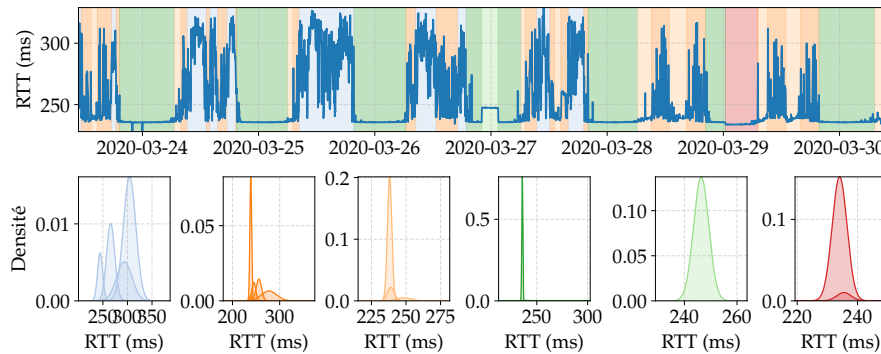
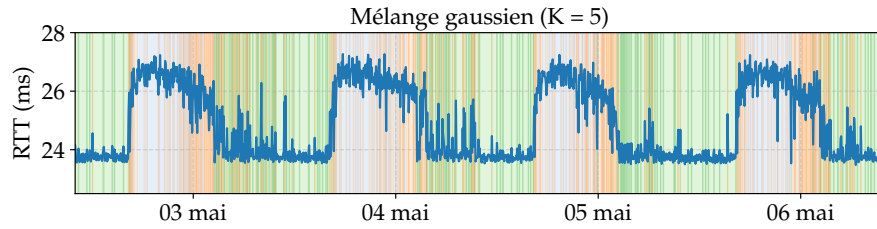
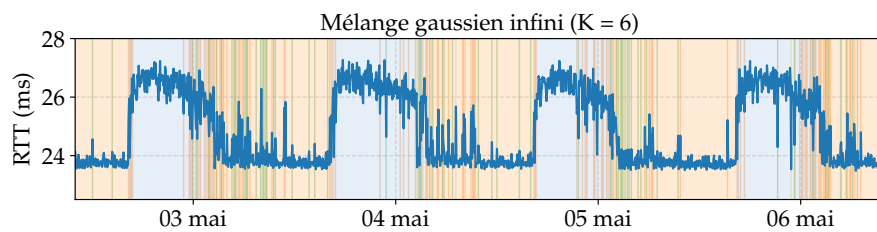


FIGURE 2.7 – Segmentation du RTT entre deux ancres RIPE Atlas sous la modélisation HDP-HMM, et densité des modèles de mélange inférés.

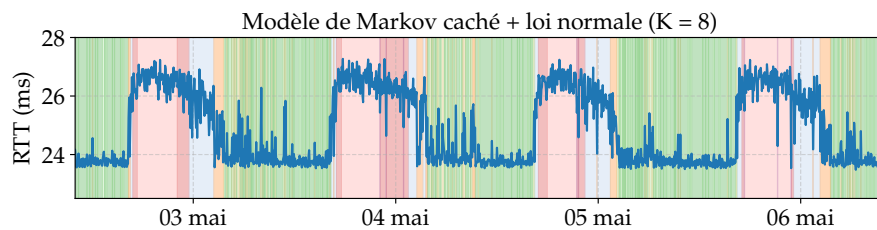
La figure 2.8 représente la segmentation obtenue avec un HDP-HMM avec un mélange infini pour loi d'observation (section 2.4.3.3), un MM paramétrique (algorithme 2.2), un MM non paramétrique (section 2.3.3) et un HMM paramétrique (algorithme 2.3). Dans le cas du MM et du HMM paramétrique, le nombre d'états cachés a été choisi en estimant le modèle pour différents nombres de composantes, et en choisissant celui qui minimise la vraisemblance pénalisée avec un critère BIC. Sur cette série en particulier, le HDP-HMM produit une segmentation proche de ce qu'un expert humain ferait, au contraire des autres modèles qui produisent trop de changements d'état.



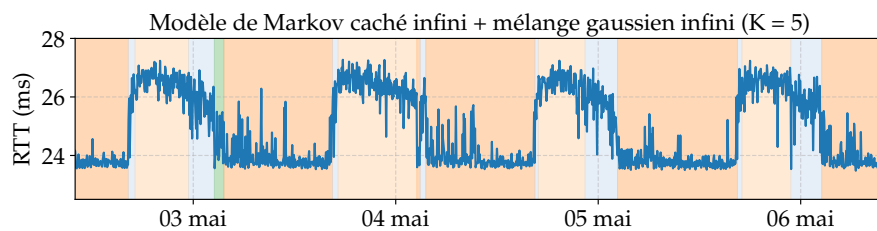
(a) Modèle de mélange gaussien estimé avec le critère d'information bayésien.



(b) Modèle de mélange gaussien à base de processus de Dirichlet.



(c) Modèle de Markov caché estimé avec le critère d'information bayésien.



(d) Modèle de Markov caché à base de processus de Dirichlet.

FIGURE 2.8 – Segmentation d'une série temporelle de RTT avec des modèles de mélange et de HMMs paramétriques et non paramétriques.

2.5.2 Implémentation du HDP-HMM à émissions DPMM

Il existe, à notre connaissance, deux implémentations du HDP-HMM, fonctionnelles et sous licence logicielle libre : `pyhsmm` [52], une librairie Python et `HDPHMM_HDPSLDS_toolbox` [38], une librairie Matlab.

Ces deux librairies supportent le HDP-HMM à émissions DPMM, mais présentent certaines limitations :

- La librairie `HDPHMM_HDPSLDS` ne supporte que la loi normale (multivariée) comme loi d'émission du DPMM.
- La librairie `pyhsmm` ne supporte pas le rééchantillonnage du paramètre `sticky`.
- Ces deux librairies souffrent d'un coût de calcul élevé pour les séries temporelles de moins de 10000 observations.

Nous avons créé notre propre implémentation, `HDPHMM.jl`, qui prend en charge le HDP-HMM avec émissions DPMM, ainsi que le rééchantillonnage de tous les hyperparamètres. Nous implémentons le même échantillonneur par blocs (section 2.4.2) que pour `pyhsmm` et `HDPHMM_HDPSLDS`. Notre implémentation exploite les librairies `Julia Distributions.jl` [16] et `ConjugatePriors.jl` [63], afin d'offrir un support pour un grand nombre (> 50) de lois d'observations.

À la différence de Python et Matlab qui sont des langages interprétés, Julia génère du code machine lors de l'exécution du programme (compilation JIT ou *Just In Time*). Julia offre ainsi, en général, des temps d'exécutions inférieurs à Python et Matlab, et similaires à d'autres langages compilés tels que C++ ou Fortran. De plus, afin d'obtenir une implémentation rapide, permettant de segmenter un grand nombre de traces, nous avons pris soin de minimiser les allocations mémoires et de maximiser l'utilisation du cache processeur.

Notre implémentation est ainsi cinq fois plus rapide (0.8 secondes contre 4 secondes) que les autres implémentations sur des séries de 2500 points, correspondants à une semaine de mesure sur RIPE Atlas (figure 2.9). Les temps de calcul sont mesurés sur un processeur Intel Core i7-7600U @ 2.80GHz.

Le tableau 2.3 compare les caractéristiques des différentes implémentations. La rapidité de notre implémentation nous permet de segmenter et d'analyser un grand nombre de séries temporelles (section 3.2), et de déployer notre méthode dans l'infrastructure de mesure RIPE Atlas (section 3.3).

2.5.3 Considérations pratiques pour l'inférence

La nature bayésienne du HDP-HMM le rend robuste aux observations aberrantes, et sa nature non paramétrique lui permet de modéliser des distributions de délai variées. Lorsqu'on travaille sur des données réelles, il convient de prendre certaines précautions afin d'obtenir des modèles de bonne qualité.

Librairie	Hyp. params.	Lois supportées
HDPHMM_HDPSLDS	$\alpha, \gamma, \kappa, \sigma$	Loi normale multivariée
HDPHMM.jl	$\alpha, \gamma, \kappa, \sigma$	Toutes les lois discrètes et continues implémentées par Distributions.jl
pyhsmm	α, γ, σ	Binomiale, Normale, Géométrique, Multinomiale, Poisson, Uniforme

TABLE 2.3 – Comparaison des implémentations du HDP-HMM. La colonne « Hyp. params. » indique les hyperparamètres dont le rééchantillonnage est supporté.

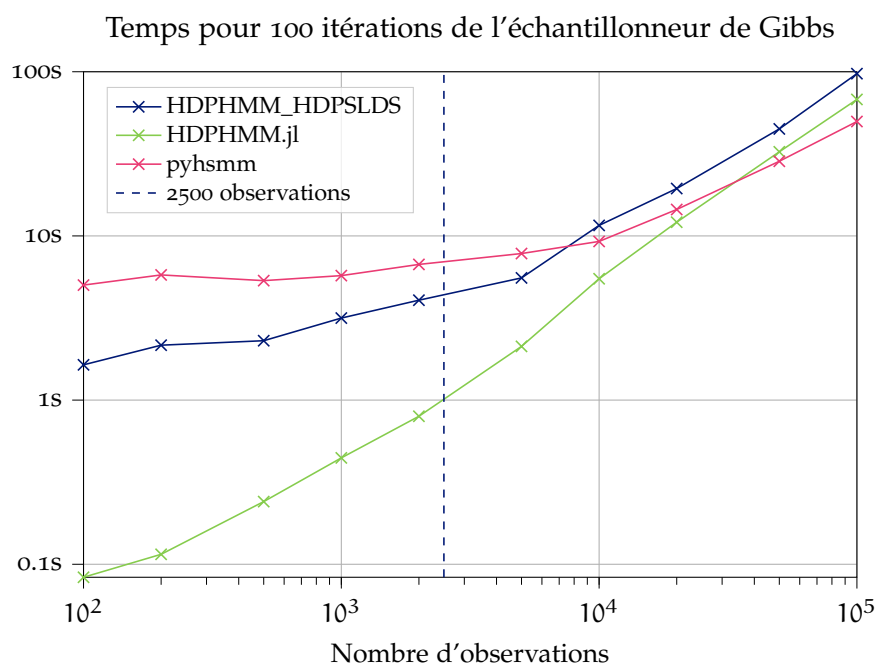


FIGURE 2.9 – Comparaison de la performance des implémentations du HDP-HMM. Temps mesurés sur un processeur Intel Core i7-7600U @ 2.80GHz. 2500 observations correspondent à une semaine de mesure sur RIPE Atlas. Paramètres de l'échantillonneur par blocs : $L = 20, L' = 5$.

2.5.3.1 Qualité des données

Le modèle de Markov caché est un modèle à temps discret. Il est donc nécessaire de s'assurer que toutes les observations, y compris les observations manquantes, sont explicitement présentes dans le jeu de données pour l'inférence. Certaines plateformes, comme RIPE Atlas (dans le cas d'une sonde déconnectée par exemple), n'indiquent pas les mesures manquantes, et renvoient une série temporelle où l'intervalle de temps n'est pas constant. Par exemple, si des mesures sont réalisées toutes les 240 secondes (4 minutes) et que la deuxième mesure est manquante, alors la plateforme peut renvoyer la séquence (0 : 2.21 ms, 480 : 2.25 ms). Il convient donc d'insérer la mesure manquante pour obtenir la séquence (0 : 2.21 ms, 240 : *, 480 : 2.25 ms).

Les données manquantes sont traitées par le HDP-HMM comme des observations qui n'apportent pas d'informations sur l'état caché. Autrement dit, la vraisemblance des observations manquantes conditionnellement à l'état caché est égale pour tous les états cachés : $p(* | z_t = k, \theta_k) = 1$ pour toutes les valeurs de k , où $*$ représente une observation manquante. Ainsi, seule la matrice de transition est prise en compte pour inférer leur état caché.

2.5.3.2 Choix des hyperparamètres

Les hyperparamètres sont les paramètres des lois a priori. Dans le cas du HDP-HMM il s'agit des paramètres de concentration γ et α qui contrôlent le nombre d'états ainsi que le nombre de transitions entre ces états, du paramètre κ qui contrôle les autotransitions, et des paramètres λ de la loi a priori sur les paramètres de la loi d'observation (voir [figure 2.6](#)). Afin de créer un modèle flexible, nous plaçons une loi a priori peu informative $\text{Gamma}(1, 100)$ sur γ et $\alpha + \kappa$ en suivant [39, sec. 6.1]. Nous utilisons un DPMM avec des composantes normales comme modèle d'émission du HDP-HMM. Nous plaçons un a priori $\text{Gamma}(1, 0.5)$ sur le paramètre de concentration du DPMM. Nous plaçons un a priori $\text{NIG}(\mu, 1, 5, 5\sigma^2)$ sur le couple (moyenne, variance) des composantes normales. μ et σ^2 correspondent à la moyenne et à la variance empirique des observations. Afin de limiter l'impact des observations aberrantes, il est possible d'utiliser des estimateurs robustes (écart absolu médian par exemple).

2.5.3.3 Initialisation des paramètres

Étant donné un grand nombre d'itérations, l'échantillonnage de Gibbs produit de façon certaine des échantillons distribués selon la loi a posteriori recherchée. On dit qu'il converge en loi vers la loi a posteriori recherchée. Cependant, si l'échantillonneur est initialisé dans une zone de faible densité de la loi a posteriori, alors la convergence peut être lente. Afin d'accélérer l'échantillonnage, nous réalisons un pre-

mier partitionnement des données à l'aide de l'algorithme k-means++ puis de l'algorithme k-means, comme décrit dans la [section 2.3.1.3](#).

2.5.3.4 Convergence de l'échantillonneur

Au contraire des méthodes du maximum de vraisemblance (algorithme EM par exemple), l'échantillonneur de Gibbs produit des échantillons distribués selon la loi a posteriori, et non une seule valeur pour chaque paramètre. Plusieurs solutions sont possibles pour se ramener à une seule valeur si besoin. Il est par exemple possible de choisir le mode ou la médiane des paramètres.

Afin d'obtenir une seule séquence d'états cachés, et segmenter les séries, nous utilisons l'approche proposée par Fox et al. [39, sec. 8]. Cette approche consiste à séparer les échantillons obtenus en deux ensembles : un ensemble de *référence* et un ensemble de *test*. Pour chaque séquence d'états cachés dans le jeu de test, on calcule sa distance de Hamming moyenne avec les séquences dans le jeu de référence. Pour une séquence de test z et une séquence de référence z' , la distance de Hamming est égale à $\sum_i \mathbb{1}_{z_i \neq z'_i}$. On choisit finalement la séquence du jeu de test qui possède la distance de Hamming moyenne minimale avec les séquences du jeu de référence.

Pour toutes nos segmentations, nous utilisons 250 itérations de l'échantillonneur de Gibbs en ignorant les 25 premiers échantillons. Nous n'avons pas observé de changement significatif des segmentations obtenues en augmentant le nombre d'itérations. En utilisant l'échantillonneur par blocs, avec $L = 20$ et $L' = 5$, cela nous permet de segmenter une série de 2500 points en 2.8 secondes sur un processeur Intel Core i7-7600U @ 2.80GHz, initialisation comprise.

2.6 VALIDATION DE LA MODÉLISATION HDP-HMM DU DÉLAI

Dans cette section nous comparons l'adéquation du HDP-HMM aux mesures de délai, à celle d'autres modèles plus simples comme le HDP-HMM avec une loi d'observation normale ou un DPMM seul (qui ne prend pas en compte la dépendance temporelle). Nous utilisons des données RIPE Atlas [98], dont un jeu de données d'instantés de changements labélisés [96]. Les méthodes et infrastructures de mesure d'Internet sont détaillées plus précisément dans le chapitre suivant ([section 3.1](#)).

2.6.1 Validation sur un jeu de changements labélisés

Bien qu'il n'existe pas de vérité terrain (*ground truth*) pour les états cachés, il existe un jeu de données d'instantés de changements labélisés dans les séries de RTT [96]. Ce jeu de données contient des séries RIPE Atlas où les changements significatifs de délai ont été labélisés par

des experts humains. La performance de segmentation des humains a été validée sur un jeu de données synthétiques où les instants de changements sont connus. Il n'existe pas, à notre connaissance, d'autres jeux de données de changements de RTT publics qui utilisent des données réelles et soient labélisés.

La détection de changement avec le HDP-HMM consiste simplement à chercher les instants de changement dans la séquence inférée des états cachés. Ce jeu de données nous permet donc de valider les instants de changements d'état inférés par le HDP-HMM. Il ne nous permet cependant pas de valider la valeur des états en eux-mêmes. Cela sera étudié dans le chapitre suivant en étudiant la corrélation avec les chemins IP et la congestion (section 3.2.1).

Dans [96] le jeu de données comprend 50 séries de RTT de longueurs différentes, pour un total de 34008 heures d'observations. La détection de changement est réalisée en minimisant

$$\sum_{i=1}^{m+1} [C(y_{\tau_{i-1}:\tau_i})] + \beta f(m) \quad (2.44)$$

où m est le nombre de changements, τ les instants de changements, C une fonction de coût qui mesure la stabilité du délai sur un intervalle consécutif, et $f(m)$ une pénalité qui limite le surapprentissage. Plusieurs fonctions de coûts et pénalités sont considérées. Nous avons comparé le HDP-HMM avec les deux méthodes les plus performantes de l'article : une fonction de coût Poisson et une fonction de coût non paramétrique. La fonction de coût non paramétrique est définie dans [46]. La fonction de coût Poisson est définie comme la log-vraisemblance maximale sous hypothèse poissonnienne :

$$C(y_{s:t}) = -\max_{\lambda} \sum_{i=s}^t \log \frac{\lambda^{[y_i]} e^{-\lambda}}{[y_i]!} \quad (2.45)$$

où $[.]$ représente la partie entière, la loi de Poisson étant à support discret. Pour cette fonction de coût, le délai est également soustrait de son minimum en pré-traitement. Une pénalité BIC modifiée [111] est utilisée :

$$\beta f(m) = -\frac{1}{2} \left[3f(m) \log n + \sum_{i=1}^{m+1} \log(r_i - r_{i-1}) \right], \quad r_i = \tau_i/n \quad (2.46)$$

Enfin, plusieurs métriques de classification sont considérées :

$$\begin{aligned} \text{Précision} &= \frac{\text{Vrais Positifs}}{\text{Vrais Positifs} + \text{Faux Positifs}} \\ \text{Rappel} &= \frac{\text{Vrais Positifs}}{\text{Vrais Positifs} + \text{Faux Négatifs}} \\ \text{F2} &= \frac{5 \times \text{Précision} \times \text{Rappel}}{4 \times \text{Précision} + \text{Rappel}} \end{aligned}$$

Un score F2 pondéré est également défini. Il est calculé sur le rappel pondéré, qui donne plus d'importance aux grands changements de RTT (voir [96] pour plus de détails).

Nous avons reproduit les résultats de l'article original à l'aide du code fourni par l'auteur [93], en incluant le DPMM et le HDP-HMM (avec émissions DPMM). Dans l'article original, les données manquantes (0.17% des observations) sont remplacées par la valeur 1000, car les méthodes de détection de changements proposées ne prennent pas en compte les valeurs manquantes (contrairement au DPMM et au HDP-HMM). Une approche alternative, étant donné le faible nombre de valeurs manquantes, consiste à remplacer les données manquantes par la valeur précédente. Les résultats sont donnés pour les deux approches dans les [tableaux 2.4](#) et [2.5](#).

Le DPMM offre un très bon rappel mais une très mauvaise précision. Autrement dit, le DPMM fournit très peu de faux négatifs (non détection) mais beaucoup de faux positifs (fausse alarme). Cela est cohérent avec la sursegmentation observée dans la [figure 2.2](#), et confirme l'hypothèse qu'un modèle à observations indépendantes ne permet pas de modéliser correctement le délai sur plusieurs heures. Si on ignore le DPMM, le HDP-HMM offre des performances supérieures en terme de rappel par rapport aux méthodes de détection de changements classiques. La précision du HDP-HMM est légèrement plus faible mais le score F2 reste égal ou supérieur aux autres méthodes.

Méthode	Précision	Rappel	Rappel _w	F2	F2 _w
DPMM	0.06	0.96	0.99	0.22	0.23
HDP-HMM	0.38	0.91	0.96	0.70	0.74
NP&MBIC	0.33	0.8	0.89	0.59	0.65
Poisson&MBIC	0.39	0.74	0.85	0.58	0.69

TABLE 2.4 – Comparaison du HDP-HMM avec des méthodes de détection de changement standards sur un jeu de données d'instantés de changements labélisés. Score médian calculé sur 50 séries RIPE Atlas labélisées. Méthodologie originale avec les observations manquantes remplacées par 1000.

Les méthodes de détection de changement standards présentés ci-dessus ont une complexité $\mathcal{O}(T)$, linéaire avec le nombre d'observations T , grâce à l'algorithme PELT (*Pruned Exact Linear Time*) [58] implémenté dans la librairie R `changepoint` [57]. L'échantillonneur que nous utilisons pour le HDP-HMM a une complexité $\mathcal{O}(TL^2)$, linéaire avec le nombre d'observations, et quadratique avec le nombre maximal d'états cachés L (voir [section 2.4.2](#)). Les deux approches ont donc une complexité similaire, bien qu'en pratique les nombreuses itérations de l'échantillonneur de Gibbs, nécessaires pour obtenir une bonne segmentation, résultent en un temps d'exécution plus long que

Méthode	Précision	Rappel	Rappel _w	F2	F2 _w
DPMM	0.02	1.0	1.0	0.1	0.1
HDP-HMM	0.43	0.87	0.94	0.72	0.76
NP&MBIC	0.31	0.77	0.88	0.58	0.62
Poisson&MBIC	0.54	0.73	0.85	0.61	0.77

TABLE 2.5 – Comparaison du HDP-HMM avec des méthodes de détection de changement standards sur un jeu de données d’instantants de changements labélisés. Score médian calculé sur les 50 séries RIPE Atlas labélisées. Méthodologie alternative avec les observations manquantes remplacées par la valeur précédente.

les méthodes de détection de changement standards. On constate une différence du temps d’exécution d’un ordre de grandeur dans les expérimentations de cette section : ~ 3 secondes avec le HDP-HMM, contre ~ 300 millisecondes avec les méthodes standards, pour une série de 2500 observations.

2.6.2 Test du rapport de vraisemblance

Dans la section précédente, nous avons comparé le DPMM et le HDP-HMM à des méthodes de détection de changement classiques. Nous avons montré que le HDP-HMM avec émissions DPMM est au moins aussi performant que les méthodes de détection classiques.

Autrement dit, le HDP-HMM détecte autant, ou plus, de changements que les méthodes classiques, sans augmenter le nombre de faux positifs. Au contraire, le DPMM fournit un nombre très élevé de faux positifs, et n’est donc pas adapté à la détection de changements dans les séries temporelles de délai. Rappelons également que le HDP-HMM permet d’inférer la séquence des états cachés, alors que les méthodes de détection classiques ne fournissent que des instants de changements. Connaître la séquence des états cachés permet de prédire les états futurs, ainsi que de reconnaître des états déjà observés dans le futur.

Dans cette section, nous comparons le DPMM au HDP-HMM gaussien, et le HDP-HMM gaussien au HDP-HMM avec émissions DPMM, à l’aide de tests d’hypothèses.

2.6.2.1 Hypothèses et interprétation

Le test du rapport de vraisemblance permet de comparer l’adéquation aux données de deux modèles. Deux hypothèses sont comparées. L’hypothèse nulle, notée H_0 , selon laquelle les données suivent un modèle de paramètres θ_0 . L’hypothèse alternative, notée H_1 , selon laquelle les données suivent un modèle de paramètres θ_1 .

On note $H_0 : \theta = \theta_0$, $H_1 : \theta = \theta_1$ et c le seuil de décision. Le test, aussi appelé « test de Neyman-Pearson », est défini ainsi :

$$\begin{cases} \text{rejeter } H_0 & \text{si } \Lambda < c \\ \text{ne pas rejeter } H_0 & \text{si } \Lambda > c \end{cases} \quad \text{avec } \Lambda = \frac{p(\mathbf{y} | \theta_0)}{p(\mathbf{y} | \theta_1)}. \quad (2.47)$$

Le seuil de décision c influe sur la probabilité de rejeter l'hypothèse nulle alors qu'elle est vraie, aussi appelée probabilité de fausse alarme, ainsi que sur la puissance du test. Les résultats sont généralement résumés dans une courbe ROC (*Receiver Operating Characteristic*).

Remarquons que le DPMM est un cas particulier du HDP-HMM où les probabilités de transitions sont identiques, quel que soit l'état. De même, le HDP-HMM gaussien est un cas particulier du HDP-HMM à émissions DPMM où il n'y a qu'une seule composante gaussienne par état. Nous réalisons ainsi deux tests : DPMM contre HDP-HMM gaussien, et HDP-HMM gaussien contre HDP-HMM à émissions DPMM.

2.6.2.2 Calibration

Pour chaque série temporelle \mathbf{y} nous cherchons le seuil c qui fournirait une probabilité de fausse alarme α si les données étaient distribuées selon H_0 . Nous procédons par simulation :

1. On estime à partir de \mathbf{y} les paramètres θ_0 et θ_1 des modèles correspondant aux deux hypothèses.
2. On génère $N = 200$ séries virtuelles $(\mathbf{y}'_n)_1^N$ de longueurs égales à celle de \mathbf{y} , d'après le modèle de paramètre θ_0 .
3. Pour chaque série virtuelle \mathbf{y}'_n , on calcule le ratio de vraisemblance $\Lambda_n = p(\mathbf{y}'_n | \theta_0) / p(\mathbf{y}'_n | \theta_1)$.
4. On cherche numériquement le seuil c tel que $\frac{1}{N} \sum_{n=1}^N \mathbb{1}_{\Lambda_n < c} = \alpha$.

2.6.2.3 Résultats

Nous considérons un jeu de données de 16000 séries temporelles représentant 7 jours de mesures (2520 observations), provenant de l'infrastructure de mesure RIPE Atlas. La construction du jeu de données sera décrite en détails dans la [section 3.2](#).

Dans le premier cas, nous comparons les deux hypothèses suivantes :

- H_0 : la série est distribuée selon un DPMM gaussien,
- H_1 : la série est distribuée selon un HDP-HMM gaussien.

Dans le second cas, nous comparons les deux hypothèses suivantes :

- H_0 : la série est distribuée selon un HDP-HMM gaussien,
- H_1 : la série est distribuée selon un HDP-HMM à émissions DPMM gaussien.

Les résultats pour différentes probabilités de fausse alarme sont présentés dans la [figure 2.10](#). Pour le premier test, avec $\alpha = 5\%$, le test rejette l'hypothèse nulle (HDP-HMM gaussien) dans 65% des cas. Pour le second test, avec $\alpha = 5\%$, le test rejette l'hypothèse nulle (DPMM) dans 95% des cas.

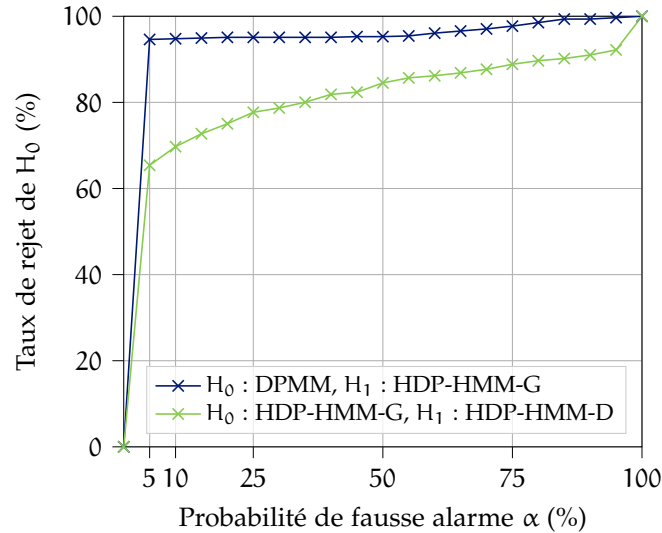


FIGURE 2.10 – Taux de rejet de H_0 en fonction de la probabilité de fausse alarme. HDP-HMM-G : HDP-HMM à émissions gaussiennes, HDP-HMM-D : HDP-HMM à émissions DPMM.

Ces résultats confirment que le HDP-HMM est un modèle préférable au DPMM pour modéliser les séries temporelles de délais. Le choix entre un modèle d'observation gaussien ou un modèle de mélange gaussien non paramétrique est moins marqué. Pour 30% des séries temporelles, il est aussi probable que les données soient générées par une loi normale que par un mélange gaussien. Cela peut s'expliquer par le fait qu'un grand nombre de séries sont simples, avec peu d'états ou des états qui sont bien modélisés par une loi normale.

2.6.3 Distribution des temps de séjour dans chaque état

Nous avons montré dans les deux sections précédentes que le HDP-HMM modélise mieux les données qu'un modèle de mélange infini (DPMM), sur une tâche de détection de changement, et à l'aide d'un test d'hypothèse. Une propriété des modèles de Markov cachés est que la distribution des temps de séjour dans un état suit une loi géométrique. Dans cette section nous comparons la distribution de la durée des segments inférés avec le HDP-HMM, à leur distribution théorique si les données étaient réellement générées par un modèle de Markov caché.

La probabilité d'obtenir d observations consécutives dans un état i d'un modèle de Markov caché est égale à [77] :

$$\mathbb{P}(z_t = i, \dots, z_{t+d} = i, z_{t+d+1} \neq i) = (\pi_{ii})^{d-1} (1 - \pi_{ii}). \quad (2.48)$$

Autrement dit, la distribution des temps de séjour dans un état i suit une loi géométrique de paramètre $1 - \pi_{ii}$. La durée des temps de séjour normalisée par la durée moyenne ($1/(1 - \pi_{ii})$) suit approximativement une loi exponentielle de moyenne 1. Dans cette section nous proposons de comparer les durées des segments inférés avec le HDP-HMM, avec la loi exponentielle.

Le diagramme quantile-quantile permet de comparer deux distributions. Si les deux distributions sont proches alors les points du diagramme suivront la ligne $y = x$. Nous considérons le même jeu de données de 16000 séries que dans la section précédente. La [figure 2.11](#) représente le diagramme quantile-quantile pour la durée des temps de séjour inférés avec le HDP-HMM.

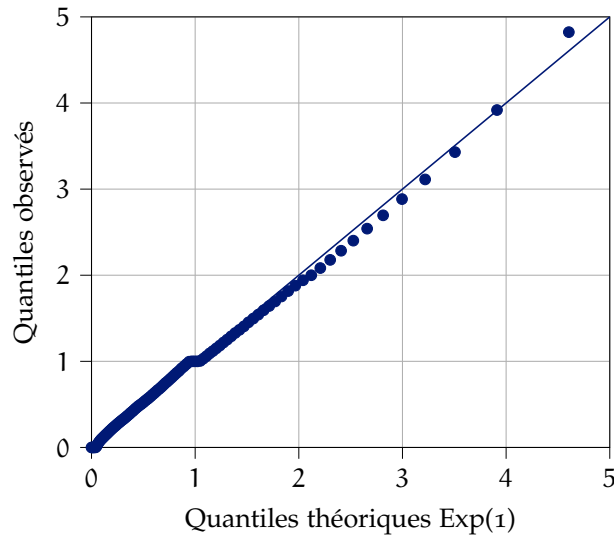


FIGURE 2.11 – Diagramme quantile-quantile des temps de séjours normalisés observés contre la loi exponentielle de moyenne 1. Les quantiles de 1 à 100 sont représentés par pas de 1.

On remarque que sur ce jeu de données RIPE Atlas que les deux distributions sont en accord. La propriété sans mémoire des états cachés semble donc être vérifiée. Dans le cas où la distribution des temps de séjour des données réelles ne suivrait pas une loi exponentielle, il existe une extension du HDP-HMM, le HDP-HSMM (*Hidden Semi-Markov Model*) [53], qui permet de modéliser des distributions plus générales.

2.7 DISCUSSION

Le HDP-HMM, en particulier avec le DPMM comme loi d'émission, est un modèle très flexible capable de s'adapter à un grand nombre de séries temporelles. Ce modèle permet de segmenter les séries temporelles, mais également de réaliser des prédictions sur les états cachés futurs grâce à la matrice de transition. Cette flexibilité vient au prix d'une complexité théorique et algorithmique plus importante que les modèles statistiques classiques. Remarquons cependant que la complexité se situe largement dans l'inférence des paramètres du modèle. Le modèle en tant que tel est très simple (HMM avec dépendance à 1-pas). Au vu des a priori très généraux choisis, il est même presque surprenant qu'un modèle aussi simple détecte aussi bien les changements dans les séries temporelles de délais.

D'un point de vue théorique le choix d'un modèle de mélange gaussien comme loi d'émission est critiquable, car la loi normale est à support dans \mathbb{R} tandis que les observations de délais ne sont jamais négatives et sont distribuées dans \mathbb{R}^+ . En pratique ce choix nous semble raisonnable pour trois raisons. 1) L'utilisation de lois normales simplifie l'inférence des paramètres par rapport à des lois plus complexes comme la loi gamma. 2) Un modèle de mélange gaussien peut approximer n'importe quelle distribution. 3) En fonction de la valeur des poids du mélange, ainsi que des moyennes et variances des différentes composantes, la probabilité de valeurs négatives peut être très faible.

Le choix d'un modèle bayésien non paramétrique peut également paraître surprenant à un moment où tous les regards sont tournés vers l'apprentissage profond et les réseaux de neurones. Dans le cadre de nos travaux, le HDP-HMM présente deux avantages par rapport aux réseaux de neurones : c'est un modèle facilement interprétable (matrice de transition, moyennes, variances ...), et sa structure markovienne nous permet d'exploiter les processus de décision markoviens pour l'optimisation de la mesure ([chapitre 4](#)). De plus, remarquons que méthodes bayésiennes non paramétriques et apprentissage profond sont complémentaires, permettant par exemple de créer des réseaux profonds s'adaptant à la complexité des données [75]. De façon similaire, HMM et réseaux de neurones peuvent être combinés pour prendre en compte les dépendances à long terme dans les données [103]. Nos travaux ne sont donc pas incompatibles avec l'apprentissage profond, et peuvent être étendus pour créer des modèles du délai plus avancés.

APPLICATION AU TRAITEMENT DE DONNÉES À GRANDE ÉCHELLE

De nos jours, les chercheurs et les opérateurs réseau bénéficient d'une quantité considérable de données sur la performance et la topologie d'Internet. Les plateformes de mesure à grande échelle telles que RIPE Atlas [98] ou CAIDA Ark [50] réalisent plusieurs millions de mesures de délai et de chemins chaque jour, en utilisant des sondes distribuées dans le monde. Ces mesures fournissent des informations sur la santé d'Internet, et sur son évolution au cours du temps. Afin d'exploiter ces grands jeux de données pour prédire et détecter les incidents ou les changements significatifs de la performance, il est nécessaire de recourir à des modèles statistiques ad hoc.

La disponibilité, en accès libre, d'une grande quantité de résultats de mesures entre un grand nombre de nœuds sur Internet est récente. Les modèles proposés avant la disponibilité de ces données (voir [section 2.2](#)) sont conçus et validés sur des petits jeux de mesures, typiquement entre des universités, qui ne représentent pas forcément Internet dans toute sa diversité. Il existe donc peu d'analyses à grande échelle des résultats fournis par les plateformes de mesure : d'une part car ces données n'étaient pas disponibles dans le passé, d'autre part car la variété des mesures entre différents points d'Internet requiert des modèles statistiques flexibles capables de s'adapter à la diversité des données.

Dans cette section nous proposons d'utiliser le HDP-HMM présenté dans le chapitre précédent afin d'automatiser le traitement d'un grand nombre de séries temporelles de délais. Dans la [section 3.1](#) nous rappelons les méthodes de mesure du délai et de la topologie sur Internet, ainsi que les principales plateformes de mesures. Dans la [section 3.2](#) nous appliquons le modèle à un grand nombre de séries temporelles RIPE Atlas et étudions la distribution de différents paramètres ainsi que la correspondance entre chemin réseau et état caché. Dans la [section 3.4](#) nous appliquons le modèle à la détection d'incidents sur de grandes infrastructures d'Internet. Enfin dans la [section 3.5](#) nous explorons l'application du modèle à d'autres types de mesures réseau telles que des mesures de congestion ou de temps de chargement de page web.

3.1 LA MESURE DE LA PERFORMANCE D'INTERNET

Internet est une interconnexion de réseaux informatiques communiquant avec le protocole IP (*Internet Protocol*). Ces réseaux sont regrou-

pés en systèmes autonomes (*Autonomous System* ou *AS*), qui contrôlent leur routage interne. Le routage entre les systèmes autonomes est effectué grâce au protocole BGP (*Border Gateway Protocol*). Les interconnexions entre les systèmes autonomes sont généralement réalisées dans des points d'échanges (*Internet eXchange Point* ou *IXP*). Les systèmes autonomes appartiennent généralement à des fournisseurs d'accès à Internet ou à de grandes entreprises (Google par exemple). La topologie et la politique de routage interne aux AS ne sont typiquement pas connues. La mesure d'Internet consiste à inférer des propriétés du réseau en réalisant des mesures à partir d'hôtes contrôlés. Par exemple, le délai entre ces deux hôtes, les routeurs traversés sur le chemin ou la bande passante disponible.

Dans cette thèse nous nous intéressons essentiellement aux mesures de délai. Comme nous le verrons dans la section suivante, le délai aller-retour est facile à mesurer et capture certains évènements affectant la performance du réseau comme les changements de routage ou de niveau de trafic. De plus, de grandes quantités de mesures historiques du délai sur Internet sont aujourd'hui disponibles en accès libre. D'autres métriques, telles que la bande passante disponible, sont plus compliquées à mesurer et peuvent être très bruitées et donc peu exploitables. Le délai permet donc d'étudier la performance et la santé du réseau à moindre coût.

3.1.1 Mesure du délai

Dans les réseaux informatiques, le délai comporte quatre composantes principales (figure 3.1). Le temps de traitement, de 100 ns à 10 µs, correspond au temps passé dans les couches logicielles (encapsulation TCP/IP/MAC). Le temps de transmission, de 10 µs à 100 µs, correspond au temps nécessaire pour envoyer les données sur le médium physique (conversion numérique/analogique, modulation ...). Le temps de propagation, de 100 µs à 100 ms ou plus, correspond au temps passé dans le médium physique. Il est fortement dépendant de la distance entre l'origine et la destination (dans une fibre optique, l'information se déplace à ~66% de la vitesse de la lumière). Enfin le temps d'attente, de zéro à plusieurs secondes correspond au temps passé dans les files d'attente de routeurs. Si le niveau de trafic est bas, alors ce délai est nul. Lorsque le niveau de trafic augmente, les files d'attente des routeurs se remplissent et ce délai augmente. Il peut être significatif lorsque les routeurs possèdent de grandes files d'attente, on parle alors de *bufferbloat* [41].

Sur Internet le temps de propagation et le temps d'attente sont prédominants, en raison de la distance géographique entre les hôtes, et de la présence de points de congestion sur le réseau. Ainsi, comme mis en avant dans la figure 2.1, les changements de délai observés sur Internet résultent souvent d'un changement dans le chemin phy-

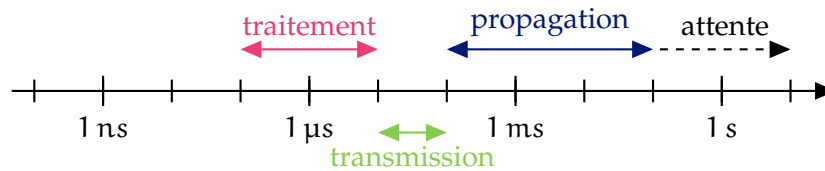


FIGURE 3.1 – Composantes du délai dans les réseaux informatiques. Échelle logarithmique. Le temps de propagation (limité par la vitesse de la lumière), et le temps d'attente dans les routeurs dominent le délai sur Internet.

sique emprunté par les paquets (changement de routage) ou d'un changement du niveau de trafic.

On distingue généralement deux types de mesure du délai sur Internet : le délai de bout en bout (*One-Way Delay* ou *OWD*) et le délai aller-retour (*Round-Trip Time* ou *RTT*). Le délai de bout en bout est la différence entre le temps d'émission du paquet et le temps de réception. Étant donné l'ordre de grandeur du délai (ms), sa mesure nécessite des horloges synchronisées précisément à l'origine et la destination (par exemple par GPS). Cela limite son applicabilité puisque la majorité des équipements sur Internet ne disposent pas de telles horloges. Le délai aller-retour est la différence entre le temps d'émission du paquet et le temps de réception par l'émetteur d'un acquittement de la destination. Il est typiquement mesuré avec l'utilitaire ping et le protocole ICMP (*Internet Control Message Protocol*).

La mesure du délai sur Internet est principalement affectée par deux phénomènes : le routage asymétrique, et l'équilibrage de charge. Dans le cas du délai aller-retour, la présence de routage asymétrique conduit les paquets à emprunter un chemin différent à l'aller et au retour. Il ne suffit donc pas de diviser par deux le délai aller-retour pour obtenir le délai de bout en bout. L'équilibrage de charge conduit quant à lui les paquets successifs à emprunter différents chemins. Deux mesures successives peuvent ainsi mesurer deux chemins différents. Une méthode permettant de garantir (sous certaines conditions) que deux mesures successives emprunteront le même chemin est présentée dans la section suivante.

3.1.2 Mesure des chemins

La mesure des chemins consiste à découvrir la topologie d'Internet à partir d'un hôte. La topologie peut-être inférée à plusieurs niveaux : au niveau des interfaces IP entre les routeurs, au niveau des routeurs, et au niveau des systèmes autonomes.

Les chemins sont le plus souvent mesurés au niveau des interfaces IP entre les routeurs, à l'aide de l'utilitaire traceroute. Un traceroute consiste à envoyer des paquets avec une durée de vie (*Time To Live* ou *TTL*) au niveau IP limitée de telle sorte à provoquer l'envoi d'un

message ICMP Time Exceeded par les routeurs, le TTL étant réduit de 1 par chaque routeur traversé sur le chemin. Ainsi un paquet est d'abord envoyé avec un TTL de 1, puis de 2, 3 et ainsi de suite jusqu'à recevoir une réponse de la destination. Les messages ICMP Time Exceeded ont pour adresse IP source l'interface du routeur faisant face à l'émetteur. Traceroute produit donc une liste d'interfaces traversées sur le réseau, et non de liens (paire d'interfaces) ou de routeurs. Traceroute est affecté par plusieurs problèmes décrits ci-dessous.

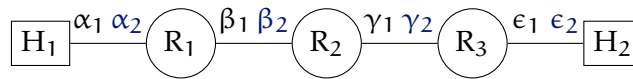


FIGURE 3.2 – Exemple de traceroute de H_1 vers H_2 sans équilibrage de charge. Les hôtes sont représentés par des rectangles, et les routeurs par des cercles. Les lettres grecques représentent les adresses IP de chaque interface.

Dans l'exemple de la figure 3.2 on s'attend à obtenir le chemin $(\alpha_2, \beta_2, \gamma_2, \epsilon_2)$. En présence de routeurs muets (par exemple R_2), certains sauts seront manquants : $(\alpha_2, *, \gamma_2, \epsilon_2)$. En cas de filtrage des sondes à un endroit du réseau ou en absence de réponse de la destination, le nombre de sauts sera inconnu : $(\alpha_2, *, *, *, \dots)$. Enfin en cas d'interface source mal configurée, des chemins impossibles ou des adresses IP privées (non uniques sur Internet) peuvent apparaître : $(\alpha_2, \beta_2, \text{privé}, \epsilon_2)$. Les différentes sources de données manquantes dans les traceroute sont détaillées dans [62].

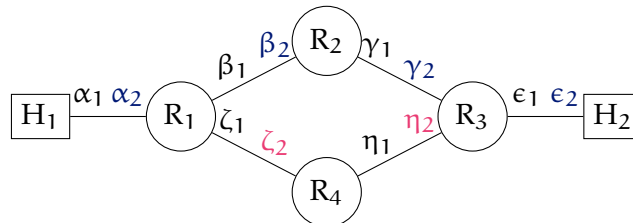


FIGURE 3.3 – Exemple de traceroute de H_1 vers H_2 en présence d'équilibrage de charge. Les hôtes sont représentés par des rectangles, et les routeurs par des cercles. Les lettres grecques représentent les adresses IP de chaque interface.

Dans l'exemple de la figure 3.3, le routeur R_4 a été ajouté. Afin d'équilibrer la charge, R_1 peut envoyer les paquets vers R_3 ou R_4 . Cet équilibrage de charge est souvent réalisé sur la base d'un identifiant de flot, contenant notamment les adresses IP et ports source et destination. Avec un identifiant de flot constant on s'attend à observer un des deux chemins suivants : $(\alpha_2, \beta_2, \gamma_2, \epsilon_2)$ ou $(\alpha_2, \zeta_2, \eta_2, \epsilon_2)$. Cependant, l'utilitaire traceroute classique ne conserve pas un identifiant de flot constant entre les sondes. Nous pourrions observer des chemins impossibles : $(\alpha_2, \zeta_2, \gamma_2, \epsilon_2)$ ou $(\alpha_2, \beta_2, \eta_2, \epsilon_2)$.

L'utilitaire `paris-traceroute` [10] a été développé afin d'adresser le problème de l'identifiant de flot non constant. Il conserve un identifiant de flot constant entre les sondes, et permet de découvrir les chemins multiples dans le réseau [11]. Récemment, jusqu'à 96 interfaces ont été observées pour un seul routeur sur Internet [107]. De façon similaire, l'utilitaire `tokyo-ping` [76] permet de réaliser des pings avec un identifiant de flot constant.

Remarquons enfin que `traceroute` n'identifie pas les interfaces en-dessous du niveau IP. Ainsi, les interfaces de niveau 2, et les tunnels qui encapsulent du trafic IP (IPsec, MPLS, ...) sont invisibles.

Comme nous venons de le voir, `traceroute` produit une liste d'interfaces traversées sur le réseau. Si l'on souhaite obtenir la topologie au niveau des routeurs, il est nécessaire de regrouper ensemble les interfaces appartenant à un même routeur. Dans la littérature, ce problème est appelé résolution d'alias (*alias resolution*). Les principales méthodes de résolution d'alias consistent à utiliser des données publiques sur la topologie d'Internet, ainsi que des heuristiques pour inférer les interfaces (adresses IP) correspondant à un même routeur [55, 56].

Si l'on souhaite obtenir la topologie au niveau des systèmes autonomes, il est nécessaire d'associer les adresses IP à leur système autonome d'origine. Une façon de résoudre ce problème consiste à récupérer les annonces du protocole de routage BGP (*Border Gateway Protocol*) utilisé pour le routage inter-AS sur Internet. Ces annonces contiennent un préfixe IP et un système autonome d'origine. À cet effet des « collecteurs de routes » disposés à plusieurs endroits d'Internet établissent des sessions BGP avec leurs voisins et enregistrent les annonces reçues. Des services comme `Routeviews` [82] ou `RIPE Routing Information Service` [80] fournissent ces données en accès libre.

Finalement, la tomographie [67] consiste à combiner les mesures de RTT et de routage afin d'inférer le délai au niveau de chaque lien, et de détecter et de localiser d'éventuels problèmes. Le problème peut-être formulé comme un problème inverse linéaire de la forme $\mathbf{Ax} = \mathbf{b}$ où \mathbf{b} est un vecteur contenant des mesures de RTT, \mathbf{x} est un vecteur représentant le délai sur chaque lien, et \mathbf{A} est une matrice de routage où $A_{ij} = 1$ si la mesure b_i passe par le lien x_j et 0 sinon. L'enjeu est de retrouver \mathbf{x} à partir de \mathbf{A} et \mathbf{b} . La principale difficulté quant à l'application de cette méthode sur Internet réside dans l'obtention de la matrice de routage. En effet, comme discuté ci-dessus, il est difficile d'obtenir des informations complètes et correctes sur le chemin avec `traceroute`.

3.1.3 Infrastructures de mesure publiques

Plusieurs projets existent pour créer des infrastructures de mesure à l'échelle d'Internet et centraliser les résultats. Parmi les projets les plus

connus nous pouvons citer RIPE Atlas [98], CAIDA Ark [50], M-Lab [32] et Planet Lab [24].

Dans nos travaux nous avons utilisé principalement des données RIPE Atlas, de par le grand nombre de mesures disponibles publiquement, la facilité d'utilisation de la plateforme, et la couverture des mesures [100]. RIPE Atlas utilise des sondes distribuées aux particuliers et aux entreprises pour réaliser des mesures vers des destinations arbitraires sur Internet. Les ancres sont un type particulier de sondes qui sont capables de réaliser plus de mesures, et qui sont typiquement situées dans des environnements stables (entreprise, centre de données). Le nombre de mesures manquantes liées à des déconnexions de sondes [95] est ainsi réduit. De plus les ancres réalisent en permanence des mesures entre elles dans une topologie de graphe complet (figure 3.4).

Ce système de mesure entre les ancres est appelé *anchoring mesh measurements*. Nous l'appellerons « système d'ancres » dans la suite de la thèse.

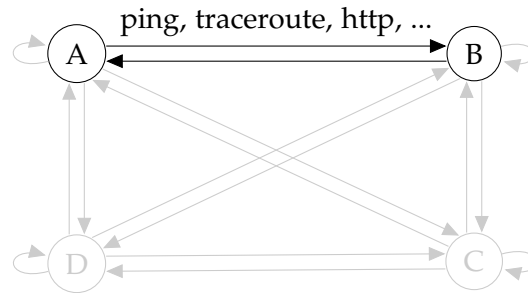


FIGURE 3.4 – Principe du système d'ancres Atlas avec 4 ancres. Les nœuds représentent les ancres et les arrêtes les mesures. Les mesures sont réalisés vers les ancres elle et entre chaque paires d'ancres, dans les deux sens. Pour n ancres, n^2 mesures de chaque type (ping, traceroute, etc.) sont réalisées. Fin mai 2020, le système d'ancres Atlas comprends 650 ancres.

Fin mai 2020, RIPE Atlas compte 11104 sondes et 650 ancres, pour 10500 résultats de mesures par seconde, soit plus de 331 milliards de résultats par an. La plateforme couvre 90% des pays, 5% des AS IPv4 et 8% des AS IPv6 [43]. Les mesures du système d'ancres réalisent 3 pings entre chaque paire d'ancres toutes les 4 minutes, et 3 traceroutes toutes les 15 minutes. `paris-traceroute` est utilisé, avec un identifiant de flot changé toutes les 15 minutes.

3.2 CARACTÉRISATION DES DONNÉES RIPE ATLAS

Le HDP-HMM nous permet de segmenter et de caractériser une grande variété de séries de mesures de délai. Dans cette section nous appliquons le HDP-HMM à un grand nombre de mesures de délai RIPE Atlas, et nous étudions les caractéristiques des modèles obtenus.

Nous commençons par étudier la correspondance entre état caché et chemin réseau. Nous étudions ensuite la distribution du nombre d'états cachés inférés sur les séries. Enfin nous étudions la relation entre variation du délai dans un état, et durée moyenne de l'état.

Nous considérons un sous-ensemble choisi aléatoirement de 10% des paires origine-destination du système d'ancres Atlas entre le 1 et le 7 février 2020, soit 16000 paires origine-destination. Nous considérons également les 541 paires qui correspondent aux mesures vers les ancrs elle-mêmes (origine et destination identiques). Pour chaque paire nous considérons les mesures de délai (ping), ainsi que les mesures de chemins (traceroute) qui nous permettront d'étudier la relation entre état caché et chemin réseau.

3.2.1 *Correspondance entre état caché et chemin réseau*

Nous faisons l'hypothèse que les états cachés correspondent au couple (chemin réseau, niveau de trafic). Dans cette section nous étudions la correspondance entre état caché et chemin réseau. Le nombre d'états cachés ne correspond pas nécessairement au nombre de chemins réseau observés pour deux raisons : certains changements de chemins ne causent pas de changement de délai (équilibre de charge entre deux liens non congestionnés par exemple), et des changements de délai peuvent apparaître sans changement de chemin (congestion par exemple). De plus, comme présenté précédemment, tous les changements de chemin n'apparaissent pas dans les mesures traceroute (changements en dessous de la couche IP par exemple).

Nous étudions donc la correspondance entre état caché et chemin réseau dans deux sens ([figure 3.5](#)) : « Combien de chemins différents apparaissent dans un état donné ? » et « Dans combien d'états différents un chemin donné apparaît-il ? ». Dans l'hypothèse où les changements de chemins causent des changements de délai, un état devrait correspondre à un petit nombre de chemins. Pour rappel, nous avons validé la capacité du HDP-HMM à détecter des changements significatifs de délai dans la [section 2.6.1](#).

Nous étudions la correspondance avec le chemin IP ([figure 3.6](#)) et le chemin AS ([figure 3.7](#)). En moyenne, un état est associé à 3.5 chemins IP (2.3 chemins AS) et un chemin IP est associé à 2.1 états (2.7 pour un chemin AS). Comme on peut s'y attendre, les états comprennent moins de chemins AS différents, mais les chemins AS sont associés à plus d'états en moyenne.

3.2.2 *Distribution du nombre d'états cachés*

Dans cette section nous nous intéressons à la distribution du nombre d'états cachés inférés dans le jeu de données Atlas. Nous étudions séparément les mesures vers les ancrs elles-mêmes (« auto mesures ») et

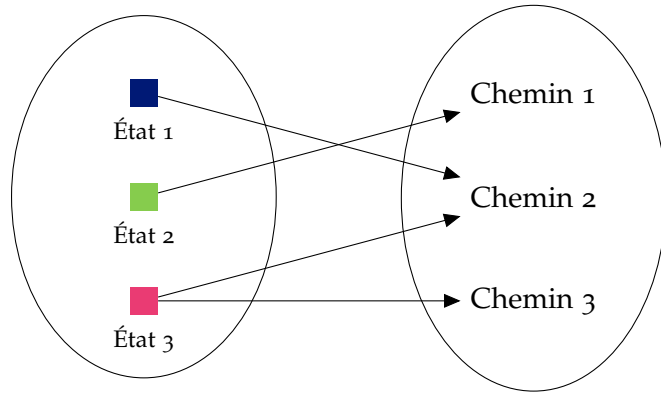


FIGURE 3.5 – Exemple de correspondance entre état inféré et chemin réseau (suite d’adresses IP ou de systèmes autonomes). Plusieurs chemins peuvent apparaître dans un état (cf. l’état 3), et un chemin peut apparaître dans plusieurs états (p. ex. le chemin 2). Dans l’hypothèse où les changements de chemins causent des changements de délai, un état devrait correspondre à un petit nombre de chemins.

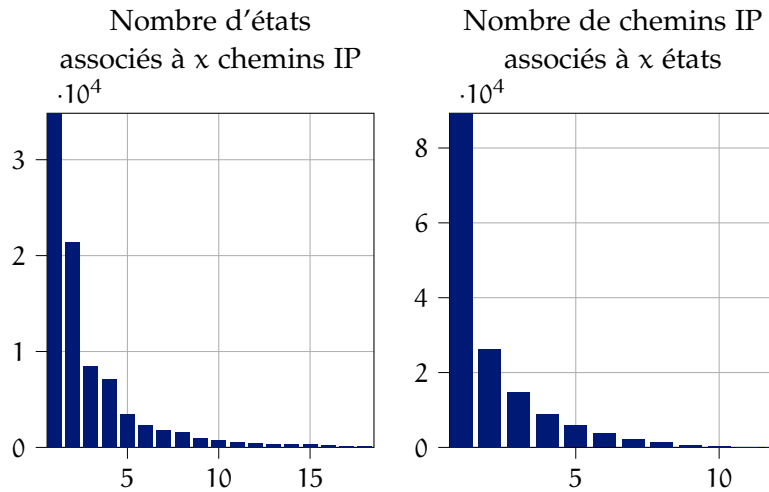


FIGURE 3.6 – Correspondance entre état inféré et chemin IP dans des mesures RIPE Atlas. En moyenne un état est associé à 3.5 chemins IP, et un chemin IP est associé à 2.1 états.

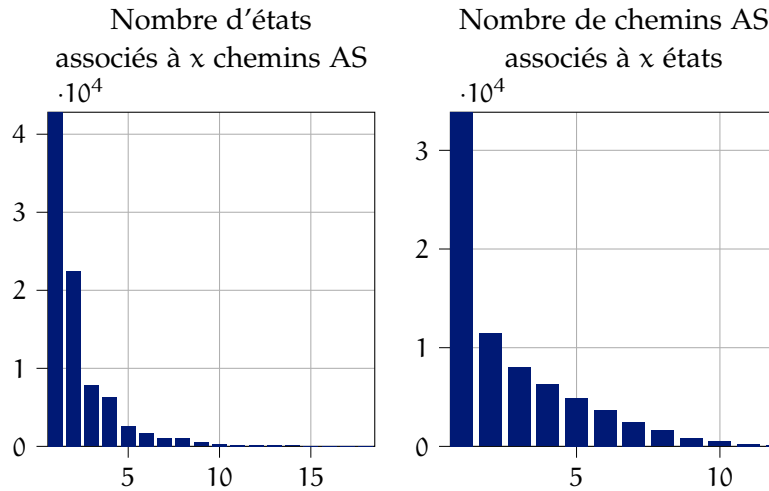


FIGURE 3.7 – Correspondance entre état inféré et chemin AS dans des mesures RIPE Atlas. En moyenne un état est associé à 2.3 chemins AS, et un chemin AS est associé à 2.7 états.

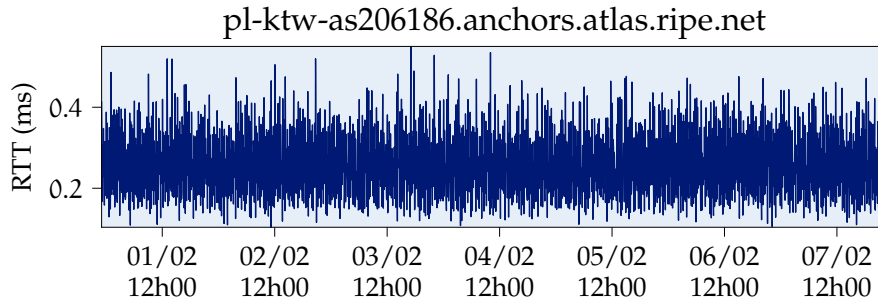
les mesures entre les ancres. Pour rappel, nous considérons des séries temporelles d'une durée d'une semaine (2520 résultats de mesures par série).

3.2.2.1 Auto mesures

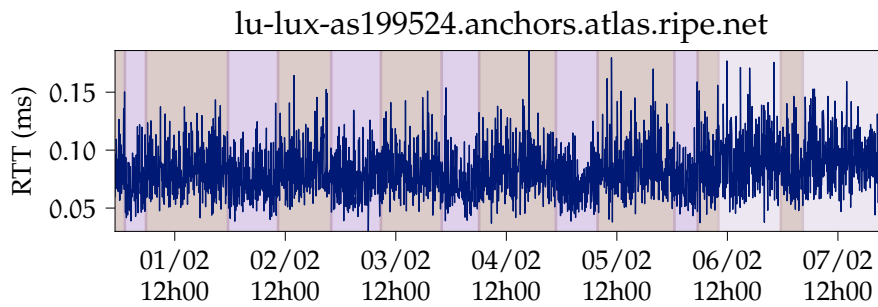
Dans le cas des auto mesures, 78% (422) des traces ont 1 état, 17% (95) ont 2 états, et les 5% (24) restants ont plus de 2 états. Ce résultat est surprenant puisque l'on peut s'attendre à ce que les mesures vers les ancres elles-mêmes soient constantes. La figure 3.8 présente des exemples de séries d'auto mesures avec différents nombres d'états.

Les 4 ancres pour lesquelles on infère 5 états sont des ancres virtuelles hébergées par Google dans 4 régions de Google Cloud Platform (6613 : us-central-1a, 6614 : us-east-1b, 6702 : eu-west-4, 6708 : eu-west-3). Les ancres virtuelles [27] ont été récemment introduites dans RIPE Atlas et se présentent sous la forme de machines virtuelles plutôt que de sondes physiques. Le délai moins stable que pour les autres ancres peut s'expliquer par le fait que l'ancre virtuelle partage ses ressources avec la machine hôte.

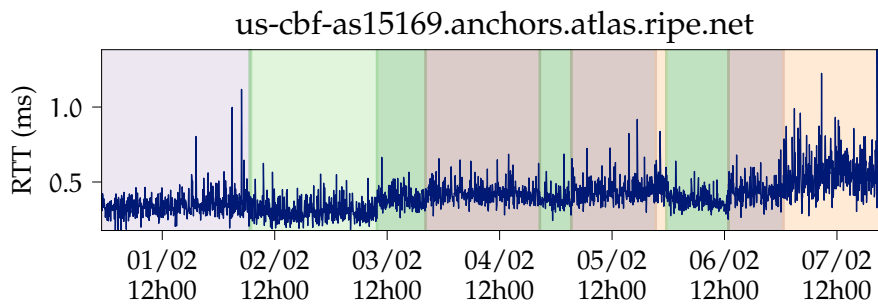
Grâce à l'a priori dépendant de la moyenne et de la variance des observations (section 2.5.3.2), le HDP-HMM est capable de segmenter des séries temporelles de façon invariante à l'échelle. Ici, le modèle est capable de segmenter des séries avec de très faibles variations de délai. Cette capacité du modèle est intéressante puisque l'échelle du délai varie grandement sur Internet en fonction de la distance entre l'origine et la destination.



(a) Série temporelle avec 1 état inféré.



(b) Série temporelle avec 3 états inférés.



(c) Série temporelle avec 5 états inférés.

FIGURE 3.8 – Exemples de segmentation de mesures vers les ancres elle-même.

3.2.2.2 Mesures vers les autres ancres

La [figure 3.9](#) représente la distribution du nombre d'états inférés pour les mesures vers les autres ancres, sur 1, 3, 7 et 14 jours.

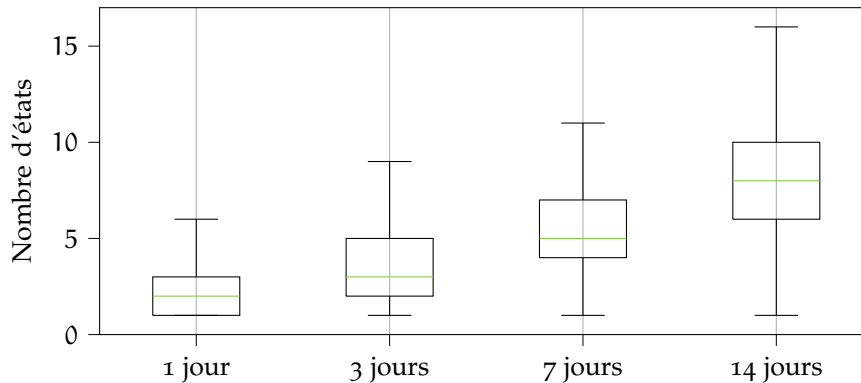


FIGURE 3.9 – Distribution du nombre d'états inféré en fonction de la durée d'observations. Le même jeu de données de 14 jours est tronqué à 7, 3 et 1 jour. La boîte est délimitée par le premier et le troisième quartile, la ligne verte représente la médiane, et les extrémités sont situées à $1.5(Q_3-Q_1)$.

Le nombre d'états appris augmente en fonction de la durée des séries temporelles. Le nombre médian d'états appris pour 1, 3, 7 et 14 jours est respectivement de 2, 3, 6 et 8.

Cela confirme la capacité du HDP-HMM à apprendre des modèles plus complexes à mesure que le nombre d'observations, et éventuellement le nombre de configurations réseau sous-jacentes, augmente.

3.2.3 Durée moyenne et coefficient de variation

Un avantage des HMMs par rapport à d'autres modèles de séries temporelles (par exemple, les modèles autorégressifs ou les réseaux de neurones) est que les paramètres peuvent être interprétés facilement, compte tenu du domaine d'application. Dans notre cas, la matrice de transition π nous renseigne sur la fréquence des changements de configuration du réseau et la relation entre eux, tandis que les distributions d'observation nous donnent notamment la valeur moyenne du délai et sa variance dans chaque état.

Sur la plupart des séries temporelles, nous pouvons distinguer deux types d'états : ceux où le délai est relativement constant (comme l'état saumon sur la [figure 3.11](#)), et les états où le délai est très variable (comme l'état rouge). On cherche donc ici à étudier la relation entre la durée moyenne d'un état, et les variations du délai dans celui-ci.

La durée moyenne d'un HMM dans un état i est donnée par la relation $1/(1 - \pi_{ii})$ où π_{ii} est la probabilité d'auto transition. Dans l'exemple de la [figure 3.11](#), la durée moyenne de l'état rouge est de 45 pas de temps (= 3 heures) et de 149.5 pas de temps (= 9 heures 58

minutes) pour l'état saumon. L'écart type du délai de l'état rouge est de 10.3 ms alors que l'écart type de l'état saumon est de 4.1 ms. Les états avec une variance élevée peuvent s'expliquer par l'équilibrage de charge intra domaine (puisque l'identifiant de flot des pings Atlas n'est pas constant), la congestion ou la présence d'équipements sur le chemin retardant le traitement des paquets ICMP.

Remarquons que l'identification précise de la cause des variations de délai sur Internet est rendue compliquée par le manque de visibilité sur les réseaux traversés. Par exemple, en l'absence de mesures réalisées directement sur les deux routeurs d'une interconnexion, il est difficile de confirmer qu'un changement de délai est dû à la congestion d'un lien. La classification des changements de délai en fonction des paramètres des états correspondants pourrait être étudiée dans des travaux futurs.

Afin d'étudier la relation entre durée et variation du délai sur un grand nombre de mesures Atlas, nous utilisons le coefficient de variation, défini comme l'écart type divisé par la moyenne. Cela nous permet d'obtenir une mesure de la variation du délai invariante à l'échelle des séries temporelles. En effet, le délai moyen varie de quelques millisecondes pour les mesures entre deux ancrés très proches, à plusieurs centaines de millisecondes pour des ancrés sur des continents différents. Une variation du délai de quelques millisecondes a donc plus d'importance entre deux ancrés proches, qu'entre deux ancrés éloignés. Le coefficient de variation permet de prendre en compte cette différence.

La [figure 3.10](#) donne le coefficient de variation du RTT par rapport à la durée moyenne dans un état. Dans l'ensemble de données analysé, la durée moyenne de l'état diminue à mesure que le coefficient de variation du RTT augmente. Cela n'est pas surprenant, car nous nous attendons à ce que les chemins Internet passent plus de temps dans des états stables.

3.3 SEGMENTATION INTÉGRÉE À RIPE ATLAS

Afin de rendre notre méthode plus largement accessible nous avons mis en place une API (*Application Programming Interface*) web publiquement accessible, permettant la segmentation de séries temporelles de mesures de délai RIPE Atlas. L'API est déployée dans l'infrastructure de mesure Atlas et réalise la segmentation des séries temporelles à la demande (< 10 secondes pour une semaine de mesures). La création et le déploiement de l'API sont le résultat d'une collaboration avec le RIPE NCC à Amsterdam en février 2019. Trois articles de blog détaillant la méthode de segmentation ainsi que l'API ont été publiés sur le site RIPE Labs [2, 69, 70]. Dans cette section nous détaillons l'architecture du service de segmentation, ainsi que les points de terminaisons de l'API.

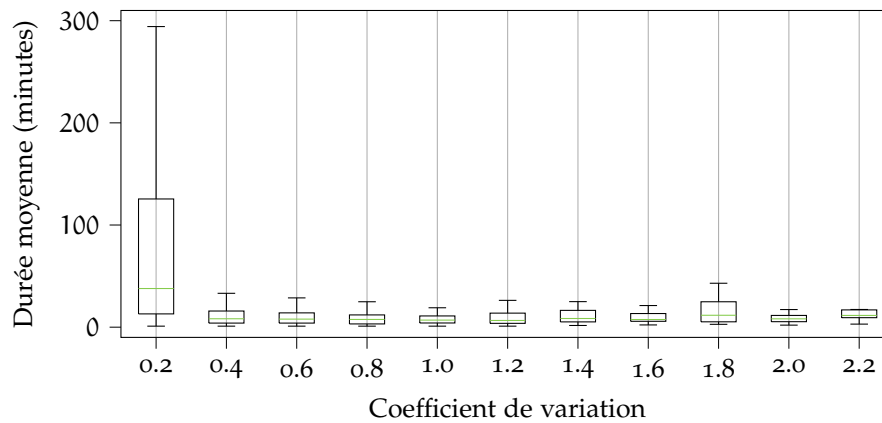
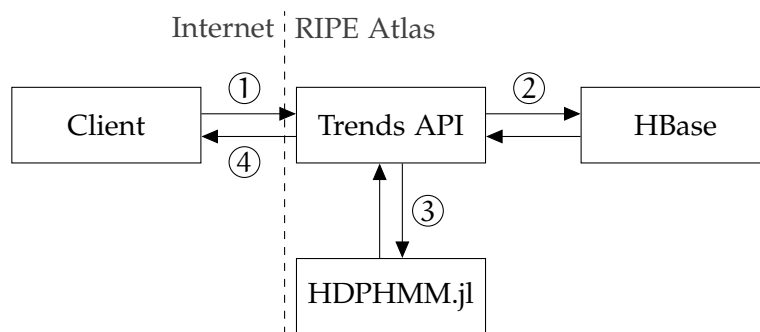


FIGURE 3.10 – Distribution de la durée moyenne des états en fonction du coefficient de variation (écart type / moyenne). 16739 séries RIPE Atlas entre le 1^{er} et le 7 février 2020.

3.3.1 Architecture

RIPE Atlas fournit une API web permettant de récupérer des résultats de mesures, étant donnée une date de début, une date de fin, une sonde, et une destination de mesure. L'objectif de la collaboration avec le RIPE NCC était de fournir une interface similaire, retournant la segmentation associée aux mesures. L'API, dénommée *trends* (tendance), est hébergée par le RIPE NCC, permettant ainsi un accès direct aux données de RIPE Atlas.

Le composant principal de l'API est la librairie Julia HPDHMM.jl, décrite dans la [section 2.5.2](#). Bien qu'il soit possible d'écrire des API web dans le langage de programmation Julia, ce dernier ne fait pas partie des langages supportés par le RIPE NCC pour écrire des services web, les deux langages principalement utilisés étant JavaScript et Python. Nous avons donc implémenté l'API web à l'aide de Node.js, un environnement d'exécution JavaScript pour les services web. L'API *trends* (Node.js) et la librairie HPDHMM.jl (Julia) communiquent en partageant directement la mémoire, ce qui évite la copie de données entre les deux environnements. Nous avons présenté ce procédé d'intégration de Julia et Node.js à l'édition 2020 de la *JuliaCon*, la conférence sur le langage de programmation Julia : *Integrate Julia and Javascript with Node.js extensions* [68].



RIPE Atlas utilise Apache HBase pour stocker les résultats de mesure. HBase est une base de données non-relationnelle, en opposition aux bases relationnelles tel que PostgreSQL, permettant de stocker et de requêter efficacement des tables contenant plusieurs millions de lignes et de colonnes.

Lorsque le client réalise une requête HTTP ①, l'API trends extrait et valide les paramètres (dates, identifiants de mesure) et réalise une requête HBase (avec le protocole Thrift) ② pour récupérer les mesures. Les mesures sont nettoyées (insertion des mesures manquantes, suppression des résultats en double), puis transmises à la librairie HDPHMM.jl pour segmentation ③. Finalement les résultats de segmentation sont mis en forme par l'API trends, puis la réponse HTTP est transmise au client ④.

3.3.2 Terminaisons

Étant donnée une paire origine-destination (identifiant de mesure et de sonde), l'API *trends* retourne la segmentation de la série temporelle de mesure de délai RIPE Atlas. L'API offre trois points de terminaisons, décrits dans le [tableau 3.1](#).

Méthode	Chemin	Paramètres
GET	/ticks/:msm_id/:prb_id	start, stop
GET	/trends/:msm_id/:prb_id	start, stop
GET	/trends/:msm_id/:prb_id/summary	start, stop

TABLE 3.1 – Terminaisons de l'API RIPE Atlas *trends*. Les paramètres start et stop sont des dates au format ISO.

La terminaison *ticks* retourne, pour une paire donnée, le RTT minimum « nettoyé » : les résultats en double dus à des problèmes de connectivité sont supprimés, et les résultats manquants sont explicitement insérés). La terminaison *trends* retourne le RTT minimum et la segmentation associée. Par exemple, l'URL <https://trends.atlas.ripe.net/api/v1/trends/1437285/6222/?start=2018-05-02&stop=2018-05-10> retourne la segmentation de la [figure 3.11](#) (le temps de segmentation

est inférieur à 10 secondes pour une semaine de données). Un résumé des séries temporelles (liste 3.1) peut aussi être demandé en ajoutant `/summary` au chemin. Les dates de début et de fin sont spécifiées au format ISO : `YYYY-MM-DDTHH:MM` où `THH:MM` est optionnel et correspond au début du jour par défaut.

Nous fournissons également des *notebooks* interactifs pour documenter et démontrer les fonctionnalités de l'API, et comparer différents modèles statistiques : <https://github.com/maxmouchet/atlas-trends-demo>.

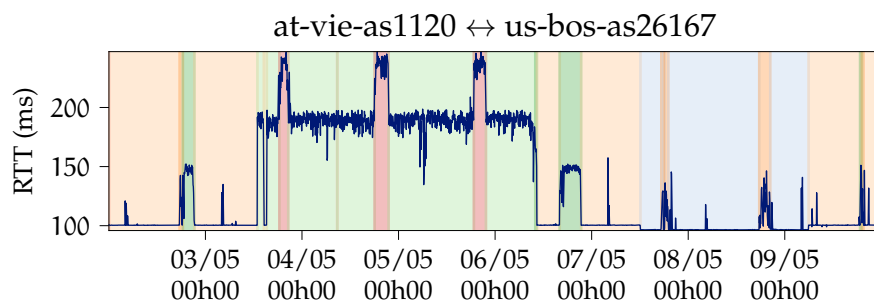


FIGURE 3.11 – Segmentation du délai sous la modélisation HDP-HMM entre deux ancres RIPE Atlas. Chaque couleur identifie un état.

```
{
  "statesCount": 7,
  "states": [
    {
      "id": 1,
      "rtt_summary": {
        "min": 95.882445,
        "max": 145.461727,
        "median": 96.529042
      }, // ...
    }, // ...
  ],
  "segments": [
    {
      "state": 3,
      "startTime": 1525219409,
      "stopTime": 1525282528,
      // ...
    }, // ...
  ]
}
```

LISTE 3.1 – Extrait de réponse au format JSON de l'API RIPE Atlas Trends.

3.4 DÉTECTION D'ANOMALIES SUR DES INFRASTRUCTURES D'INTERNET

RIPE Atlas possède des sondes dans 5% des systèmes autonomes IPv4 et 8% des systèmes autonomes IPv6 (section 3.1.3). Ce nombre peut paraître faible, mais en réalité un grand nombre de mesures traversent des systèmes autonomes importants, comme des fournisseurs de transit, et des points d'échange. Les infrastructures cœur d'Internet (transit, point d'échange) sont particulièrement représentées, tandis que les systèmes en bordure (fournisseurs d'accès) sont moins représentés. Ainsi, lorsqu'un incident se produit sur une grande infrastructure (panne de courant par exemple), les mesures de délai passant par ces infrastructures vont observer des changements de délai, voire des pertes de paquets.

Nous proposons d'utiliser les mesures de délai Atlas pour détecter des anomalies sur les grandes infrastructures d'Internet. En particulier nous proposons d'analyser la fréquence des changements d'état dans les séries de mesures segmentées, afin de détecter des changements de délai synchrones sur plusieurs paires origine-destination, potentiellement indicateurs d'anomalies.

Dans cette section nous commençons par discuter de la couverture d'Internet par les mesures RIPE Atlas. Ensuite, afin de valider notre méthode, nous analysons deux évènements déjà discutés dans la littérature, fournissant ainsi une vérité terrain contre laquelle vérifier nos résultats : l'incident d'AMS-IX en mai 2015 [1, 37, 42] et de DE-CIX Francfort en avril 2018 [3].

3.4.1 Couverture et choix des mesures

Le [tableau 3.2](#) représente les systèmes autonomes et les points d'échange les plus présents dans les traceroutes du système d'ancre Atlas à un instant donné.

Pour créer ce tableau nous avons récupéré l'ensemble des traceroutes du système d'ancre, le 2 février 2020 entre 00h et 00h30. Pour obtenir les systèmes autonomes présents dans les traceroutes, nous avons fait correspondre chaque adresse IP avec l'AS origine correspondant dans la table de routage d'Internet extraite de RouteViews. Pour obtenir les points d'échanges (IXP) présents dans les traceroutes, nous avons fait correspondre chaque adresse IP aux sous-réseaux d'interconnexion des IXP, extraits de PeeringDB.

Nous dénombrons 510 AS et 19 IXP visibles dans plus de 1000 paires. Un nombre important de paires origine-destination d'Atlas passent donc au travers d'importantes infrastructures d'Internet, telles que des points d'échanges (AMS-IX, DE-CIX, ...) et des fournisseurs de transit (Level 3 par exemple). RIPE Atlas fournit également des mesures vers les 13 serveurs DNS racine pour chaque sonde, mais ces

mesures sont plus difficiles à exploiter, pour détecter des anomalies dans le délai entre une origine et une destination, à cause du routage *anycast* employé par les serveurs DNS.

En effet, pour un serveur DNS racine, il peut exister plusieurs instances localisées à des endroits géographiques différents, qui partagent la même adresse IP. Ce type de routage, dit *anycast*, exploite le protocole de routage d'Internet (BGP), qui pour plusieurs routes vers une même destination préférera celle où le nombre de systèmes autonomes traversés est le plus faible. Ainsi, les clients sont dirigés vers le serveur DNS le moins loin d'eux, en nombre de systèmes autonomes. Cependant, cela rend ces mesures difficiles à exploiter puisqu'une même adresse IP peut correspondre à différents endroits géographiques.

Systeme autonome ou point d'échange	Numéro	Paires
TELIANET Telia Company AB	AS1299	47225 (13%)
NTT-COMMUNICATIONS-2914	AS2914	45363 (12%)
LEVEL3	AS3356	44949 (12%)
COGENT-174	AS174	42041 (11%)
HURRICANE	AS6939	31745 (8%)
DE-CIX Frankfurt		28639 (8%)
AMS-IX		23094 (6%)
LINX LON1		15163 (4%)
France-IX Paris		2661 (0.7%)
Equinix Singapore		2421 (0.7%)

TABLE 3.2 – Top 5 des systèmes autonomes et des points d'échanges présents dans les traceroutes du système d'ancres RIPE Atlas le 2 février 2020 entre 00h et 00h30.

Atlas fournit ainsi des informations très utiles pour permettre la détection d'anomalies dans le délai dans d'importantes infrastructures d'Internet. Il est cependant nécessaire de choisir judicieusement les paires origine-destination à utiliser pour surveiller une infrastructure, afin de limiter le coût de calcul. Notre approche consiste à choisir les paires origine-destination pour lesquelles l'infrastructure cible a été observée récemment (la veille, par exemple) dans les mesures de chemin (traceroute).

Bien que nous n'utilisions pas cette approche, il est possible de réduire le nombre de paires origine-destination à considérer en choisissant des paires « dissimilaires ». En effet, plusieurs paires peuvent emprunter le même chemin, ou des chemins très similaires. Une mesure de distance entre les paires permettant de choisir des paires dissimilaires est proposée dans [48] : plus l'intersection entre les ensembles d'adresses IPs apparaissant dans les mesures traceroute de

chacune des paires est grand, plus les deux paires sont considérées comme similaires.

3.4.2 Incident du 13 mai 2015 à AMS-IX

Le 13 mai 2015, une boucle de niveau 2 (tempête de diffusion) sur un commutateur a saturé le réseau d'interconnexion du point d'échange AMS-IX situé à Amsterdam [1]. L'évènement a duré 7 minutes et 2 secondes, de 10 h 22 min 12 s à 10 h 29 min 14 s (GMT), avant que l'interface du commutateur ne soit déconnectée. Lors de cet incident, plusieurs pairs localisés à AMS-IX ont perdu leurs sessions BGP. Cet incident est étudié dans [1] en analysant le pourcentage de paires origine-destination passant par le réseau d'interconnexion d'AMS-IX au cours du temps, d'après les mesures traceroute.

Dans cette section nous étudions le problème en utilisant les mesures de délai, au lieu des mesures de chemin traceroute. Comme nous l'avons vu dans la [section 3.2.1](#), les changements de chemin IP résultent souvent en des changements de délai, et le HDP-HMM permet de détecter les changements significatifs de délai dans les séries temporelles de mesures de délai. Nous avons analysé 74400 paires origine-destination pour lesquelles le réseau d'interconnexion d'AMS-IX a été vu au-moins une fois dans les mesures traceroute la veille de l'incident. Pour chaque paire origine-destination nous avons segmenté les séries de mesures de délai, et extrait les instants de changements d'états. Cette analyse prends 2h30 en traitant 4 séries temporelles en parallèle.

Par défaut, RIPE Atlas réalise des mesures de délai toutes les 4 minutes, dans un intervalle de 2 minutes afin de maximiser la couverture temporelle sur toutes les sondes participant à la mesure. Nous avons ainsi regroupé les instants de changement dans des groupes de 6 minutes. La fréquence de changement résultante est présentée dans la [figure 3.12](#). La durée réelle de l'incident est représentée en rouge. L'incident correspond clairement à une augmentation du nombre de changements. La fréquence reste d'abord haute pendant plusieurs heures, le temps que tous les pairs se replient sur des chemins alternatifs, et que certains reviennent sur AMS-IX (voir [1] pour plus de détails sur l'incident).

Nous observons également un pic entre 14 h 45 et 15 h. Après investigation nous avons observé que la quasi-totalité des changements qui ont eu lieu dans cette période sont liés à des mesures visant le serveur DNS racine A. Nous avons répété une procédure similaire pour toutes les paires origine-destination d'Atlas vers ce serveur DNS. Nous avons observé un pic similaire, cependant tous les AS sources étaient affectés de manière égale, ce qui nous laisse à penser que le pic a été causé par un évènement proche (dans la topologie du réseau) de l'une des instances du serveur DNS.

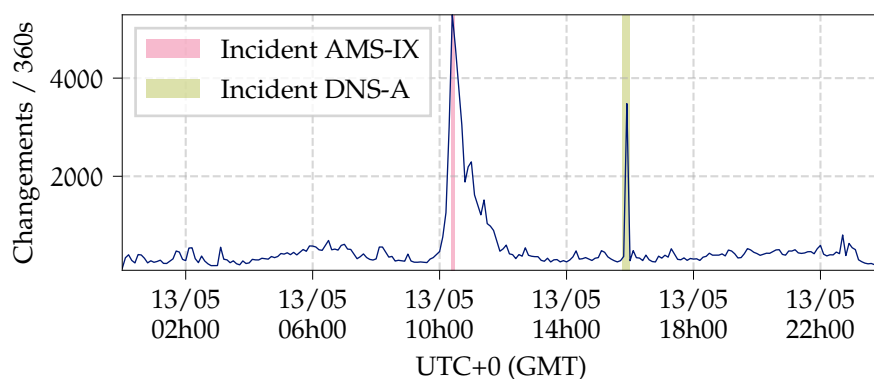


FIGURE 3.12 – Fréquence de changements le 13 mai 2015 pour les 20k paires qui ont vu AMS-IX dans leurs traceroutes la veille.

3.4.3 Incident du 9 avril 2018 à DE-CIX

Entre le 9 et le 10 avril 2018, certains pairs situés au point d'échange DE-CIX à Francfort, ont perdu leur connectivité avec les serveurs de route du point d'échange [3]. Cet incident a causé un re routage du trafic vers d'autres interconnexions, ou une interruption du trafic. Une analyse des taux de mise à jour BGP reçus par les collecteurs de route localisés à DE-CIX a montré que le taux de mise à jour a fortement diminué entre 19 h 43 et 23 h 28 le 9 avril, et entre 2 h 02 et 3 h 51 le 10 avril. En appliquant la même méthodologie que pour l'incident d'AMS-IX, la fréquence des changements d'état est présentée dans la [figure 3.13](#). Les deux pics les plus importants correspondent exactement aux deux instants où le taux de mise à jour BGP tend vers zéro. Les deux pics plus petits correspondent aux deux instants où les collecteurs ont à nouveau reçu des mises à jour.

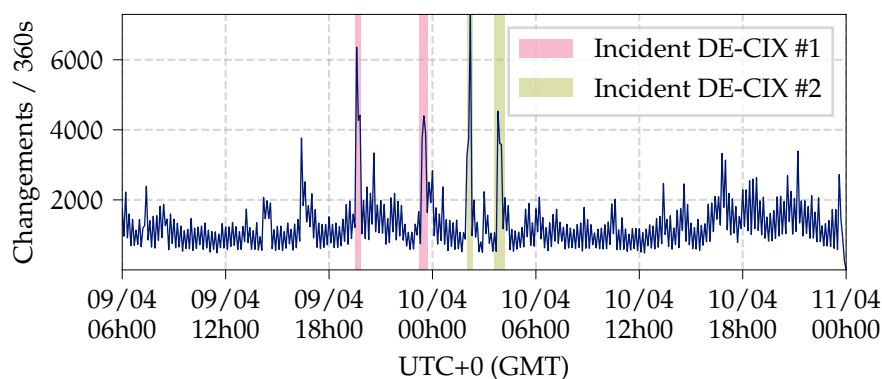


FIGURE 3.13 – Fréquence de changements entre le 9 et le 10 avril 2018 pour les 60k paires qui ont vu DE-CIX Francfort dans leurs traceroutes la veille.

3.5 AUTRES TYPES DE MESURES RÉSEAU

3.5.1 Mesures de congestion CAIDA MANIC

Le projet CAIDA MANIC (*Measurement and ANalysis of Internet Congestion*) [65] est un projet d'analyse de la congestion inter domaine (entre les systèmes autonomes) sur Internet. La méthode TSLP (*Time Series Latency Probing*) [64] est utilisée pour mesurer la congestion inter domaine [29]. Une fois qu'un lien d'appairage entre deux AS a été identifié, des sondes ICMP sont envoyées à l'extrémité proche (le dernier routeur dans le premier AS) et à l'extrémité distante (le premier routeur dans le deuxième AS) de la liaison (figure 3.14). L'intuition est qu'en présence de congestion les files d'attente des routeurs se rempliront et le délai entre l'extrémité proche et l'extrémité distante augmentera. En utilisant le même modèle que pour les séries de RTT RIPE Atlas, nous segmentons la série temporelle de différence de délai (extrémité éloignée - extrémité proche) à partir de mesures accessibles au public.

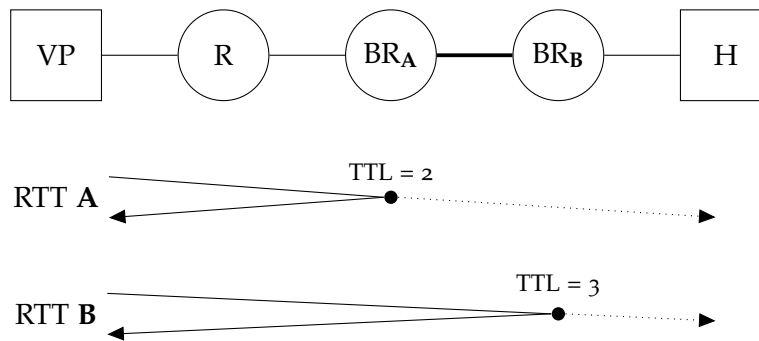


FIGURE 3.14 – Méthode *Time Series Latency Probing* pour estimer le délai entre deux routeurs de bordure d'AS. VP : sonde (*Vantage Point*), R : routeur intermédiaire sur le chemin, BR : routeurs de bordure, H : destination.

Dans la figure 3.15, nous montrons la segmentation résultant pour un lien d'appairage subissant une congestion périodique. Trois états sont appris. L'état vert, correspondant à une liaison non saturée, a un écart-type de 0,1 ms, tandis que les écarts-types pour les états rouge et bleu sont respectivement de 7 ms et 11 ms. L'état bleu semble correspondre à un état de « précongestion », où le niveau de trafic est particulièrement élevé, tandis que l'état rouge semble correspondre à un lien saturé. Étant donné que le modèle tient compte des dépendances temporelles, il est capable de séparer clairement ces deux états même si leurs distributions se chevauchent (figure 3.16).

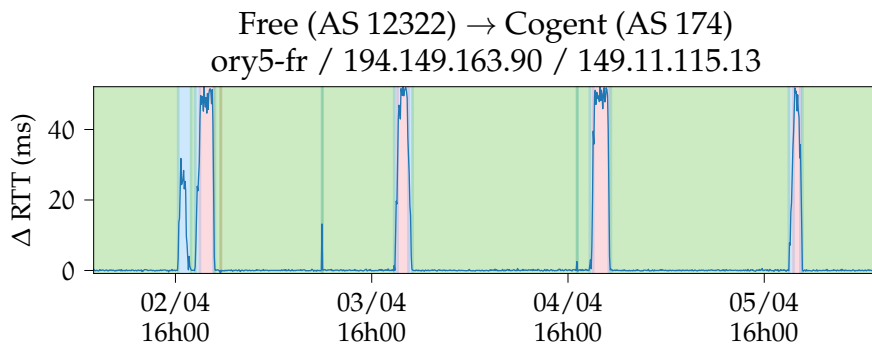


FIGURE 3.15 – Segmentation d’une série temporelle de différence RTT (lointain - proche) obtenue avec des sondes TSLP du projet CAIDA MANIC. Chaque couleur représente un état caché.

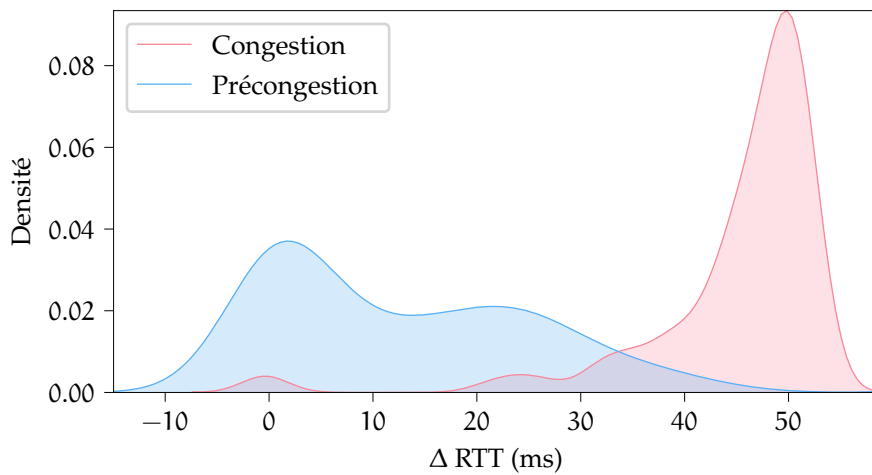


FIGURE 3.16 – Densité de probabilité des états de précongestion et de congestion sur une série de différence temporelle de RTT segmentée.

3.5.2 Mesures de temps de chargement Orange Web View

Finalement, nous avons testé l'utilisation de notre méthode de segmentation avec des mesures de temps de chargement de pages web (PLT), obtenus à partir de la plateforme Orange Web View [74, 88]. Web View est une plateforme de mesure qui simule des sessions de navigation sur des sites web et mesure différentes métriques de qualité de service. La figure 3.17 montre le PLT du site Web baidu.com mesuré à partir d'une sonde située à Lannion, en France. Remarquons que l'intervalle d'échantillonnage de ces données (6h) est très différent de celui des données RIPE Atlas (4 minutes).

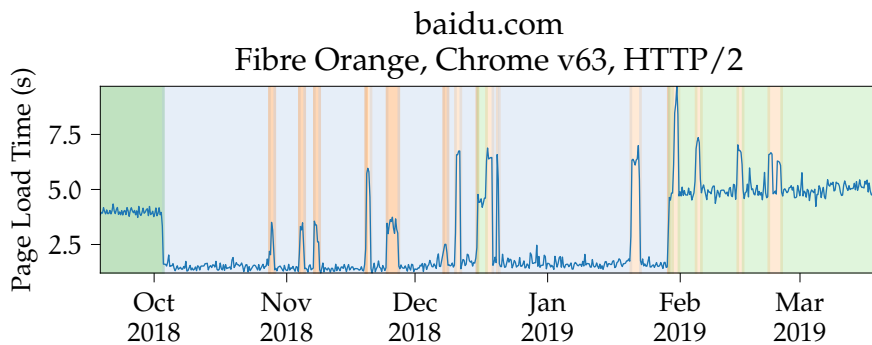


FIGURE 3.17 – Segmentation d'une série temporelle de temps de chargement de page obtenue à partir d'une sonde Web View. Chaque couleur représente un état caché.

Dans [87, ch. 5] le PLT est utilisé pour prédire la qualité d'expérience perçue par l'utilisateur sur un site web. L'auteur remarque que la segmentation des séries de mesures de PLT à l'aide du HDP-HMM fournit des caractéristiques supplémentaires (séquences des états cachés, matrice de transition) qui permettent d'améliorer la prédiction de la qualité d'expérience.

MÉTROLOGIE PARCIMONIEUSE AVEC DES PROCESSUS DE DÉCISION MARKOVIENS

4.1 PROBLÈME

La surveillance des réseaux de communication (*network monitoring*) conduit parfois à construire une représentation centralisée de l'état du réseau, à partir d'indicateurs de santé et de performance. C'est une composante essentielle de l'administration des réseaux qui permet notamment d'optimiser l'allocation des ressources et de détecter les incidents. La résolution temporelle et spatiale de cette représentation dépend de la fréquence de mesure des indicateurs, et de leur couverture. Les mesures peuvent être réalisées de manière passive, en observant le trafic, ou de manière active, en injectant des données dans le réseau. Les méthodes de surveillance doivent donc être capables de gérer un grand nombre de ressources et de métriques sans surcharger les ressources réseau. L'échelle et la complexité grandissante des réseaux de communication requièrent l'utilisation de méthodes de surveillance « intelligentes » (ou parcimonieuses) qui limitent le coût de surveillance tout en conservant la capacité de prendre des décisions opérationnelles informées.

Les méthodes de métrologie parcimonieuse sont particulièrement utiles dans les *overlays de routage* [5, 34, 86]. Un *overlay de routage* (figure 4.1) est composé d'hôtes déployés à différents endroits d'Internet. Les nœuds de l'overlay surveillent la qualité des chemins Internet entre eux, et coopèrent pour transmettre le trafic de n'importe quel autre nœud. En ajoutant des sauts intermédiaires dans le chemin pris par les paquets, les nœuds peuvent influencer le chemin global, sans modifier le mécanisme IP sous-jacent utilisé pour calculer les routes. Cette approche permet de construire des *overlays* qui se réparent et s'optimisent seuls [22]. Ces *overlays* sont capables de surveiller la qualité des chemins Internet entre leurs nœuds et d'adapter le routage en fonction de métriques dépendantes des applications, et des observations du réseau sous-jacent.

La plupart des technologies existantes surveillent l'ensemble des paires de nœuds. Cette approche a pour avantage de garantir le choix optimal des chemins. Cependant, elle requiert un coût de mesure de l'ordre de $\mathcal{O}(n^2)$ quand le nombre de nœuds n augmente, puisque la topologie de l'overlay de routage est celle d'un graphe complet. Le coût peut rapidement devenir prohibitif [23], en particulier si d'autres métriques de performances doivent être considérées telles que la bande passante.

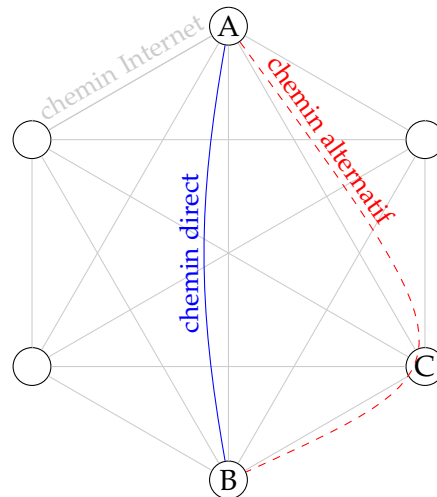


FIGURE 4.1 – Overlay de routage avec 6 nœuds et 15 arrêtes (chemins directs). Un chemin alternatif est construit en transmettant les paquets par un ou plusieurs nœuds intermédiaires.

Les méthodes de métrologie parcimonieuse peuvent également être utiles dans des applications de partage de charge dans des réseaux étendus à définition logicielle (*software-defined wide area network* ou *SD-WAN*). On peut par exemple considérer le cas de plusieurs liens satellites opérant sur des fréquences distinctes, et où l'on choisit dynamiquement le lien avec la plus grande bande passante disponible. Cette performance varie au cours du temps, et doit donc être surveillée. Cependant, étant donnée la capacité limitée de ces liens, une mesure trop fréquente pourrait perturber le bon fonctionnement du réseau. Une méthode de métrologie parcimonieuse permettrait alors de réduire le nombre de mesures, sans trop impacter le choix optimal du lien de communication.

Dans ce chapitre, nous proposons une stratégie de surveillance et de routage qui permet de construire des overlays de routage avec un grand nombre de nœuds sans surcharger le réseau de mesures. Notre méthode est parcimonieuse dans le temps et dans l'espace. Elle décide à chaque instant, quels chemins doivent être mesurés, au lieu de mesurer tous les chemins systématiquement. Comme nous allons le montrer sur un overlay comportant 30 nœuds simulés avec des données réelles, il est possible de réduire le coût de mesure de 91% en moyenne, au prix d'une augmentation moyenne de seulement 0.07% du délai par rapport à une stratégie de mesure de toutes les paires.

Plusieurs approches permettent d'obtenir une stratégie de surveillance parcimonieuse. Dans la section 4.4 nous proposons une première solution qui minimise le coût de mesure à chaque pas de temps, sans prendre en compte l'impact de la mesure à l'instant t sur les mesures futures, dans le cas de 1 et 2 chemins stochastiques. Dans

la section 4.5 nous généralisons cette solution à un nombre illimité de chemins et en prenant en compte l'impact à long terme des mesures.

4.2 TRAVAUX CONNEXES

Les réseaux overlay sont utilisés dans différentes applications, comme l'optimisation des réseaux de diffusion de contenu [6, 97], la protection contre les attaques par déni de service (DoS) [99, 108], et l'optimisation du routage.

Les premiers travaux montrant l'intérêt des overlays pour l'optimisation du routage sont Detour [86] et RON (*Resilient Overlay Network*) [5], introduits en 1999 et 2001. De nombreux travaux plus récents s'en sont inspirés [49, 66, 71, 78]. Tous ces travaux considèrent que la performance des chemins est mesurée en permanence entre tous les nœuds de l'overlay, selon une certaine métrique dépendante de l'application considérée.

Une mesure entre toutes les paires garantit l'optimalité du routage, puisque la performance des chemins est parfaitement connue. Cependant, le coût de mesure croît de façon quadratique avec le nombre de nœuds, puisqu'il existe $\mathcal{O}(n^2)$ paires dans un overlay de n nœuds. RON est capable de supporter 50 nœuds avant que le coût de mesure ne devienne prohibitif [5].

Le problème de la réduction du nombre de mesure dans les overlays de routage n'a été traité que récemment. Dans [21] une approche basée sur un réseau de neurone permet d'obtenir un routage quasi-optimal, sans nécessiter la mesure de l'ensemble des chemins en permanence. Dans [22] un résultat similaire est obtenu avec l'algorithme EXP₃ [9].

Dans ces deux travaux [21, 22], on suppose qu'un nombre prédéterminé de chemins sont mesurés à chaque instant, et l'objectif est de déterminer le sous-ensemble de chemins à mesurer pour minimiser le coût de routage. Dans ce chapitre, nous proposons une méthode qui fournit un compromis optimal entre le coût de mesure, et la qualité du routage, sans contrainte sur le budget de mesure.

4.3 OPTIMISATION DU ROUTAGE

Considérons l'overlay de routage de la figure 4.1. Les nœuds peuvent communiquer par plusieurs chemins, dont la longueur (le délai, par exemple) peut évoluer au cours du temps. À chaque instant les nœuds souhaitent choisir le chemin le plus court, ou le plus rapide, selon une certaine métrique (le délai par exemple). De plus, on suppose que la mesure de la longueur du chemin est coûteuse en ressources (réseau ou de calcul). Les nœuds souhaitent donc minimiser le nombre de mesures qu'ils réalisent, tout en continuant à choisir le chemin de longueur minimale.

Plus les performances des chemins sont stables dans le temps, plus il est possible de réduire le nombre de mesures. En effet, si la longueur d'un chemin change toutes les heures, alors il est inutile de mesurer toutes les minutes.

Lorsqu'ils ne mesurent pas un chemin, les nœuds doivent être capables de prédire sa longueur à l'aide d'un modèle statistique. Ce modèle est mis à jour en fonction des mesures passées. Afin de formaliser le problème, notons $M^i(t)$ la décision de mesure prise pour le chemin i à l'instant t , et $L^i(t)$ la longueur du chemin i au temps t . $M^i(t)$ est égal à 1 lorsque l'on mesure le chemin i , et 0 sinon. La connaissance disponible sur un chemin, après l'éventuelle mesure de $L^i(t)$, est alors représentée par l'ensemble

$$I_t^i = \sigma\{L^i(u) : M^i(u) = 1, u \leq t\} \quad (4.1)$$

où σ représente une tribu. L'ensemble I_t^i représente la connaissance passée, incluant la mesure au temps t si $M^i(t) = 1$, utile pour prédire les valeurs futures du chemin i . Les nœuds choisissent alors le chemin dont la longueur espérée, conditionnellement aux mesures passées, est la plus courte :

$$C(t) = \underset{i}{\operatorname{argmin}} \mathbb{E}(L^i(t) \mid I_t^i). \quad (4.2)$$

Dans la suite de ce chapitre nous considérons le délai comme métrique de longueur des chemins. La grande quantité de données disponibles en accès libre nous permettra de valider notre approche. Nous devons alors répondre à deux questions : « Quel modèle statistique choisir, et comment calculer l'espérance conditionnelle du délai ? » et « Comment choisir la décision de mesure M afin de limiter le nombre de mesures, sans trop impacter la qualité du choix du chemin le plus court ? ».

REMARQUE SUR LES NOTATIONS Des notations similaires sont couramment utilisées dans le domaine des statistiques bayésiennes, et dans le domaine du contrôle stochastique, pour faire référence à des concepts différents. Afin de rester proche des notations habituellement utilisées dans chaque domaine, certaines notations de ce chapitre sont différentes de celles utilisées dans les [chapitres 2](#) et [3](#). Notamment, la matrice de transition sera dénotée P et non plus π , et π dénotera une politique de décision. En cas de doute, les notations dans chaque contexte sont explicitées au début de la thèse.

4.3.1 Filtrage et prédiction

Comme nous l'avons vu dans le [chapitre 2](#), le délai sur Internet peut être modélisé de manière fiable par un modèle de Markov caché. Le délai est alors fonction d'un état caché qui évolue au cours du

temps selon une chaîne de Markov. La modélisation markovienne est intéressante dans le cadre de la métrologie parcimonieuse, car elle permet de faire des prédictions (grâce à la matrice de transition) et, comme nous le verrons, elle nous permet d'exploiter les processus de décision markoviens afin de proposer une solution générale au problème de métrologie parcimonieuse.

4.3.1.1 Chaîne de Markov

Considérons d'abord le cas où le délai $L^i(t)$ est parfaitement observé. Le délai peut prendre K^i valeurs différentes qui correspondent chacune à un état d'une chaîne de Markov. On note l_j^i le délai du chemin i dans l'état $j \in \{1, \dots, K^i\}$. Le filtre $\gamma_{t,t}^i$ est un vecteur ligne à K^i éléments, dont l'élément j donne la probabilité que le chemin i soit dans l'état j , étant donnée la connaissance accumulée au temps t . Le premier indice t fait référence à I_t^i , et le second indice fait référence à $L^i(t)$. Si l'on vient de mesurer, alors on connaît exactement l'état. Sinon, il suffit de multiplier notre croyance à l'instant précédent par la matrice de transition :

$$\gamma_{t,t}^{(i)}(j) = \mathbb{P}(L^i(t) = l_j^i | I_t) = \begin{cases} \mathbb{1}_{L^i(t)=l_j^i} & \text{si } M^i(t) = 1 \\ \left(\gamma_{t-1,t-1}^i P^{(i)}\right)_j & \text{si } M^i(t) = 0 \end{cases}. \quad (4.3)$$

Le prédictor $\gamma_{t,t+u}^i(j) = \mathbb{P}(L^i(t+u) = l_j^i | I_t)$ donne la probabilité des états u pas en avant :

$$\begin{aligned} \gamma_{t,t+u}^{(i)} &= \gamma_{t,t}^{(i)} \left(P^{(i)}\right)^u \\ \gamma_{t,t+1}^{(i)} &= \gamma_{t,t}^{(i)} P^{(i)}. \end{aligned} \quad (4.4)$$

L'espérance du délai à l'instant t , conditionnellement aux mesures passées, est alors donnée par :

$$\bar{L}^i(t) = \mathbb{E}(L^i(t) | I_t^i) = \sum_j l_j^i \gamma_{t,t}^{(i)}(j). \quad (4.5)$$

4.3.1.2 Modèle de Markov caché

Dans le cas d'un modèle de Markov caché, le délai est doublement aléatoire.

À chaque instant, l'état de la chaîne de Markov sous-jacente évolue de façon aléatoire, et le délai est aléatoire conditionnellement à cet état. Plus formellement, le délai est distribué selon la loi $p(L^i(t) | S^i(t))$, où $S^i(t)$ est l'état au temps t de la chaîne de Markov sous-jacente du chemin i . De plus, on associe à chaque état caché la valeur moyenne du délai dans cet état : $l_j^i = \mathbb{E}(L^i(t) | S^i(t) = j)$. Ainsi, réaliser une mesure ne nous indique plus directement l'état : il est dit « caché ».

Le filtre $\gamma_{t,t}^i$ donne alors la probabilité d'être dans un état donné, conditionnellement aux mesures passées : $\gamma_{t,t}^i(j) = \mathbb{P}(S^i(t) = j \mid I_t^i)$.

Lorsqu'on ne mesure pas ($M^i(t) = 0$) alors le filtre est mis à jour de la même façon que dans le cas d'une chaîne de Markov (équation 4.3) : $\gamma_{t,t}^i = \gamma_{t-1,t-1}^i P^{(i)}$. Lorsqu'on mesure ($M^i(t) = 1$), la valeur de $L^i(t)$ apporte de l'information sur l'état caché $S^i(t)$. Le filtre est alors mis à jour de la façon suivante :

$$\begin{aligned} \gamma_{t,t}^i(j) &= \mathbb{P}(S^i(t) = j \mid I_{t-1}^i, L^i(t)) \\ &\propto p(S^i(t) = j, I_{t-1}^i, L^i(t)) \\ &\propto p(L^i(t) \mid S^i(t) = j, I_{t-1}^i) \mathbb{P}(S^i(t) = j \mid I_{t-1}^i) \quad (4.6) \\ &\propto p(L^i(t) \mid S^i(t) = j) \gamma_{t-1,t-1}^i(j) \\ &\propto p(L^i(t) \mid S^i(t) = j) \left(\gamma_{t-1,t-1}^i P^{(i)} \right)_j \end{aligned}$$

En résumé, nous avons

$$\gamma_{t,t}^{(i)} \propto \begin{cases} \gamma_{t-1,t-1}^i P^{(i)} & \text{si } M^i(t) = 0 \\ \gamma_{t-1,t-1}^i B^i(t) & \text{si } M^i(t) = 1 \end{cases} \quad (4.7)$$

où $B^i(t)$ est une matrice diagonale $K \times K$ avec pour éléments diagonaux les valeurs de la loi conditionnelle $p(L^i(t) \mid S^i(t) = j)$. Le coefficient de proportionnalité peut être calculé facilement à l'aide de la condition de normalisation $\sum_j \gamma_{t,t}^{(i)}(j) = 1$.

4.4 OPTIMISATION 1-PAS EN AVANT

Dans cette section nous déterminons la politique de mesure myope dans le cas de deux nœuds pouvant communiquer par deux chemins (figure 4.2). On parle de politique myope, ou « 1-pas en avant », car l'on ne considère pas, pour l'instant, l'impact futur des décisions de mesure à un instant.

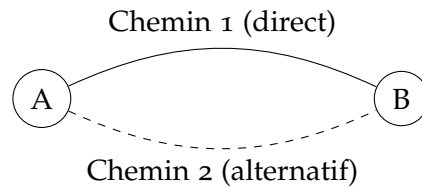


FIGURE 4.2 – Configuration à deux nœuds.

Afin d'introduire progressivement le problème, nous dérivons dans cette section les solutions analytiques dans le cas de 1 et 2 chemins stochastiques. Une solution générale pour un nombre illimité de chemins, et prenant en compte l'impact futur des décisions de mesure, sera présentée dans la section 4.5.

4.4.1 Solution pour 1 chemin stochastique

Commençons par le cas où le délai d'un chemin évolue selon une chaîne de Markov, ou un modèle de Markov caché, à temps discret à K états, et où le délai de l'autre chemin est constant (équivalent à une chaîne de Markov à un seul état). Afin de simplifier les notations, nous omettons l'indice i qui représente ici toujours le chemin stochastique. Le délai du chemin déterministe est noté l . La décision de routage est donnée par :

$$C(t) = \begin{cases} 1 & \text{si } \mathbb{E}(L(t) \mid I_t) < l \\ 0 & \text{sinon} \end{cases}. \quad (4.8)$$

Lorsque $C(t) = 1$ on emprunte le chemin stochastique, et lorsque $C(t) = 0$ on emprunte le chemin déterministe.

4.4.1.1 Récompense

Afin de déterminer l'action de mesure optimale $M^*(t)$ à chaque instant, nous devons définir une fonction objectif qui effectue un compromis entre le nombre de mesures et le choix optimal du chemin. Pour cela, on associe à chaque mesure un coût c qui indique que les mesures fréquentes ne sont pas encouragées, car, par exemple, elles peuvent saturer le réseau. On définit également le gain de délai $G(t)$ comme la différence de délai entre le chemin déterministe et le chemin sélectionné :

$$G(t) = \mathbb{1}_{C(t)=1}(l - L(t)). \quad (4.9)$$

Ainsi si on emprunte le chemin stochastique et que celui-ci est effectivement plus court que le chemin déterministe, alors le gain de délai $G(t)$ sera positif. Finalement, nous définissons notre fonction objectif à maximiser comme le gain de délai pénalisé par le coût de mesure :

$$\tilde{G}(t) = \mathbb{1}_{C(t)=1}(l - L(t)) - \mathbb{1}_{M(t)=1}c. \quad (4.10)$$

Remarquons que $\tilde{G}(t)$ dépend de $L(t)$ qui est aléatoire.

Afin de déterminer l'action de mesure optimale, nous devons calculer le gain pénalisé espéré $\mathbb{E}(\tilde{G}(t) \mid I_{t-1}, M(t))$ en fonction de la décision de mesure $M(t)$. Le conditionnement par I_{t-1} et $M(t)$ indique que nous supposons la décision de mesure connue, mais que le résultat de mesure n'est pas pris en compte dans la décision de mesure, puisqu'il n'est pas disponible au moment où cette dernière est prise.

CAS MESURE Dans le cas où l'on mesure le chemin stochastique, alors le délai $L(t)$ est connu au moment où on prend la décision de routage $C(t)$. Le chemin stochastique est choisi si $L(t) < l$. On a alors :

$$\begin{aligned} \mathbb{E}(\tilde{G}(t) | I_{t-1}, M(t) = 1) &= \mathbb{E}(\mathbb{1}_{L(t) < l} (l - L(t)) | I_{t-1}) - c \\ &= \mathbb{E}(\mathbb{1}_{L(t) < l} | I_{t-1})l - \mathbb{E}(\mathbb{1}_{L(t) < l} L(t) | I_{t-1}) - c \\ &= \mathbb{P}(L(t) < l | I_{t-1})l - \mathbb{E}(\mathbb{1}_{L(t) < l} L(t) | I_{t-1}) - c. \end{aligned} \quad (4.11)$$

Si le délai est modélisé par une chaîne de Markov, on a :

$$\begin{aligned} \mathbb{P}(L(t) < l | I_{t-1}) &= \sum_{j: l_j < l} \gamma_{t-1,t}(j) \\ \mathbb{E}(\mathbb{1}_{L(t) < l} L(t) | I_{t-1}) &= \sum_{j: l_j < l} \gamma_{t-1,t}(j) l_j. \end{aligned} \quad (4.12)$$

Dans le cas d'un modèle de Markov caché on a :

$$\begin{aligned} \mathbb{P}(L(t) < l | I_{t-1}) &= \sum_j \gamma_{t-1,t}(j) \int_0^l p_j(x) dx \\ \mathbb{E}(\mathbb{1}_{L(t) < l} L(t) | I_{t-1}) &= \sum_j \gamma_{t-1,t}(j) \int_0^l x p_j(x) dx \end{aligned} \quad (4.13)$$

où $p_j(x) = p(L(t) = x | S(t) = j)$.

CAS PRÉDICTION Dans le cas où l'on ne mesure pas le chemin stochastique, alors la décision de routage est prise sur la base de l'espérance du délai du chemin stochastique. Autrement dit, on choisit le chemin stochastique si $\bar{L}(t) < l$ où

$$\bar{L}(t) = \mathbb{E}(L(t) | I_{t-1}) = \sum_j \gamma_{t-1,t}(j) l_j. \quad (4.14)$$

Pour rappel, si le délai est modélisé par un modèle de Markov caché, $l_j = \mathbb{E}(L(t) | S(t) = j)$. Remarquons que dans le cas où l'on ne mesure pas, la décision de routage est déterministe par rapport à I_{t-1} . On a alors :

$$\mathbb{E}(\tilde{G} | I_{t-1}, M(t) = 0) = \mathbb{1}_{\bar{L}(t) < l} (l - \bar{L}(t)). \quad (4.15)$$

4.4.1.2 Décision de mesure

Les nœuds souhaitent choisir à chaque instant la décision de mesure optimale $M^*(t)$, c'est à dire celle qui maximise la récompense :

$$M^*(t) = \operatorname{argmax}_{M \in \{0,1\}} \mathbb{E}(\tilde{G}(t) | I_{t-1}, M). \quad (4.16)$$

Cette décision dépend uniquement du prédicteur $\gamma_{t-1,t}$. Afin de simplifier les calculs, nous considérons ici que le chemin stochastique

est modélisé par chaîne de Markov à 2 états : $\{1, 2\}$. Nous pouvons ainsi exprimer la récompense en fonction de $x = \gamma_{t-1,t}(1)$ puisque $\gamma_{t-1,t}(1) + \gamma_{t-1,t}(2) = 1$. Nous supposons également que $l_1 < l < l_2$. En effet, si $l < \min(l_1, l_2)$ alors la décision de routage serait de toujours choisir le chemin déterministe. À l'inverse, si $l > \max(l_1, l_2)$, alors la décision serait de toujours choisir le chemin stochastique. Il n'y aurait donc jamais besoin de mesurer le chemin stochastique.

Afin de déterminer la décision de mesure qui maximise le gain pénalisé, nous réécrivons les récompenses en fonction de $x = \gamma_{t-1,t}(1) = 1 - \gamma_{t-1,t}(2)$. Dans le cas où $M(t) = 1$ nous avons

$$\begin{aligned} \mathbb{P}(L(t) < l \mid I_{t-1}) &= \sum_{j:l_j < l} \gamma_{t-1,t}(j) = x \\ \mathbb{E}(\mathbb{1}_{L(t) < l} L(t) \mid I_{t-1}) &= \sum_{j:l_j < l} \gamma_{t-1,t}(j) l_j = x l_1 \end{aligned} \quad (4.17)$$

et

$$\begin{aligned} \mathbb{E}(\tilde{G}(t) \mid I_{t-1}, M(t) = 1) &= \mathbb{P}(L(t) < l \mid I_{t-1})l - \mathbb{E}(\mathbb{1}_{L(t) < l} L(t) \mid I_{t-1}) - c \\ &= x l - x l_1 - c = (l - l_1)x - c \end{aligned} \quad (4.18)$$

d'après les équations 4.11 et 4.12. Dans le cas où $M(t) = 0$ nous avons

$$\bar{L}(t) = l_1 x + l_2(1 - x) = (l_1 - l_2)x + l_2 \quad (4.19)$$

et

$$\begin{aligned} \mathbb{E}(\tilde{G} \mid I_{t-1}, M(t) = 0) &= \mathbb{1}_{\bar{L}(t) < l} (l - \bar{L}(t)) \\ &= \mathbb{1}_{(l_1 - l_2)x + l_2 < l} (l - ((l_1 - l_2)x + l_2)) \\ &= \mathbb{1}_{x > \frac{l - l_2}{l_1 - l_2}} ((l - l_2) - (l_1 - l_2)x) \end{aligned} \quad (4.20)$$

d'après les équations 4.14 et 4.15.

La différence entre les deux récompenses est donnée par :

$$\begin{aligned} \Delta &= \mathbb{E}(\tilde{G} \mid I_{t-1}, M(t) = 1) - \mathbb{E}(\tilde{G} \mid I_{t-1}, M(t) = 0) \\ &= ((l - l_1)x - c) - ((l_2 - l_1)x - (l_2 - l)) \mathbb{1}_{x > \frac{l_2 - l}{l_2 - l_1}}. \end{aligned} \quad (4.21)$$

Quand $x > \frac{l_2 - l}{l_2 - l_1}$ on a $\Delta = (l_2 - l)(1 - x) - c$ et $\Delta > 0$ si $x < 1 - \frac{c}{l_2 - l}$. Quand $x < \frac{l_2 - l}{l_2 - l_1}$ on a $\Delta = (l - l_1)x - c$ et $\Delta > 0$ si $x > \frac{c}{l - l_1}$. On obtient donc les décisions de mesure optimales suivantes :

$$\begin{aligned} x > \max\left(\frac{l_2 - l}{l_2 - l_1}, 1 - \frac{c}{l_2 - l}\right) &\Rightarrow M^*(t) = 0 \\ \frac{l_2 - l}{l_2 - l_1} < x < 1 - \frac{c}{l_2 - l} &\Rightarrow M^*(t) = 1 \\ \frac{c}{l - l_1} < x < \frac{l_2 - l}{l_2 - l_1} &\Rightarrow M^*(t) = 1 \\ x < \min\left(\frac{l_2 - l}{l_2 - l_1}, \frac{c}{l - l_1}\right) &\Rightarrow M^*(t) = 0 \end{aligned} \quad (4.22)$$

Les intervalles de mesure au milieu de l'équation 4.22 correspondant à la décision de mesure $M^*(t) = 1$ existent si $\frac{l_2-l}{l_2-l_1} < 1 - \frac{c}{l_2-l}$ et si $\frac{c}{l-l_1} < \frac{l_2-l}{l_2-l_1}$.

Remarquons que ces deux inégalités sont équivalentes à $c < \frac{(l-l_1)(l_2-l)}{(l_2-l_1)}$. Autrement dit, si le coût de mesure c est supérieur ou égal à $\frac{(l-l_1)(l_2-l)}{(l_2-l_1)}$, la décision optimale est de ne jamais mesurer. En effet, le gain lié à l'optimisation du routage serait plus faible que le coût de mesure.

Si le coût de mesure est raisonnable, il peut être judicieux de mesurer le chemin stochastique. C'est le cas quand $\frac{c}{l-l_1} < x < \frac{l_2-l}{l_2-l_1}$ et $\frac{l_2-l}{l_2-l_1} < x < 1 - \frac{c}{l_2-l}$. C'est-à-dire quand l'incertitude sur l'état du chemin stochastique est grande. Au contraire, si x est proche de 0 ou 1, plus précisément si $x < \frac{c}{l-l_1}$ ou si $x > 1 - \frac{c}{l_2-l}$, alors il n'est pas nécessaire de mesurer.

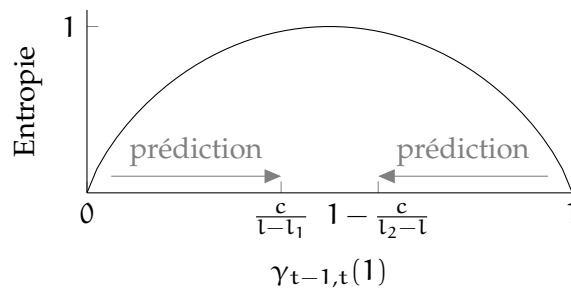


FIGURE 4.3 – Politique à seuil.

Remarquons que la politique obtenue est une politique dite à seuil (figure 4.3). Lorsque l'on fait une prédiction, on perd en connaissance sur l'état du chemin. L'entropie du prédicteur augmente. Les mesures augmentent la connaissance sur l'état du chemin, et réduisent l'entropie du prédicteur. Ces politiques sont intéressantes, car elles ne contiennent que peu de paramètres (les seuils de décisions) et sont ainsi très faciles à implémenter.

4.4.2 Solution pour 2 chemins stochastiques

Considérons maintenant le cas de deux chemins stochastiques. La décision de routage est donnée par :

$$C(t) = \begin{cases} 1 & \text{si } \mathbb{E}(L^1(t) | I_t^1) < \mathbb{E}(L^2(t) | I_t^2) \\ 2 & \text{sinon} \end{cases} \quad (4.23)$$

Autrement dit, on choisit le chemin dont l'espérance du délai conditionnellement aux mesures passées est la plus faible. On note également $M^i(t) \in \{0, 1\}$ la décision de mesure, et c^i le coût de mesure sur le chemin i .

4.4.2.1 Récompense

Notre fonction objectif à maximiser, le gain pénalisé, est défini comme l'opposé du délai du chemin sélectionné moins le coût de mesure pour les chemins mesurés :

$$\tilde{G}(t) = -\mathbb{1}_{C(t)=1}L^1(t) - \mathbb{1}_{C(t)=2}L^2(t) - \mathbb{1}_{M^1(t)=1}c^1 - \mathbb{1}_{M^2(t)=1}c^2. \quad (4.24)$$

Ainsi on pénalise l'action de mesurer et le choix d'un chemin de délai important.

Tout comme dans le cas d'un seul chemin stochastique, le gain pénalisé dépend de $L^i(t)$ qui est aléatoire. Afin de déterminer la décision de mesure optimale, nous devons donc calculer l'espérance du gain pénalisé conditionnellement à la décision de mesure, ici noté R comme « Récompense » :

$$\begin{aligned} R &= \mathbb{E}(\tilde{G}(t) \mid I_{t-1}, M^1(t), M^2(t)) \\ &= -\mathbb{E}(\mathbb{1}_{C(t)=1}L^1(t) \mid I_{t-1}, M^1(t), M^2(t)) \\ &\quad - \mathbb{E}(\mathbb{1}_{C(t)=2}L^2(t) \mid I_{t-1}, M^1(t), M^2(t)) \\ &\quad - \mathbb{1}_{M^1(t)=1}c^1 - \mathbb{1}_{M^2(t)=1}c^2. \end{aligned} \quad (4.25)$$

CAS $R = (0,0)$ Dans le cas où l'on ne mesure aucun chemin, on choisit le chemin dont le délai espéré, conditionnellement aux mesures passées, est le plus faible :

$$C(t) = \operatorname{argmin}_{i \in \{1,2\}} \bar{L}^i(t) \quad (4.26)$$

avec

$$\bar{L}^i(t) = \mathbb{E}(L^i(t) \mid I_{t-1}^i) = \sum_j \gamma_{t-1,t}^{(i)}(j) l_j^i. \quad (4.27)$$

La récompense est alors égale à l'opposé du délai espéré du chemin choisi :

$$R = -\mathbb{1}_{C(t)=1}\bar{L}^1(t) - \mathbb{1}_{C(t)=2}\bar{L}^2(t). \quad (4.28)$$

CAS $R = (1,1)$ Dans le cas où l'on mesure les deux chemins, on choisit le chemin dont le délai mesuré est le plus faible :

$$C(t) = \operatorname{argmin}_{i \in \{1,2\}} L^i(t). \quad (4.29)$$

La récompense est alors égale à :

$$\begin{aligned} R &= -\mathbb{E}(\mathbb{1}_{L^2(t) \leq L^1(t)}L^2(t) \mid I_{t-1}) \\ &\quad - \mathbb{E}(\mathbb{1}_{L^2(t) > L^1(t)}L^1(t) \mid I_{t-1}) \\ &\quad - c_1 - c_2 \\ &= -\sum_i \sum_j \gamma_{t-1,t}^{(1)}(i)\gamma_{t-1,t}^{(2)}(j) \min(l_i^1, l_j^2) - c^1 - c^2. \end{aligned} \quad (4.30)$$

CAS R = (0,1) Dans le cas où l'on ne mesure qu'un seul chemin, ici le deuxième chemin, on choisit le chemin mesuré si son délai actuel est inférieur (ou égal) au délai espéré de l'autre chemin :

$$C(t) = \begin{cases} 2 & \text{si } L^2(t) \leq \bar{L}^1(t) \\ 1 & \text{sinon} \end{cases}. \quad (4.31)$$

La récompense est alors égale à :

$$\begin{aligned} R &= -\mathbb{E}(L^2(t)\mathbb{1}_{L^2(t) \leq \bar{L}^1(t)} | I_{t-1}) - \mathbb{E}(L^1(t)\mathbb{1}_{L^2(t) > \bar{L}^1(t)} | I_{t-1}) - c_2 \\ &= -\sum_{j:l_j^2 \leq \bar{L}^1(t)} \gamma_{t-1,t}^{(2)}(j)l_j^2 - \bar{L}^1(t)\mathbb{P}(L^2(t) > \bar{L}^1(t)) - c_2 \\ &= -\sum_{j:l_j^2 \leq \bar{L}^1(t)} \gamma_{t-1,t}^{(2)}(j)l_j^2 - \bar{L}^1(t) \sum_{j:l_j^2 > \bar{L}^1(t)} \gamma_{t-1,t}^{(2)}(j) - c_2. \end{aligned}$$

Le cas R = (1,0) est similaire au cas présent, à une permutation des indices près.

Les résultats sont résumés dans le [tableau 4.1](#). Les résultats dans le cas où le délai est modélisé par un modèle de Markov caché sont donnés dans [106, annexe C].

M(t)	C(t)	R
(0,0)	$\mathbb{1}_{\bar{L}^2(t) \leq \bar{L}^1(t)}$	$-\mathbb{1}_{C(t)=2}\bar{L}^2(t) - \mathbb{1}_{C(t)=1}\bar{L}^1(t)$
(0,1)	$\mathbb{1}_{L^2(t) \leq \bar{L}^1(t)}$	$-\sum_{j:l_j^2 \leq \bar{L}^1(t)} \gamma_{t-1,t}^{(2)}(j)l_j^2 - \bar{L}^1(t) \sum_{j:l_j^2 > \bar{L}^1(t)} \gamma_{t-1,t}^{(2)}(j) - c^2$
(1,0)	$\mathbb{1}_{\bar{L}^2(t) \leq L^1(t)}$	$-\sum_{i:l_i^1 < \bar{L}^2(t)} \gamma_{t-1,t}^{(1)}(i)l_i^1 - \bar{L}^2(t) \sum_{i:l_i^1 \geq \bar{L}^2(t)} \gamma_{t-1,t}^{(1)}(i) - c^1$
(1,1)	$\mathbb{1}_{L^2(t) \leq L^1(t)}$	$-\sum_{i,j} \gamma_{t-1,t}^{(1)}(i)\gamma_{t-1,t}^{(2)}(j) \min(l_i^1, l_j^2) - c^1 - c^2$

TABLE 4.1 – Décisions de routage et récompenses 1-pas en avant pour deux chemins markoviens, en fonction de la décision de mesure.

4.4.2.2 Décision de mesure

Les nœuds souhaitent choisir à chaque instant la décision de mesure optimale $(M^0, M^1)^*$ qui maximise la récompense :

$$(M^0, M^1)^* = \operatorname{argmax}_{M^0, M^1} \mathbb{E}(\tilde{G}(t) | I_{t-1}, M^0, M^1). \quad (4.32)$$

Comme nous le verrons dans la [section 5.1](#), une politique à seuil existe également dans le cas de deux chemins ou plus, mais nous n'avons

pas réussi à obtenir une forme close pour les frontières de décision. La solution est donc numérique et peut être implémentée en calculant la récompense pour les quatre valeurs de $(M^0, M^1)^*$ possibles, et en choisissant la valeur qui maximise la récompense. Nous avons vérifié que les résultats obtenus sont cohérents avec le cas de 1 chemin stochastique et 1 chemin déterministe. En effet si l'on fixe la valeur de l'un des prédicteurs à 0 ou 1, cela revient à supposer que le chemin possède un délai constant (l_0^i ou l_1^i). Dans ce cas, la décision de mesure pour l'autre chemin est identique à la politique à seuil (équation 4.22).

4.5 OPTIMISATION À HORIZON DE TEMPS INFINI

Jusqu'à présent, nous avons considéré une politique « myope » qui ne prend pas en compte l'impact de la décision de mesure courante sur les décisions de mesure et de routage futures. Cependant, si l'on mesure le chemin au temps t , alors la connaissance accumulée nous permettra d'améliorer notre croyance sur l'état du chemin dans le futur, permettant ainsi de meilleures décisions de routage, et diminuant le besoin de mesurer à nouveau ce chemin dans un futur proche. Dans cette section, nous généralisons la politique myope présentée précédemment. Nous proposons une politique qui prend en compte les récompenses futures dans le critère d'optimisation, et qui peut être apprise pour un nombre illimité de chemins. Pour cela, nous formulons le problème dans le cadre des processus de décision markoviens.

De façon simplifiée, un processus de décision markovien (MDP ou *Markov Decision Process*) peut être vu comme une chaîne de Markov contrôlée. Les probabilités de transition à l'instant t dépendent non seulement de l'état courant s_t , mais également d'une action a_t . On associe une récompense $R(a, s_t, s_{t+1})$ à la transition de l'état s_t à s_{t+1} suite à l'action a . On cherche alors la correspondance action-état (appelée politique) optimale qui maximise une récompense cumulée sur tous les pas de temps. Les processus de décision markoviens ont été introduits par Richard Bellman en 1957 [15]. Leur résolution repose sur la méthode de programmation dynamique, également développée par Bellman. Plus récemment, la théorie de l'apprentissage par renforcement [101] se consacre à l'apprentissage en ligne des processus de décision markoviens.

Voir [33] pour une anecdote sur le choix du nom « programmation dynamique ».

Comme nous le verrons dans la section suivante, nous pouvons formuler le problème dans le cadre des MDPs car nous disposons d'un modèle markovien, le HDP-HMM, pour modéliser la dynamique du délai sur les chemins.

4.5.1 Formulation MDP

On considère \mathcal{P} chemins différents entre une même origine et destination. Chaque chemin est indicé par $i \in \{1, \dots, \mathcal{P}\}$ et le délai $L^i(t)$

de chaque chemin est modélisé par une chaîne de Markov à K^i états. On note $M^i \in \{0, 1\}$ la décision de mesure (1 si on mesure, 0 sinon), et $c^i \in \mathbb{R}^+$ le coût de mesure sur le chemin i .

Toute l'information I_t^i fournie par les mesures passées jusqu'au temps t , et utile pour prédire les valeurs futures $L^i(t+k), k \geq 0$ est résumée dans le filtre $\gamma_{t,t}^{(i)}$:

$$\gamma_{t,t}^{(i)}(j) = \mathbb{P}(L^i(t) = l_j^i \mid I_t^i). \quad (4.33)$$

Remarquons que ce filtre prend des valeurs continues, comprises entre 0 et 1. Hors, la résolution des processus de décision markoviens à espace d'état continu est compliquée, puisqu'il n'est pas possible d'énumérer tous les états du système (ici, toutes les valeurs possibles du filtre), dans le but de rechercher une politique optimale.

Cependant, on peut également remarquer que l'information I_t^i peut être résumée pour chaque chemin i par la paire $s^i = (\tau^i, L_{\text{last}}^i)$ où τ^i est le nombre de pas de temps depuis la dernière mesure, et L_{last}^i est la dernière valeur mesurée. Par exemple, si nous sommes au temps t et que nous venons juste de mesurer la valeur de $L^i(t)$, alors $\tau^i = 0$ et L_{last}^i est la valeur mesurée de $L^i(t)$. Au temps t , une valeur $\tau^i \geq 1$ signifie que L^i a été mesuré pour la dernière fois à $t - \tau^i$ et que la valeur correspondante était L_{last}^i . Nous pouvons réécrire le filtre en fonction de ces notations :

$$\gamma_{t,t}^{(i)} = e_{L_{\text{last}}^i} \left(P^{(i)} \right)^{\tau^i} \quad (4.34)$$

où e_i est le vecteur ligne avec un 1 à la position i , et $P^{(i)}$ est la matrice de transition du chemin i .

Nous pouvons ainsi définir l'état de la chaîne de Markov contrôlée comme :

$$\mathbf{s} = \{s^i\}_i = \{(\tau^i, L_{\text{last}}^i)\}_i, \quad i = 1, \dots, \mathcal{P}. \quad (4.35)$$

Avec cette définition l'espace d'états est discret. Remarquons que nous pouvons rendre l'espace d'état fini en définissant une valeur maximum τ_{max}^i . En effet, après un certain temps sans mesure, le filtre atteint la distribution stationnaire de la chaîne de Markov du chemin. Sa valeur ne change plus. En pratique τ_{max} sera réglé suffisamment proche de la valeur pour laquelle le filtre atteint la distribution stationnaire.

On note $\mathbf{M} = \{M^i\}_i$ le vecteur d'action. Avec ces définitions, \mathbf{s} est une chaîne de Markov à temps discret contrôlée par la décision de mesure \mathbf{M} .

Les probabilités de transition de la chaîne de Markov contrôlée par \mathbf{M} sont calculées comme suit :

$$\mathbb{P}(\mathbf{s}' \mid \mathbf{s}, \mathbf{M}) = \prod_i \mathbb{P}(s'_i \mid s_i, M^i). \quad (4.36)$$

Lorsqu'on ne mesure pas un chemin ($M^i = 0$), on incrémente seulement le nombre de pas de temps depuis la dernière mesure, le

dernier état mesuré ne changeant pas. Les probabilités de transition sont donc de la forme

$$\mathbb{P}[(\tau^i + 1, L_{\text{last}}^i) \mid (\tau^i, L_{\text{last}}^i), M^i = 0] = 1. \quad (4.37)$$

Au contraire, si on mesure un chemin ($M^i = 1$), on réinitialise le nombre de pas de temps depuis la dernière mesure à 0, et la dernière valeur mesurée du chemin est aléatoire. Les probabilités de transition sont alors de la forme

$$\mathbb{P}[(0, l_j^i) \mid (\tau^i, L_{\text{last}}^i), M^i = 1] = e_{L_{\text{last}}^i} \left(P^{(i)} \right)^{\tau^i+1} e_j^T \quad (4.38)$$

pour $j = 1, \dots, K^i$.

Afin de définir le problème MDP, nous devons définir la récompense instantanée, c'est à dire la récompense après avoir appliqué \mathbf{M} dans l'état \mathbf{s} et être arrivé dans l'état \mathbf{s}' . Nous définissons d'abord le délai minimum espéré sur les \mathcal{P} chemins :

$$D(\mathbf{s}') = \min_i \mathbb{E}(L^i(t) \mid I_t^i) = \min \left(\underbrace{\min_{i:\tau^i \geq 1} \mathbb{E}(L^i \mid \tau^i, L_{\text{last}}^i)}_{\text{chemins non mesurés}}, \underbrace{\min_{i:\tau^i=0} L_{\text{last}}^i}_{\text{chemins mesurés}} \right). \quad (4.39)$$

avec $\mathbb{E}(L^i \mid \tau^i, L_{\text{last}}^i) = \sum_j l_j^i \gamma_{\tau^i, t}^{(i)}(j)$.

Nous pouvons alors définir la récompense instantanée pour être arrivé dans l'état \mathbf{s}' après avoir effectué l'action \mathbf{M} dans l'état \mathbf{s} comme :

$$R(\mathbf{M}, \mathbf{s}, \mathbf{s}') = -D(\mathbf{s}') - \sum_i c^i \mathbb{1}_{M^i=1}. \quad (4.40)$$

Remarquons que cette récompense ne dépend pas de l'état initial \mathbf{s} , et que l'action \mathbf{M} est comprise dans l'état final \mathbf{s}' puisque $\tau^i = 0$ si $M^i = 1$. On peut alors réécrire la récompense instantanée ainsi :

$$R(\mathbf{s}') = -D(\mathbf{s}') - \sum_i c^i \mathbb{1}_{\tau^i=0}. \quad (4.41)$$

La récompense espérée de l'action \mathbf{M} dans l'état \mathbf{s} est donnée par :

$$\bar{R}(\mathbf{s}, \mathbf{M}) = \sum_{\mathbf{s}'} \mathbb{P}(\mathbf{s}' \mid \mathbf{s}, \mathbf{M}) R(\mathbf{s}'). \quad (4.42)$$

On note \mathcal{M} l'ensemble des actions possibles et \mathcal{S} l'ensemble des états. On note également Π l'ensemble des politiques π , où $\pi = \{\pi_0, \pi_1, \pi_2, \dots\}$ et $\pi_t : \mathcal{S} \rightarrow \mathcal{M}$. Pour une politique donnée, $\pi_t(\mathbf{s})$ représente la décision de mesure $M(t)$ qui est prise au temps t si le système est dans l'état \mathbf{s} .

Finalement, nous introduisons la récompense cumulée amortie à horizon infini J , définie comme :

$$J_\pi(\mathbf{s}_0) = \mathbb{E}_\pi \left(\sum_{t=1}^{\infty} \rho^t R(\mathbf{s}_t) \mid \mathbf{s}_0 \right) \quad (4.43)$$

où $0 < \rho < 1$ est un facteur d'amortissement qui donne plus d'importance aux récompenses dans un futur proche, et garantit que la somme soit finie.

4.5.2 Solution pour n chemins stochastiques

Le problème consiste désormais à maximiser J_π sur Π . Les processus de décision markoviens ont une politique optimale qui est stationnaire dans le temps, c'est-à-dire que $\pi_0 = \pi_1 = \dots = \pi_T$ ([61, th. 6.4.1]). À partir de maintenant $\pi_t(\mathbf{s})$ ne dépend donc plus que de \mathbf{s} .

La fonction de valeur optimale et la politique optimale sont définies comme :

$$V(\mathbf{s}) = \max_{\mu} J_\pi(\mathbf{s}) \quad \pi^* = \operatorname{argmax}_{\pi} J_\pi(\mathbf{s}). \quad (4.44)$$

La fonction de valeur optimale est l'unique solution de l'équation de Bellman ([61, th. 6.4.1]). Dans notre cas celle-ci est donnée par l'équation suivante :

$$V(\mathbf{s}) = \max_{\mathbf{M} \in \mathcal{M}} \sum_{\mathbf{s}'} \mathbb{P}(\mathbf{s}' | \mathbf{s}, \mathbf{M}) (R(\mathbf{s}') + \rho V(\mathbf{s}')). \quad (4.45)$$

La fonction V peut-être approximée aussi précisément que souhaitée grâce à l'algorithme d'itération sur la valeur ([61, sec. 6.4.2]). L'algorithme d'itération sur la valeur est un algorithme d'approximation itératif :

$$V_{n+1}(\mathbf{s}) = \max_{\mathbf{M} \in \mathcal{M}} \sum_{\mathbf{s}'} \mathbb{P}(\mathbf{s}' | \mathbf{s}, \mathbf{M}) (R(\mathbf{s}') + \rho V_n(\mathbf{s}')) \quad (4.46)$$

avec $V_0(\mathbf{s}) = 0$. La politique optimale $\pi^*(\mathbf{s})$ est alors obtenue à la convergence comme :

$$\pi^*(\mathbf{s}) = \operatorname{argmax}_{\mathbf{M} \in \mathcal{M}} \sum_{\mathbf{s}'} \mathbb{P}(\mathbf{s}' | \mathbf{s}, \mathbf{M}) (R(\mathbf{s}') + \rho V(\mathbf{s}')). \quad (4.47)$$

Une condition de terminaison classique est $\|V_{n+1} - V_n\|_\infty < \epsilon$, où $\|\mathbf{x}\|_\infty = \max(|x_1|, \dots, |x_n|)$. Il est également possible d'obtenir la fonction de valeur associée à une politique grâce à l'algorithme d'évaluation de la politique :

$$V_{n+1}^\pi(\mathbf{s}) = \sum_{\mathbf{s}'} \mathbb{P}(\mathbf{s}' | \mathbf{s}, \pi(\mathbf{s})) (R(\mathbf{s}') + \rho V_n^\pi(\mathbf{s}')) \quad (4.48)$$

avec $V_0^\pi(\mathbf{s}) = 0$.

La complexité de l'algorithme d'itération sur la valeur dépend du nombre d'états, qui est $\prod_i K^i \tau_{\max}^i$ et ainsi grandit vite avec le nombre de chemins et de valeurs possibles pour le délai. Il existe d'autres algorithmes de résolution des MDP, tels que l'algorithme d'itération sur la politique, mais leur complexité reste similaire. Nous verrons dans le [chapitre 5](#) plusieurs façons de réduire la complexité du problème.

Une fois la politique obtenue, elle peut-être déployée dans le contrôleur MDP ([figure 4.4](#)).

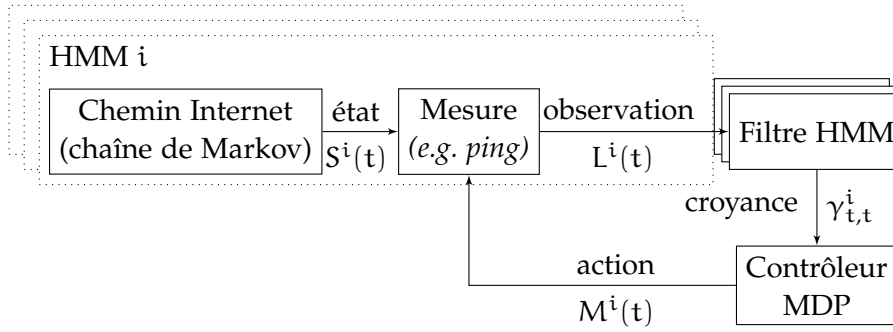


FIGURE 4.4 – Schéma de principe de la métrologie parcimonieuse.

4.6 RÉSULTATS

Nous présentons d’abord des résultats sur des modèles synthétiques, puis nous simulons un overlay de routage avec des données de délai RIPE Atlas.

4.6.1 1 chemin stochastique

Considérons le cas d’un chemin stochastique et d’un chemin déterministe (figure 4.2). Le délai du chemin déterministe est $l = 8$. Le délai du chemin stochastique est modélisé par une chaîne de Markov à 2 états de délais $l_1 = 5, l_2 = 10$ et de matrice de transition $P = \begin{bmatrix} 0.99 & 0.01 \\ 0.02 & 0.98 \end{bmatrix}$. Le coût de mesure est de $c = 0.65$, ce qui est inférieur au seuil $(l_2 - l)(l - l_1)/(l_2 - l_1)$ (section 4.4.1.2). La distribution stationnaire de la chaîne de Markov est $(\pi_1 = 2/3, \pi_2 = 1/3)$, ce qui correspond à un délai moyen de 6.7.

Nous appliquons la politique 1 pas en avant de la section 4.4.1. La décision optimale est de mesurer le chemin stochastique lorsque le prédicteur $\gamma_{t-1,t}(1)$ est dans l’intervalle $[x_{\min}, x_{\max}]$ avec $x_{\min} = c/(l - l_1) = 0.22$ et $x_{\max} = 1 - c/(l_2 - l) = 0.675$. En dehors de cet intervalle, la décision optimale est de ne pas mesurer.

Une trajectoire de la chaîne de Markov représentant le chemin stochastique est simulée pour 3000 instants. La politique 1 pas en avant est ensuite appliquée instant par instant. La figure 4.5 montre les résultats. La première série représente la trajectoire générée ainsi que les instants de mesure. La seconde série représente la valeur du prédicteur 1-pas en avant $x = \gamma_{t-1,t}(1)$. À chaque fois qu’une mesure est réalisée, le filtre est mis à jour à $\gamma_{t,t}(1) = 1$ ou 0, en fonction du résultat de la mesure ($L(t) = l_1$ ou $L(t) = l_2$). Le prédicteur 1-pas en avant $\gamma_{t,t+1}$ est donc donné par $e_i P$ avec $i = 1$ ou 2. Comme on peut le voir sur la figure, entre deux mesures, le prédicteur converge géométriquement vers la distribution stationnaire de la chaîne de Markov (π_1, π_2) . Dès qu’un seuil x_{\min} ou x_{\max} est atteint, une nouvelle mesure est réalisée.

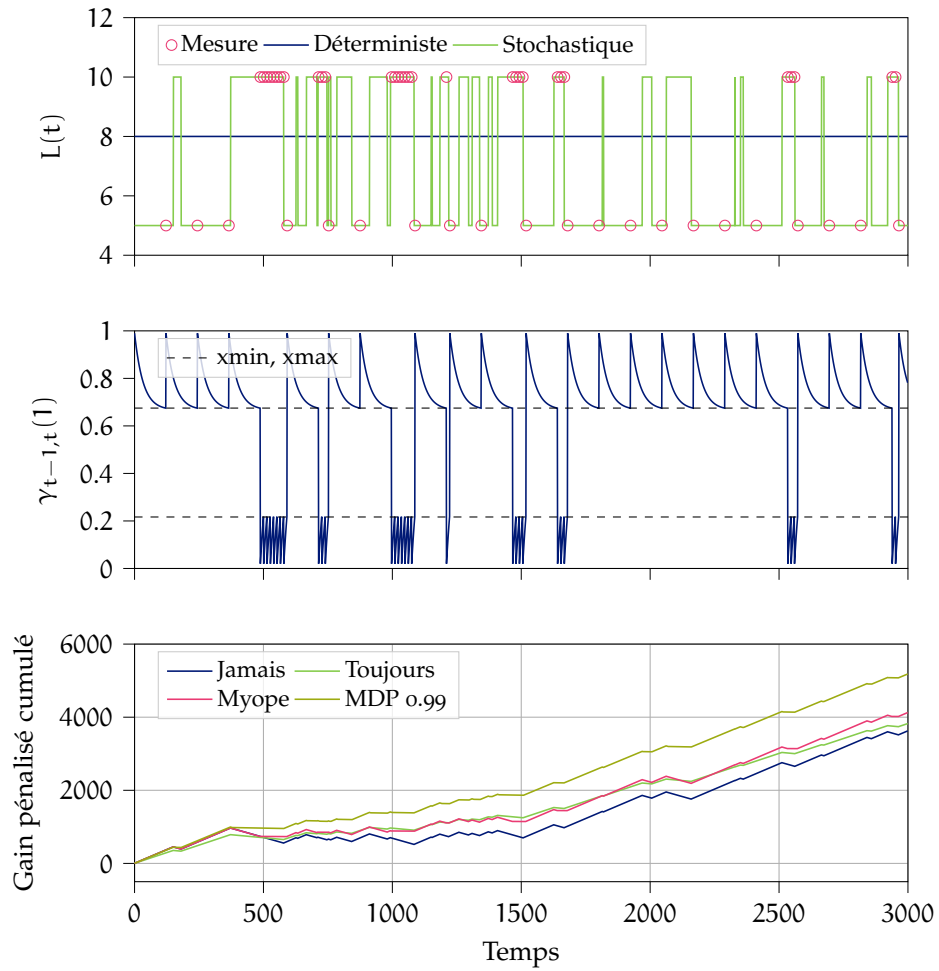


FIGURE 4.5 – Simulation de quatre politiques : ne jamais mesurer, toujours mesurer, myope et MDP ($\rho = 0.99$). (Haut) Trajectoire simulée de la chaîne de Markov et instants de mesure avec la politique myope. (Milieu) Valeurs du prédicteur pour la politique myope. (Bas) Gain pénalisé cumulé au cours du temps.

Le gain pénalisé cumulé $\sum_{t=1}^T \tilde{G}(t)$, où $\tilde{G}(t) = \mathbb{1}_{C(t)=1}(l - L(t)) - \mathbb{1}_{M(t)=1}c$, est représenté dans la troisième série. La politique à seuil est comparée à deux politiques constantes : ne jamais mesurer ($M(t) = 0$ pour tout t) et toujours mesurer ($M(t) = 1$ pour tout t). Le gain pénalisé cumulé évolue de façon aléatoire, mais en moyenne on observe une tendance linéaire avec une pente $\mathbb{E}(\tilde{G}(t) | M^*)$ qui est plus grande pour la politique myope que pour les politiques constantes.

Nous avons ensuite comparé la politique myope de la [section 4.4](#) à la politique MDP de la [section 4.5](#). Nous avons considéré différentes valeurs pour le facteur d'amortissement ρ dans l'approche MDP : 0.01, 0.50 et 0.99. Nous avons calculé le nombre total de mesures et le gain pénalisé moyen $\frac{1}{T} \sum_{t=1}^T \tilde{G}(t)$ pour les quatre politiques : ne jamais mesurer, toujours mesurer, myope, MDP. Ces résultats ont été moyennés sur 100 trajectoires aléatoires. Les résultats sont présentés dans le [tableau 4.2](#).

	Jamais	Toujours	Myope	MDP		
				0.01	0.50	0.99
Mesures	0	3000	49.83	54.83	203.57	416.1
Gain délai	1.36	2.02	1.55	1.56	1.83	1.93
Gain pénalisé	1.36	1.37	1.53	1.55	1.78	1.83

TABLE 4.2 – Comparaison de la performance de quatre politiques : ne jamais mesurer, toujours mesurer, myope et MDP ($\rho = 0.01, 0.50, 0.99$). Statistiques calculées sur 100 trajectoires aléatoires.

On remarque que lorsque ρ tend vers 0, la performance de la politique myope et MDP sont proches, comme on peut s'y attendre. Quand ρ augmente de 0.01 à 0.99, le gain pénalisé moyen augmente, mais aussi le nombre de mesures, respectivement de 1.55 à 1.83 et de 54 à 416. Remarquons que pour le cas de $\rho = 0.5$, la performance en termes de gain pénalisé moyen est proche de la performance que l'on obtiendrait avec $\rho = 0.99$ (1.78 contre 1.83) mais que le nombre de mesures est divisé par 2 (203 au lieu de 416).

Une formule pour calculer le gain pénalisé moyen théorique est donnée dans [106, annexe. A]. En appliquant cette formule, on trouve un gain pénalisé moyen de 1.33 pour la politique ne jamais mesurer, 1.35 pour la politique toujours mesurer, et 1.52 pour la politique myope. On remarque que les valeurs obtenues par simulation sont très proches.

4.6.2 2 chemins stochastiques

Nous considérons désormais deux chemins stochastiques, chacun représenté par une chaîne de Markov à deux états. Les valeurs du

délai pour le premier chemin sont $l_1^1 = 0.5$ et $l_2^1 = 2$, et le coût de mesure est 0.05. Pour le second chemin, les valeurs du délai sont de $l_1^2 = 1$ et $l_2^2 = 3$, et le coût de mesure est défini à 0.15. Les matrices de transition sont :

$$P^{(1)} = \begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix}, P^{(2)} = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix} \quad (4.49)$$

Nous calculons d'abord la politique myope. Dans le cas de deux chemins, les récompenses (tableau 4.1) peuvent être exprimées en fonction de $x^1 = \gamma_{t-1,t}^{(1)}(1)$ et $x^2 = \gamma_{t-1,t}^{(2)}(1)$, puisque $\gamma_{t-1,t}^{(i)}(1) + \gamma_{t-1,t}^{(i)}(2) = 1$.

Comme discuté dans la section 4.4.1.2, nous n'avons pas réussi à obtenir de solution analytique pour les frontières de décision dans le cas de 2 chemins. Nous utilisons donc une approche numérique en discrétisant l'espace $[0, 1]^2$ du couple (x^1, x^2) . Pour chaque point (valeur du prédicteur) nous calculons la récompense pour chacune des actions de mesure possible $((0, 0), (0, 1), (1, 0), (1, 1))$ d'après le tableau 4.1, et nous choisissons la récompense maximale. Le résultat obtenu est présenté dans la figure 4.6, où les quatre régions de décision sont présentées avec des couleurs différentes.

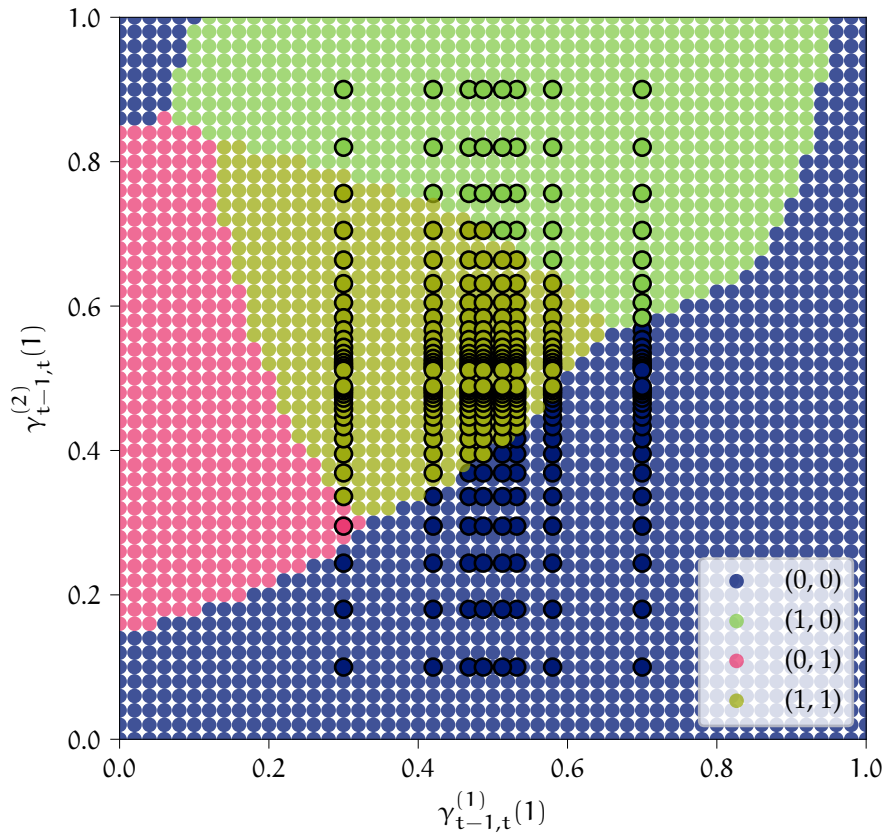


FIGURE 4.6 – Décisions de mesure pour la politique myope et MDP ($\rho = 0.01$) dans le cas de deux chemins stochastiques à deux états.

Nous pouvons vérifier que lorsqu'il n'y a pas d'incertitude sur l'état de l'un des chemins ($\gamma_{t-1,t}^{(i)}(1) = 0$ ou 1), alors la décision de mesure optimale pour l'autre chemin est consistante avec les résultats obtenus dans le cas d'un chemin déterministe, et d'un chemin modélisé par une chaîne de Markov à 2 états (équation 4.22).

Supposons qu'il n'y ait pas d'incertitude sur le chemin i , c'est à dire $x^i = 0$ ou $x^i = 1$. Le délai du chemin i est alors l , avec $l = l_2^i$ si $x^i = 0$ et $l = l_1^i$ si $x^i = 1$. On note j le second chemin. Si le coût de mesure c^j est supérieur à $c_{\max} = (l_2^j - l)(l - l_1^j)/(l_2^j - l_1^j)$ alors le délai sur le second chemin ne doit jamais être mesuré. Au contraire, si $c^j < c_{\max}$, alors la décision de mesure optimale dépend de la valeur du prédicteur $x^j = \gamma_{t-1,t}^j(1)$. Plus précisément, le délai du chemin doit être mesuré seulement si $x^j \in [x_{\min}, x_{\max}]$ avec $x_{\min} = c^j/(l - l_1^j)$ et $x_{\max} = 1 - c^j/(l_2^j - l)$.

Nous avons considéré quatre cas : $x^1 = 0$ ou 1 , et $x^2 = 0$ ou 1 . Pour chacun de ces cas, nous avons calculé les seuils c_{\max} , x_{\min} , x_{\max} qui doivent être utilisés pour prendre une décision de mesure sur l'autre chemin. Les résultats sont présentés dans les tableaux 4.3 et 4.4.

x^1	l^1	c_{\max}^2	x_{\min}^2	x_{\max}^2
0	2	$\frac{(l_2^2 - l)(l - l_1^2)}{l_2^2 - l_1^2} = 1$	$\frac{c^2}{l - l_1^2} = 0.15$	$1 - \frac{c^2}{l_2^2 - l} = 0.85$
1	0.5	$\frac{(l_2^2 - l)(l - l_1^2)}{l_2^2 - l_1^2} = -0.625$	$M^2 = 0$ car $c^2 = 0.15 > c_{\max}^2$	

TABLE 4.3 – Conditions de mesure sur la « bordure » de l'espace d'état lorsque l'état du chemin 1 est parfaitement connu.

x^2	l^2	c_{\max}^1	x_{\min}^1	x_{\max}^1
0	3	$\frac{(l_2^1 - l)(l - l_1^1)}{l_2^1 - l_1^1} = -1.67$	$M^1 = 0$ car $c^1 = 0.05 > c_{\max}^1$	
1	1	$\frac{(l_2^1 - l)(l - l_1^1)}{l_2^1 - l_1^1} = 0.33$	$\frac{c^1}{l - l_1^1} = 0.1$	$1 - \frac{c^1}{l_2^1 - l} = 0.95$

TABLE 4.4 – Conditions de mesure sur la « bordure » de l'espace d'état lorsque l'état du chemin 2 est parfaitement connu.

En particulier, on remarque que, comme $l_1^1 = 0.5 < l_2^1$ et $l_1^1 < l_2^1$, si $x^1 = 1$, alors la valeur de c_{\max}^2 est négative. La décision de mesure sur le chemin 2 est donc $M^2 = 0$ quelque soit le coût de mesure c^2 . De façon similaire, comme $l_2^2 = 3 > l_1^2$ et $l_2^2 > l_1^2$, si $x^2 = 0$ alors la valeur de c_{\max}^1 est négative. La décision de mesure sur le chemin 1 est donc $M^1 = 0$ quelque soit le coût de mesure c^1 . Dans les deux autres cas limite ($x^1 = 0$ et $x^2 = 1$), les bornes x_{\min} et x_{\max} ont été calculées. Ces bornes sont cohérentes avec la politique de la figure 4.6.

Finalement, nous avons comparé la politique myope à la politique MDP obtenue avec un petit facteur d'amortissement $\rho = 0.01$. Les décisions du MDP sont représentées par des points, et les mêmes

couleurs sont utilisées pour permettre la comparaison entre les deux politiques. Comme on peut le voir dans la [figure 4.6](#), les décisions sont les mêmes puisque ρ est très proche de 0. Remarquons que dans le cas de chaînes de Markov (en opposition à des modèles de Markov cachés), le prédicteur $\gamma_{t-1,t}^{(i)}$ prend un nombre fini de valeurs, de la forme $e_{\tau_{\text{last}}^i} (P^{(i)})^{\tau^i}$. C'est pour cela que les décisions MDP sont représentées par des points de couleurs dans l'espace $[0, 1]^2$, et que l'ensemble de l'espace $[0, 1]^2$ n'est pas échantillonné.

4.6.3 2 chemins entre des sondes RIPE Atlas

Dans les deux sections précédentes, nous avons comparé la performance des différentes politiques de mesure (statique, myope, MDP) sur des données synthétiques. Dans cette section nous comparons les politiques sur des données réelles issues de mesures sur Internet. Nous considérons 3 sondes RIPE Atlas localisées à Hong Kong (HK), en Lituanie (LV) et au Kazakstan (KZ), ainsi que deux chemins avec la même origine (HK) et la même destination (KZ) : le chemin Internet direct $\text{HK} \rightarrow \text{KZ}$ (chemin 1), et le chemin $\text{HK} \rightarrow \text{LV} \rightarrow \text{KZ}$ avec le nœud LV comme intermédiaire (chemin 2).

Le RTT entre chaque paire de nœuds est mesuré pendant 7 jours, toutes les 10 minutes, résultant en 1008 observations par paire. Le chemin indirect est construit en additionnant le délai $\text{HK} \rightarrow \text{LV}$ et $\text{LV} \rightarrow \text{KZ}$. Les performances du chemin direct et du chemin indirect sont représentées dans la partie haute de la [figure 4.7](#). On remarque que le chemin indirect est parfois plus court que le chemin direct. On modélise les deux chemins avec un HMM gaussien à deux états. Les distributions du délai pour le premier chemin sont $\mathcal{N}(350, 30)$ et $\mathcal{N}(440, 100)$. Les distributions du délai pour le second chemin sont $\mathcal{N}(320, 30)$ et $\mathcal{N}(370, 30)$. Les matrices de transition sont :

$$P^{(1)} = \begin{bmatrix} 0.995 & 0.005 \\ 0.05 & 0.95 \end{bmatrix}, P^{(2)} = \begin{bmatrix} 0.995 & 0.005 \\ 0.005 & 0.995 \end{bmatrix} \quad (4.50)$$

La politique MDP optimale a été calculée pour $\rho = 0.9$ avec l'algorithme d'itération sur la valeur, et en supposant que le coût de mesure de chaque chemin est de $c^1 = c^2 = 4$. Pour cela, nous avons simplifié les modèles HMM en chaînes de Markov, en ignorant le bruit gaussien et en supposant que le délai du chemin i dans l'état k est sa valeur moyenne l_i^k . Une fois que la politique de mesure μ^* est obtenue, elle est appliquée séquentiellement sur les séries temporelles de délai de $t = 0$ à $T = 1008$. À chaque instant t , les filtres des modèles de Markov cachés $\gamma_{t,t}^{(i)}$ sont mis à jour d'après l'[équation 4.7](#). L'état de la chaîne de Markov contrôlée est également mis à jour. Puisque l'on n'observe pas directement l'état du chemin i lorsque l'on réalise une mesure, l'état

l_j^i est choisi tel que j soit l'indice de l'état HMM le plus probable étant données les observations passées. C'est à dire $j = \operatorname{argmax}_j \gamma_{t,t}^{(i)}(j)$.

Dans notre jeu de données, à peu près 2% des données sont manquantes. Cela correspond aux cas où la sonde RIPE Atlas ne répond pas. Les données manquantes sont gérées en réalisant une prédiction : $\gamma_{t,t}^{(i)} = \gamma_{t-1,t-1}^{(i)} P^{(i)}$.

Les résultats sont présentés dans la [figure 4.7](#). Le délai des chemins est représenté dans la partie supérieure. La partie du milieu indique les instants de mesure, ainsi que la valeur des filtres $\gamma_{t,t}^{(i)}$. La partie du bas indique le chemin choisi à chaque instant ($C(t)$).

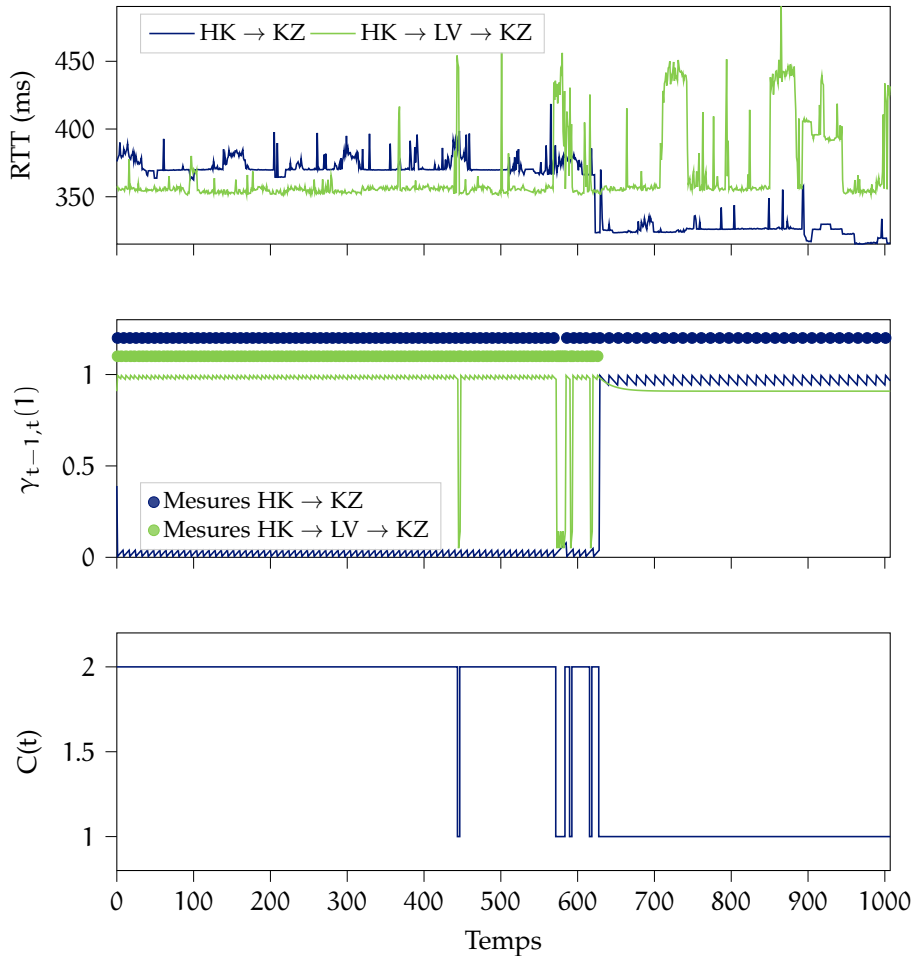


FIGURE 4.7 – Simulation de la politique MDP ($\rho = 0.9$) entre deux sondes RIPE Atlas. (Haut) Valeurs du délai observées. (Milieu) Instants de mesure et valeurs des prédicteurs. (Bas) Chemin choisi.

La fréquence de mesure est de 11% pour le lien direct, et de 12.5% pour le lien indirect. Cette fréquence dépend bien sûr du coût de mesure c^i et du facteur d'amortissement ρ . Plus ρ est important plus les mesures sont fréquentes, et plus c^i est important moins les mesures seront fréquentes. Le [tableau 4.5](#) résume les indicateurs de

performance. Tandis que le délai moyen sur le chemin direct est de 355 ms et de 367 ms sur le chemin indirect, il est de 345 ms sur le chemin optimisé. Le chemin optimisé est plus court de 10 ms que le chemin indirect.

On remarque également que le délai moyen avec la politique MDP est très proche (1 milliseconde) de celui avec une politique de mesure constante, alors que le nombre de mesures est dix fois plus faible avec la politique MDP. Il n'est donc pas nécessaire de mesurer constamment pour obtenir un routage quasi-optimal.

	Routage statique		Routage optimisé	
	Direct	Indirect	Toujours	MDP ρ 0.9
Mesures	0	0	3000	229
Délai moyen (ms)	355	367	344	345
Délai pénalisé	355	367	356	346

TABLE 4.5 – Évaluation de la politique MDP ($\rho = 0.9$) entre deux sondes RIPE Atlas. Comparaison avec un choix de chemin statique (direct ou indirect), et une mesure constante.

4.6.4 Simulation d'un overlay de routage

Dans la section précédente, nous avons comparé les politiques de mesure sur deux chemins issus de mesures RIPE Atlas. Les chemins utilisés dans la section précédente ont été choisis car ils illustrent bien le principe d'un overlay de routage, où le fait de faire transiter les données par un nœud intermédiaire permet parfois de réduire le délai. De plus, ces chemins sont modélisables par un modèle de Markov caché à 2 états, ce qui permet de visualiser simplement les valeurs des filtres au cours de la simulation.

Dans cette section, nous comparons les politiques sur 277 paires de 2 chemins, également issues de mesures RIPE Atlas, et nous montrons qu'il est possible de diminuer le délai moyen en relayant les paquets, et ce sans mesurer en permanence. Afin d'automatiser l'apprentissage du modèle pour chaque chemin, nous utilisons le HDP-HMM présenté dans le [chapitre 2](#), et nous n'imposons pas de limite sur le nombre d'états par chemins.

4.6.4.1 Construction de l'overlay

Afin de simuler un overlay de routage comportant 30 nœuds, nous choisissons aléatoirement 30 ancres RIPE Atlas : 5 sur chacun des 6 continents. Comme présenté dans la [section 3.1.3](#), les ancres Atlas effectuent toute les 4 minutes des mesures de délai entre elles. Ces mesures sont effectuées entre chaque paire d'ancre dans une topologie

de graphe complet, de façon analogue à la topologie d'un overlay de routage. Ainsi, le *chemin direct* entre deux nœuds de l'overlay est simulé par les mesures de délai entre deux ancrés. Le *chemin alternatif* est construit en additionnant les mesures entre l'ancre origine et l'ancre intermédiaire, et entre l'ancre intermédiaire et l'ancre destination. Parmi toutes les ancrés intermédiaires nous choisissons celle qui permet d'obtenir le chemin alternatif le plus court en moyenne.

Nous avons conservé les paires origine-destination de deux chemins pour lesquelles le chemin alternatif est plus court au-moins 10% du temps, résultant en 277 paires origine-destination. Nous avons appris le modèle pour chaque chemin à l'aide du HDP-HMM à émissions DPMM.

Finalement, nous définissons le *chemin optimal* comme le chemin qui résulterait d'une décision de routage optimale à chaque instant, si les valeurs exactes du délai sur chaque chemin étaient connues. Le délai du chemin optimal est toujours le plus faible.

4.6.4.2 Simulation

Pour chaque paire origine-destination, nous avons simulé deux politiques de mesure : la politique « toujours mesurer » et la politique obtenue par résolution du MDP. On fixe le coût de mesure c pour chaque chemin à 1, τ_{\max} à 100, et le facteur d'amortissement ρ du MDP à 0.9.

Remarquons que deux implémentations du « simulateur » sont possibles. Puisque nous considérons ici des modèles de Markov cachés, la mesure du délai ne nous permet pas d'être certain de l'état caché du chemin. Le filtre HMM (équation 4.7) nous donne la probabilité qu'un chemin soit dans un état particulier, étant données les mesures passées. Lorsqu'une mesure est réalisée par le simulateur, deux approches sont alors possibles afin de déterminer le chemin le plus court : utiliser le délai espéré du chemin après mise à jour du filtre HMM associé, ou utiliser la valeur mesurée du délai.

Nous choisissons ici d'utiliser le délai espéré calculé avec le filtre HMM, car cette approche nous permet de valider la modélisation HMM des chemins. En effet, si la modélisation HMM capture correctement la dynamique du délai sur les chemins, alors le délai obtenu avec la politique « toujours mesurer » devrait être proche du délai du chemin optimal défini ci-dessus. Au contraire, si nous utilisons la valeur mesurée du délai, alors le délai de la politique « toujours mesurer » serait égale au délai du chemin optimal.

4.6.4.3 Résultats

Les résultats de simulation sont présentés dans le [tableau 4.6](#). Les colonnes « direct » et « alternatif » correspondent aux cas où l'on choisit toujours le chemin direct, ou toujours le chemin alternatif. La première

ligne correspond au nombre de mesure moyen par pas de temps, sur les deux chemins. Une valeur de 0 indique que l'on ne mesure jamais, tandis qu'une valeur de 2 indique que les 2 chemins sont mesurés en permanence. La seconde ligne indique le pourcentage de temps pendant lequel le délai obtenu est égal au délai du chemin optimal, tel que défini ci-dessus. Finalement, la troisième ligne indique l'écart moyen au délai optimal. Ces valeurs sont calculées pour chacune des 277 paires, et les valeurs médianes sont reportées dans le tableau.

	Routage statique		Routage optimisé	
	Direct	Alternatif	Toujours	MDP $\rho = 0.9$
Mesures	0	0	2	0.12
Délai opt. (%)	73	27	82	80
Écart délai opt. (ms)	2.02	5.9	0.34	0.6

TABLE 4.6 – Comparaison des politiques de routage statiques et optimisés. Valeurs médianes calculées sur 277 paires origine-destination.

On remarque tout d'abord que l'utilisation d'une politique de routage optimisée, qui choisit dynamiquement entre le chemin direct et le chemin alternatif, permet de réduire considérablement le délai moyen. En effet, le chemin direct est en moyenne plus long de 2.02 ms par rapport au chemin optimal. On remarque également que le délai moyen obtenu avec la politique de mesure constante est très proche du délai optimal (0.34 ms). Comme discuté ci-dessus, cela confirme que la modélisation HMM capture suffisamment bien les différents états du délai et leur dynamique.

Concernant la politique MDP, on constate que le délai obtenu avec cette politique est très proche (0.6 ms) du délai optimal, alors que chaque chemin n'est mesuré en moyenne qu'une fois sur dix. Cela correspond à une réduction du coût de mesure de 94% par rapport à une politique de mesure constante. Ce résultat montre que l'approche MDP fournit une politique de mesure parcimonieuse, où un petit nombre de mesures dans des états particuliers fournissent suffisamment de connaissance sur l'état du réseau, pour choisir le chemin le plus court, la plupart du temps.

4.6.4.4 Impact du coût de mesure

Afin d'étudier l'effet du coût de mesure sur la fréquence des mesures, nous avons reproduit les simulations précédentes sur les 277 paires, en variant le coût de mesure. Nous avons comparé le délai moyen de la politique MDP par rapport à la politique « toujours mesurer », pour différentes valeurs de c . La [figure 4.8](#) présente la distribution du nombre de mesures moyen sur toutes les paires de deux chemins, en fonction du coût de mesure. La [figure 4.9](#) présente

la distribution de l'écart de délai par rapport à la politique « toujours mesurer », en fonction du nombre moyen de mesures.

Lorsque le coût de mesure c est grand, la politique MDP tend vers la politique « ne jamais mesurer ». À l'inverse, lorsque le coût de mesure est faible, la politique MDP tend vers la politique « toujours mesurer ». Remarquons que pour les très petites valeurs de c , le nombre médian de mesures est nettement inférieur à 2 (il tombe rapidement à 0.5). En effet, les performances des chemins sont très stables, et pour la plupart des chemins, la probabilité de rester dans l'état actuel est extrêmement élevée. Le modèle HDP-HMM et la politique MDP saisissent le fait que, même si le coût de mesure est très faible, il n'est pas toujours nécessaire de mesurer pour prendre une décision de routage optimale.

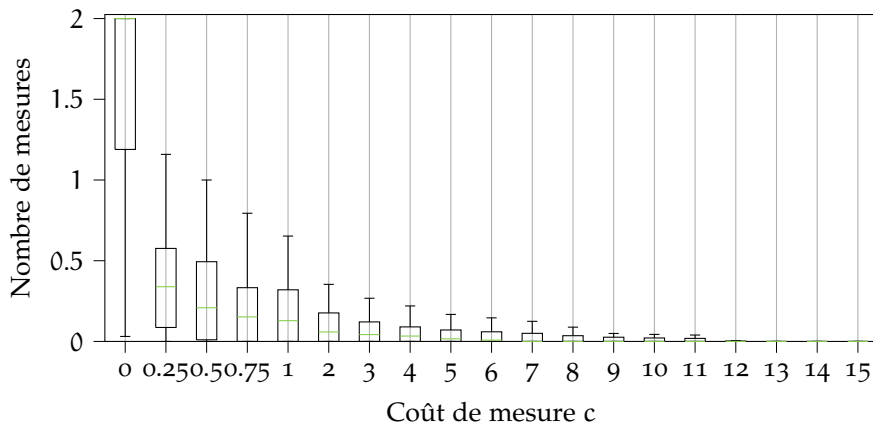


FIGURE 4.8 – Nombre de mesures en fonction du coût de mesure.

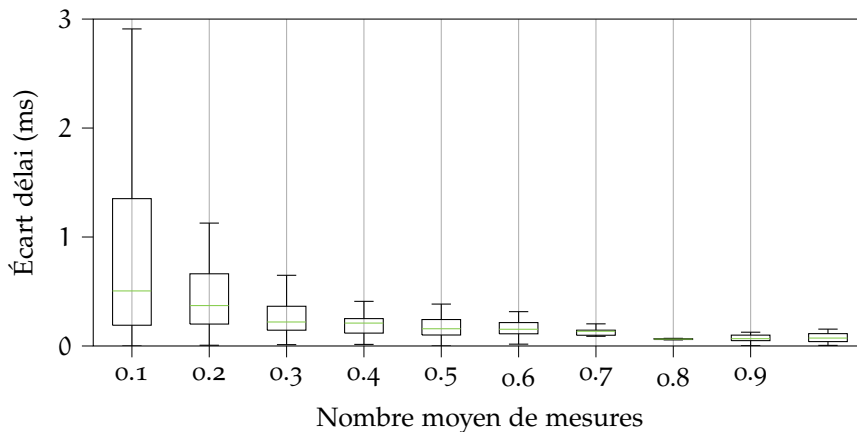


FIGURE 4.9 – Écart au délai de la politique toujours mesurer en fonction du nombre moyen de mesures (regroupés dans des intervalles de 0.1).

HEURISTIQUES POUR LA MÉTROLOGIE PARCIMONIEUSE

La méthode de métrologie parcimonieuse présentée dans le chapitre précédent permet de réduire considérablement le nombre de mesures dans un overlay de routage. Cependant la méthode souffre d'une complexité importante qui limite son applicabilité sur des scénarios réalistes. En effet, le nombre d'états du problème est exponentiel avec le nombre de chemins. En pratique cela rend la méthode trop coûteuse en ressources de calcul pour l'appliquer sur plus de 3 chemins (entre une origine et une destination).

Dans cette section nous détaillons des algorithmes de résolution approximatifs (ou « heuristiques ») dont la complexité est beaucoup plus faible, ce qui permet de résoudre le problème pour plus de 3 chemins.

Dans la [section 5.1](#) nous détaillons la structure du problème. Dans la [section 5.2](#) nous présentons l'algorithme *receding horizon*, une version en-ligne de l'algorithme d'itération sur la valeur. Dans la [section 5.3](#) nous présentons l'algorithme d'agrégation de chemins qui permet de résoudre le problème avec un coût linéaire avec le nombre de chemins. Enfin, dans la [section 5.4](#) nous évaluons la performance de ces algorithmes, sur des données synthétiques et réelles.

5.1 STRUCTURE DU PROBLÈME

Nous avons vu, dans le chapitre précédent, que la politique myope pour un chemin stochastique est une politique à seuil ([figure 4.3](#)). Nous avons également vu que dans le cas d'une politique MDP avec deux chemins stochastiques, la politique est constante sur des régions connexes ([figure 4.6](#)).

Dans cette section, nous montrons comment la structure du problème implique une politique optimale à seuil. Nous détaillons également la complexité du problème et proposons une modification du problème pour réduire la complexité du problème d'exponentielle à polynomiale avec le nombre de chemins.

5.1.1 Espace d'état

Le filtre $\gamma^{(i)}$ représente la connaissance sur les états cachés du chemin i . Il prend ses valeurs dans le simplexe à $(K^i - 1)$ dimensions :

$$\left\{ \gamma^{(i)} \in \mathbb{R}^{K^i} : \mathbf{1}'\gamma^{(i)} = 1, \quad 0 \leq \gamma^{(i)}(j) \leq 1 \quad \forall j \in \{1, \dots, K^i\} \right\} \quad (5.1)$$

L'espace d'état du problème est le produit de l'espace (le simplexe) de chaque filtre. Ainsi, dans le cas de 1 chemin à 2 états, l'espace d'état est le segment unité (figure 4.3). Dans le cas de 2 chemins à 2 états, il s'agit du carré unité (figure 5.1). Les flèches représentent le sens dans lequel on se déplace lorsqu'on ne mesure pas.

Si on ne mesure jamais, les filtres convergent vers la distribution stationnaire de la chaîne de Markov modélisant le délai du chemin, ici notée $\pi_1^{(1)}$ pour le chemin 1 et $\pi_1^{(2)}$ pour le chemin 2.

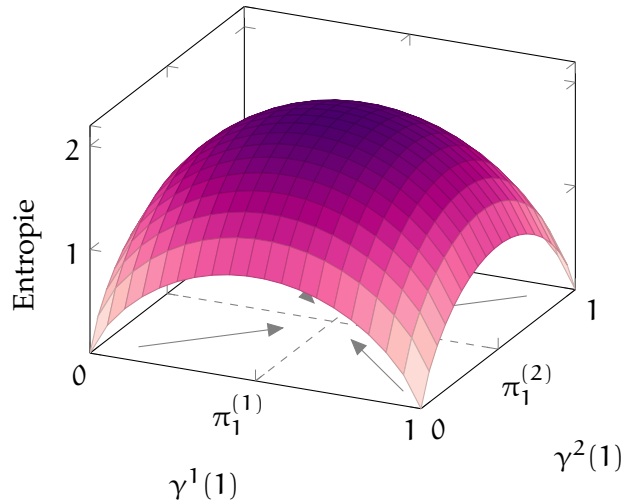


FIGURE 5.1 – Espace d'état et entropie du filtre pour 2 chemins de 2 états. π représente la distribution stationnaire.

5.1.1.1 Entropie et incertitude

Chaque élément de l'espace d'état est un vecteur de probabilité, qui donne la probabilité que chaque chemin soit dans un état particulier, étant donné les mesures passées. Une façon de visualiser l'incertitude associée à un état consiste à utiliser l'entropie. L'entropie (de Shannon) se calcule ainsi pour un chemin :

$$H^i = - \sum_j \gamma^{(i)}(j) \log_2 \gamma^{(i)}(j). \tag{5.2}$$

Elle vaut 0 lorsqu'on est certain de l'état du chemin (après une mesure), et elle augmente lorsqu'on réalise des prédictions. L'entropie sur l'ensemble des chemins, est la somme de l'entropie sur chaque chemin : $H = \sum_i H^i$.

L'entropie en fonction de l'état dans le cas de 2 chemins de 2 états est représentée dans la figure 5.1. Elle est minimale lorsqu'on est certain de l'état de l'ensemble des chemins, et elle augmente lorsqu'on fait des prédictions.

5.1.1.2 Discrétisation

Dans le cas d'une chaîne de Markov, le filtre ne peut prendre qu'un ensemble dénombrable de valeurs : $\gamma^{(i)} = e_{L_{\text{last}}^i} (P^{(i)})^{\tau^i}$ où L_{last}^i est le dernier état mesuré, et τ^i le nombre de pas de temps depuis la dernière mesure. Cet ensemble de valeurs dépend de la matrice de transition des chemins. Un exemple est donné dans la [figure 5.2](#). Dans le cas d'un modèle de Markov caché nous nous ramenons à un espace d'état discret en définissant L_{last}^i comme l'état caché le plus probable lorsque l'on mesure.

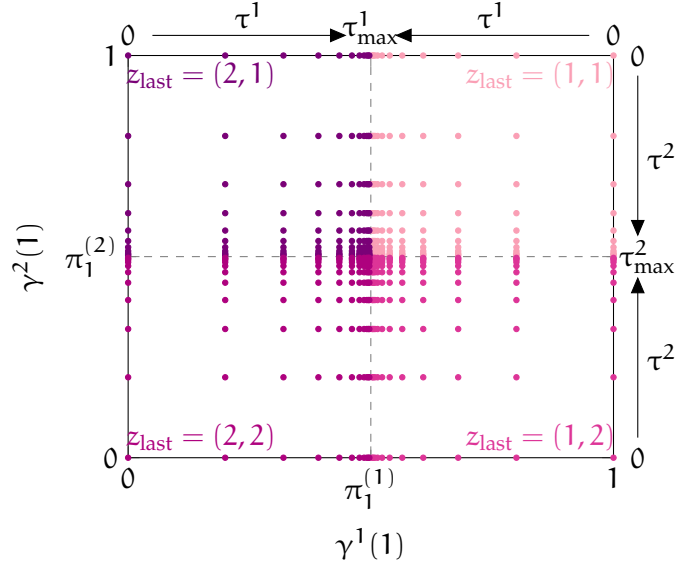


FIGURE 5.2 – Espace d'état discret contenu dans l'espace d'état continu pour 2 chemins de 2 états. π représente la distribution stationnaire.

5.1.2 Transitions

Pour chaque chemin, trois types de transitions sont possibles :

$$\begin{cases} (\tau^i, L_{\text{last}}^i) \rightarrow (\tau^i + 1, L_{\text{last}}^i) & \text{si } M^i = 0 \text{ et } \tau^i < \tau_{\text{max}}^i \\ (\tau_{\text{max}}^i, L_{\text{last}}^i) \rightarrow (\tau_{\text{max}}^i, L_{\text{last}}^i) & \text{si } M^i = 0 \text{ et } \tau^i = \tau_{\text{max}}^i \\ (\tau^i, L_{\text{last}}^i) \rightarrow (0, l_j^i), \quad j = 1, \dots, K^i & \text{si } M^i = 1. \end{cases} \quad (5.3)$$

Remarquons que pour une politique donnée, tous les états pour lesquels τ^i est supérieur au premier instant de mesure du chemin i sont inatteignables. En effet, dès lors que l'on mesure le chemin i , τ^i est remis à zéro. Remarquons également que tous les états pour lesquels $\tau^i = 0$ sont situés sur la « bordure » de l'espace d'état. Les politiques de mesure sont donc des politiques à seuil : elles peuvent être résumées par le nombre de pas de temps avant le premier instant de mesure d'un chemin, en partant d'un état de bordure.

Cela est cohérent en terme d'information, puisque cela signifie que l'on ne peut jamais atteindre un état où l'on possède moins d'informations sur un chemin que l'état dans lequel on mesure. L'action de mesurer réduit toujours l'incertitude.

5.1.3 Complexité

La complexité des algorithmes de résolution des processus de décision markoviens dépend du nombre d'états $|S|$ et du nombre d'actions $|A|$ dans le problème. Afin de simplifier les calculs suivants, nous supposons, sans perte de généralité, que tous les chemins ont le même nombre d'états.

Considérons P chemins à K états. Nous avons $|S| = (K \times \tau_{\max})^P$ états, et $\binom{P}{k} = \frac{P!}{k!(P-k)!}$ actions où l'on mesure k chemins parmi P . Le nombre d'états atteignables après une action dépend du nombre de chemins mesurés :

- Si aucun chemin n'est mesuré, alors seulement l'état pour lequel τ^i est incrémenté de $\mathbf{1}$ pour tous les chemins est atteignable.
- Si un seul chemin est mesuré alors il y a K états atteignables. $(\tau^i = 0, l_j^i)$ avec $j = 1, \dots, K$ pour le chemin mesuré. Les autres chemins ne pourront atteindre que l'état pour lequel τ^i est incrémenté de $\mathbf{1}$.
- De manière générale, si p chemins sont mesurés, alors K^p états sont atteignables, puisque chaque mesure peut prendre K valeurs différentes.

Ainsi le nombre d'états atteignables depuis un état donné, en considérant toute les actions possibles, est

$$\binom{P}{0}K^0 + \binom{P}{1}K^1 + \dots + \binom{P}{P}K^P = (1 + K)^P. \quad (5.4)$$

De façon équivalente, on peut également remarquer que chacun des P chemins peut prendre $(1 + K)$ valeurs futures : soit $(\tau^i + 1, L_{\text{last}}^i)$ si on ne mesure pas, soit $(0, j)$ avec $j \in \{1, \dots, K\}$ si on mesure. On retrouve alors $(1 + K)^P$ états atteignables en prenant en compte l'ensemble des chemins.

Remarquons que si on limite le nombre de chemins pouvant être mesurés à chaque instant à $\mathbf{1}$, alors on obtient une expression linéaire avec P et non plus exponentielle :

$$\binom{P}{0}K^0 + \binom{P}{1}K^1 = 1 + KP. \quad (5.5)$$

Pour chaque état s , l'algorithme d'itération sur la valeur requiert $\mathcal{O}(|S'|)$ opérations où $|S'|$ est le nombre d'états atteignables à partir de s , en prenant en compte toute les actions possibles. Dans notre problème, et comme nous l'avons vu ci-dessus, le nombre d'actions et le nombre d'états atteignables ne dépendent pas de l'état courant.

Chaque itération de l'algorithme d'itération sur la valeur requiert donc $\mathcal{O}(|S| \times |S'|)$ opérations, où $|S|$ est le nombre total d'états du problème. Soit de l'ordre de $(K \times \tau_{\max})^P \times (1 + K)^P$ opérations dans le cas général, et $(K \times \tau_{\max})^P \times (1 + KP)$ si l'on se restreint à la mesure d'un seul chemin au maximum. Remarquons que le terme $(K \times \tau_{\max})^P$ est dominant car τ_{\max} est en général grand. Dans les sections suivantes nous présentons des méthodes qui permettent de réduire la complexité.

5.2 OPTIMISATION À HORIZON DE TEMPS RECLANT

La fonction objectif définie dans la section précédente (équation 4.43) est une somme sur un nombre infini de termes :

$$J_{\pi}(\mathbf{s}_0) = E_{\pi} \left(\sum_{t=1}^{\infty} \rho^t R(\mathbf{s}_t) \mid \mathbf{s}_0 \right). \quad (5.6)$$

Une manière d'approximer un MDP à horizon de temps infini, et de réduire son coût de sa résolution, consiste à limiter cette somme à un nombre fini de termes, pour obtenir un MDP à horizon de temps fini. Cette approximation fournit une politique $\pi = \{\pi_0, \pi_1, \dots, \pi_{H-1}\}$ qui commence au temps t et se finit au temps $t + H - 1$ où H est l'horizon de temps considéré. Cependant, cette approche fournit une politique non stationnaire : l'action optimale pour un même état dépend de l'instant de décision.

Une politique stationnaire peut être obtenue par l'approche *receding horizon* (« à horizon de temps reculant ») [45, ch. 4] :

1. Au temps t et pour l'état courant \mathbf{s} , résoudre le MDP sur un horizon de temps fini $\{t, t + H - 1\}$.
2. Appliquer uniquement la première action de la politique résultante, $\pi_0(\mathbf{s})$.
3. Déterminer l'état atteint au temps $t + 1$.
4. Recommencer l'étape 1 au temps $t + 1$ sur l'horizon $\{t + 1, t + H\}$.

On parle d'horizon de temps reculant, puisqu'à chaque pas de temps, on choisit la décision qui maximise la récompense courante, plus les récompenses espérées jusqu'à H pas en avant. Ainsi, le futur est pris en compte, mais à un niveau de complexité plus faible, puisque le nombre d'états considérés est largement réduit.

L'approximation à horizon de temps reculant est implémentée de façon similaire à l'algorithme d'itération sur la valeur (équation 4.46), mais sur un horizon de temps fini H et à partir de l'état courant \mathbf{s} :

$$V_h(\mathbf{s}) = \max_{\mathbf{M} \in \mathcal{M}} \sum_{\mathbf{s}'} \mathbb{P}(\mathbf{s}' \mid \mathbf{s}, \mathbf{M}) (R(\mathbf{s}') + \rho V_{h-1}(\mathbf{s}')) \quad (5.7)$$

avec $V_0(\mathbf{s}) = 0$.

Il s'agit d'un algorithme *en ligne*, qui est exécuté à chaque pas de temps à partir de l'état courant \mathbf{s} , en opposition à l'algorithme

d'itération sur la valeur qui est un algorithme *hors ligne*, exécuté une seule fois sur tous les états. Au lieu de calculer la politique optimale pour l'ensemble des $(K \times \tau_{\max})^P$ états avec l'algorithme d'itération sur la valeur, on ne considère ici que les $((1 + K)^P)^H$ états atteignables à partir de l'état courant en un horizon de temps H . Cette quantité reste grande, même pour des petites valeurs de P et H . Cependant, comme nous allons le voir dans la section suivante, il existe des états redondants parmi les $((1 + K)^P)^H$ états, ce qui nous permet d'atteindre une complexité plus faible.

5.2.1 Complexité

5.2.1.1 Cas général

En considérant toutes les actions possibles, il existe $(1 + K)^P$ états atteignables à partir d'un autre état (équation 5.4). Il existe donc $((1 + K)^P)^H$ états atteignables à un horizon de temps H , en partant d'un état particulier. Cependant, le nombre d'états distincts à un horizon de temps H est bien plus petit.

Considérons un chemin i dans l'état $(\tau^i, L_{\text{last}}^i)$ et regardons les états qu'il peut prendre H pas en avant. Si le chemin n'est jamais mesuré, alors il sera dans l'état $(\tau^i + H, L_{\text{last}}^i)$. Si le chemin a été mesuré pour la dernière fois au pas h , alors il sera dans l'un des états $(H - h, l_j^i)$ où j peut prendre K valeurs. Il existe ainsi $1 + KH$ états distincts pour un chemin à l'horizon H , et $(1 + KH)^P$ états distincts pour P chemins. Remarquons que le nombre d'états ne dépend plus de τ_{\max} ($(K \times \tau_{\max})^P$) mais de H , qui est en pratique beaucoup plus petit.

Cette optimisation peut être implémentée simplement dans le calcul de $V_H(s)$ (équation 5.7), en mettant en cache les valeurs de V pour le couple (h, s) dans une structure de données qui permet l'accès à une valeur en temps constant (un dictionnaire, par exemple).

5.2.1.2 Limitation à 1 seul chemin mesuré

Considérons maintenant le cas où on ne mesure qu'un seul chemin à chaque instant. Le nombre d'états atteignables à partir d'un état donnée est de $1 + KP$ dans ce cas (équation 5.5), au lieu de $(1 + K)^P$ dans le cas général (équation 5.4).

En H pas, on peut mesurer $n \leq \min(H, P)$ chemins différents. En effet, si l'horizon de temps est supérieur ou égal au nombre de chemins ($H \geq P$), alors on peut mesurer P chemins différents. Au contraire, si l'horizon de temps est inférieur au nombre de chemins ($H < P$), alors on ne pourra mesurer que H chemins différents.

Supposons qu'un chemin différent soit mesuré à chaque instant, et qu'on ne mesure pas deux fois le même chemin. Chaque chemin peut prendre K états différents lorsqu'il est mesuré. Au bout de H pas, le système peut donc être dans K^n états différents.

Il existe $C_P^n = \frac{P!}{n!(P-n)!}$ façons de choisir les n chemins que l'on va mesurer. Remarquons que si $H \geq P$, tous les chemins sont mesurés, il n'existe donc qu'une seule combinaison : $C_P^P = 1$. Ces n mesures peuvent être ordonnancés de $A_H^n = \frac{H!}{(H-n)!}$ façons différentes, l'ordre des mesures impactant les états finaux.

On obtient donc le nombre d'états distincts suivant :

$$\sum_{n=0}^{\min(H,P)} A_H^n C_P^n K^n = 1 + \sum_{n=1}^{\min(H,P)} \frac{H!}{(H-n)!} \frac{P!}{(P-n)!n!} K^n. \quad (5.8)$$

Dans le cas où $P = 1$ on retrouve l'expression $1 + KH$, et dans le cas où $H = 1$ on obtient $1 + KP$. Finalement, quand $P > H$, le nombre d'états distincts est de $\mathcal{O}(P^H K^H)$.

5.2.1.3 Exemple numérique

Considérons un cas réaliste, où $K = 8$, $P = 5$ et $H = 3$. Il existe $(1 + KH)^P = 9765625$ états dans le cas général, et 34681 états si on se limite à la mesure d'un seul chemin au maximum à chaque instant.

5.3 AGRÉGATION DE CHEMINS

L'approche *receding horizon* présentée dans la section précédente permet de réduire considérablement le coût de résolution du MDP. Au lieu de résoudre le problème hors-ligne pour $(K \times \tau_{\max})^P$ états, il suffit de considérer $(1 + KH)^P$ états à chaque pas de temps, dans le cas général. Cependant, ce nombre d'états reste exponentiel en fonction du nombre de chemins. Dans cette section nous proposons une heuristique qui permet d'obtenir une complexité linéaire avec le nombre de chemins.

À chaque pas de temps, et pour chaque chemin i , nous proposons de résoudre un MDP à 2 chemins (quelque soit le nombre total de chemins dans le problème) : le chemin stochastique i , et un chemin virtuel déterministe qui « résume » l'ensemble des autres chemins. Deux actions sont possibles dans ce MDP : « mesurer le chemin stochastique », et « ne pas mesurer le chemin stochastique ».

Le délai, déterministe, du chemin virtuel est défini comme le délai espéré sur les P chemins en excluant le chemin i :

$$\bar{D}_{-i}(\mathbf{s}_{-i}, \mathbf{M}^{-i}) = \sum_{\mathbf{s}'_{-i} \in \mathcal{S}_{-i}} \mathbb{P}(\mathbf{s}'_{-i} | \mathbf{s}_{-i}, \mathbf{M}^{-i}) D(\mathbf{s}'_{-i}) \quad (5.9)$$

où les probabilités de transition $\mathbb{P}(\mathbf{s}'_{-i} | \mathbf{s}_{-i}, \mathbf{M}^{-i})$ et le délai espéré minimum $D(\mathbf{s}'_{-i})$ sont définis de façon analogue au chapitre précédent (équations 4.36 et 4.39), en excluant le chemin i :

$$\mathbb{P}(\mathbf{s}'_{-i} | \mathbf{s}_{-i}, \mathbf{M}^{-i}) = \prod_{j \neq i} \mathbb{P}(s'_j | s_j, M^j) \quad (5.10)$$

et

$$\begin{aligned}
D(\mathbf{s}'_{-i}) &= \min_{j \neq i} \mathbb{E}(L^j(t) \mid I_t^j) \\
&= \min \left(\underbrace{\min_{j: \tau^i \geq 1, j \neq i} \mathbb{E}(L^j \mid \tau^j, L_{\text{last}}^j)}_{\text{chemins non mesurés}}, \underbrace{\min_{j: \tau^i = 0, j \neq i} L_{\text{last}}^j}_{\text{chemins mesurés}} \right). \quad (5.11)
\end{aligned}$$

Autrement dit, étant donné une action \mathbf{M} , le délai du chemin virtuel est le délai espéré si le chemin i n'existait pas, et qu'on appliquait l'action \mathbf{M} dans l'état courant.

Comme *receding horizon*, l'algorithme d'agrégation de chemins est appliqué en ligne. À chaque pas de temps, on initialise $\mathbf{M} = (0, \dots, 0)$, puis pour chaque chemin i on met à jour M^i d'après la solution du MDP comportant le chemin i , et le chemin déterministe de délai $\bar{D}_{-i}(\mathbf{s}_{-i}, \mathbf{M}^{-i})$. On applique ensuite l'action \mathbf{M} , et on recommence la procédure au pas de temps suivant. Cela requiert de résoudre P MDPs avec chacun un espace d'état comportant $K \times \tau_{\max}$ états.

5.4 RÉSULTATS

Nous proposons deux types de résultats de validation. Nous décrivons d'abord les résultats obtenus avec un petit scénario synthétique pour lequel la politique optimale peut être calculée. Le but est d'évaluer et de comparer la précision des approximations proposées. Nous évaluons ensuite la robustesse et le passage à l'échelle des solutions proposées en utilisant des mesures de délai collectées sur Internet.

5.4.1 Données synthétiques

5.4.1.1 Premier exemple

Dans le premier exemple, nous montrons que la politique myope peut avoir des performances relativement mauvaises comparées aux deux approximations. Nous considérons deux chemins de deux états chacun. Les délais sont identiques pour les deux chemins : $\mathbf{l}^1 = \mathbf{l}^2 = (0.01, 100)$. La matrice de transition est :

$$P_1 = P_2 = \begin{bmatrix} \beta & 1 - \beta \\ 1 - \beta & \beta \end{bmatrix}. \quad (5.12)$$

Si β est égal à 0.5, le fait de mesurer l'état d'un chemin n'apporte aucune information sur les évolutions futures. En effet, peu importe l'état mesuré à l'instant t , le chemin a la même probabilité (0.5) de se trouver dans l'état 1 ou 2 à l'instant $t + 1$. Au contraire, lorsque β est proche de 0 ou 1, le fait de mesurer l'état courant apporte beaucoup d'informations sur l'état futur. Par exemple, si β est proche de 0 et que l'on mesure l'état 1 pour un chemin, alors il y a une probabilité proche de 1 que le chemin se trouve dans l'état 2 à l'instant suivant.

Pour différentes valeurs de β , nous avons calculé la politique optimale, ainsi que les politiques obtenues avec les heuristiques *receding horizon* et à agrégation de chemins, pour tous les états de l'espace d'état. On choisit un coût de mesure $c = 25$, et un facteur d'amortissement $\rho = 0.99$.

Nous utilisons l'algorithme d'évaluation de politique (équation 4.48), pour obtenir la fonction de valeur à partir d'une politique donnée. Pour l'approche *receding horizon* et à agrégation de chemin, nous calculons l'erreur relative moyenne (MRE) par rapport à la fonction de valeur optimale :

$$\frac{1}{|S|} \sum_s \frac{V_\mu(s) - V^*(s)}{V^*(s)}. \quad (5.13)$$

Les résultats sont donnés dans la figure 5.3. On remarque que la politique myope diffère largement de la politique optimale lorsque les transitions de la chaîne de Markov sont très fréquentes ou peu fréquentes, et qu'il y a une grande différence entre le délai des deux états.

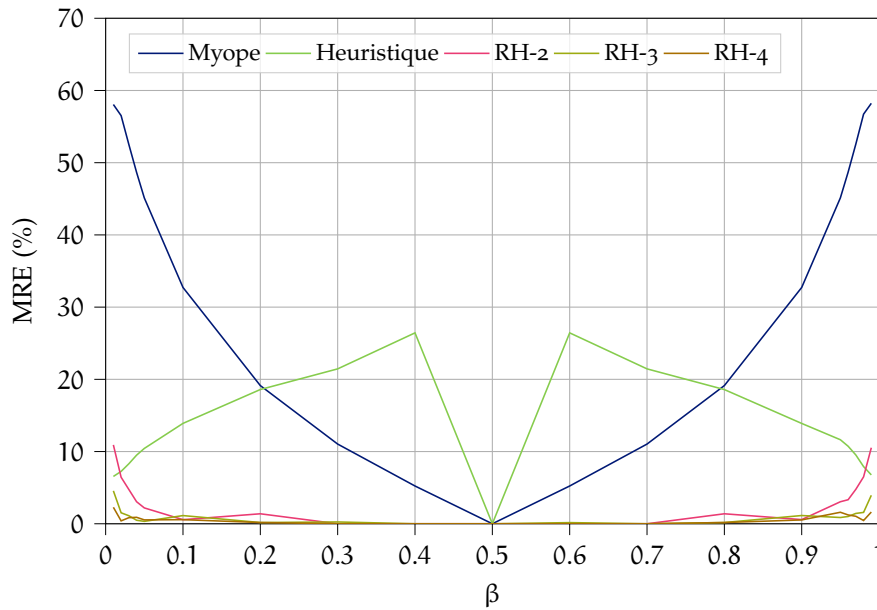


FIGURE 5.3 – Comparaison de l'erreur relative moyenne des différentes politiques par rapport à la politique MDP optimale, pour deux chemins synthétiques et différentes valeurs de β (probabilité d'auto-transition). $c = 25$ et $\rho = 0.99$.

5.4.1.2 Second exemple

Dans le second exemple, nous montrons que la politique myope peut être plus proche de la politique optimale que les méthodes approximées dans certains cas. Nous considérons trois chemins avec

les délais suivants : $\mathbf{l}^1 = (1, 3)$, $\mathbf{l}^2 = (0.5, 4)$ et $\mathbf{l}^3 = (0.25, 3.75)$. Les matrices de transition sont les suivantes :

$$P_1 = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}, P_2 = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}, P_3 = \begin{bmatrix} 0.65 & 0.35 \\ 0.35 & 0.65 \end{bmatrix}. \quad (5.14)$$

Remarquons que l'espérance du délai dans la distribution stationnaire des trois chemins est proche : 1.99, 1.9 et 2. On choisit $\tau_{\max} = 20$ pour les trois chemins, ce qui donne 64000 états possibles pour les valeurs du filtre.

ρ	c	MRE (%)		
		Heuristique	Horizon reculant	Myope
0.99	0.5	5.56	0.10	0.61
0.99	0.25	5.26	0.20	0.87
0.99	0.125	7.07	0.10	0.10
0.99	0.0625	5.20	0.15	0.15
0.999	0.5	5.57	0.21	0.72
0.999	0.25	5.45	0.30	1.00
0.999	0.125	7.35	0.23	0.23
0.999	0.0625	5.3	0.27	0.27

TABLE 5.1 – Comparaison de l'erreur relative moyenne des différentes politiques par rapport à la politique MDP optimale, pour trois chemins synthétiques et différentes valeurs de c et ρ .

L'erreur relative moyenne pour les différentes politiques est donnée dans le [tableau 5.1](#). On remarque que pour les deux exemples précédents les deux approximations sont relativement proches de la politique optimale, tandis que la politique myope a des performances plus variables.

5.4.2 Données réelles

Afin de valider nos méthodes sur des données réelles, où la propriété markovienne n'est pas toujours respectée, nous simulons un overlay de routage en utilisant des mesures de RTT NLNOG. Ces mesures de RTT ont été collectées en 2014 lors d'une expérience de mesure où 20 nœuds du *ring* NLNOG ont été utilisés. Le *ring* NLNOG est une infrastructure de mesure composée de 550 nœuds répartis dans 57 pays [73]. Le RTT est mesuré entre chaque paire de nœuds toutes les deux minutes, sur une période de cinq jours en utilisant des pings ICMP, de façon similaire au système d'ancres RIPE Atlas ([section 3.1.3](#)). Chaque paire origine-destination comporte 3836 échantillons. Ce jeu de données est particulièrement utile puisque les mesures sont réalisées entre chaque

nœud dans une topologie de graphe complet (*full mesh*). Ainsi nous pouvons simuler le délai entre deux nœuds A et B passant par un nœud intermédiaire C en ajoutant le délai entre A et C et C et B. Pour chaque paire dans le jeu de données, nous avons choisi les chemins alternatifs passant par un et un seul nœud intermédiaire, qui ont un délai plus court au moins 10% du temps.

Finalement, comme dans la [section 4.6.4](#), nous définissons le *chemin optimal* comme le chemin qui résulterait d'une décision de routage optimale à chaque instant, si les valeurs exactes du délai sur chaque chemins étaient connues. Le délai du chemin optimal est toujours le plus faible.

Sur les 380 paires origine-destination, nous avons choisi 4 paires pour lesquelles 1) une réduction du délai peut être obtenue en choisissant un autre chemin que la route IP directe, et 2) pour lesquelles la route de délai minimale n'est pas toujours la même. Le modèle markovien pour chaque paire a été appris avec le HDP-HMM décrit dans le [chapitre 2](#).

5.4.2.1 Temps de calcul

Les paires sont décrites dans le [tableau 5.2](#). La colonne « Temps/pas » indique le temps de calcul des méthodes approximées (H : agrégation de chemin, RH : *receding horizon*) à chaque pas de temps, sur un processeur Intel Core i7-7600U @ 2.80GHz. Dans tous les cas, le temps de traitement est au maximum de 6 secondes, ce qui est inférieur au pas de temps des mesures de 2 minutes. Remarquons que pour trois des quatre scénarios, comportant plus de 2 chemins, le coût de calcul $\mathcal{O}(|S|^2)$ de la politique optimale rend impossible son calcul dans un temps raisonnable.

Paire	Chemins	États	Temps/pas (s)	
			H	RH
Paris - Santiago	4	56×10^9	0.44	6
Haifa - Santiago	4	16×10^9	0.27	0.87
Singapour - Hong Kong	3	168×10^6	0.25	0.05
Paris - Tokyo	2	60×10^3	0.10	0.1×10^{-3}

TABLE 5.2 – Scénarios et temps de traitement pour l'heuristique d'aggrégations de chemins (H) et l'approche à horizon de temps reculant (RH, horizon = 3).

5.4.2.2 Résultats

La politique myope, ainsi que les deux méthodes approximées, ont été appliquées sur les données réelles avec $\tau_{\max} = 100$, $\rho = 0.99$ et $c = 0.5$.

Les résultats obtenus pour les 4 scénarios sont présentés dans le [tableau 5.3](#). Les abréviations suivantes sont utilisées : H-S pour Haifa - Santiago, P-S pour Paris - Santiago, S-H pour Singapour - Hong Kong et P-T pour Paris - Tokyo. La colonne « Mesures moyennes » indique le nombre de chemins mesurés en moyenne par pas de temps. La colonne « Délai opt. (%) » indique le pourcentage de temps pour lequel le délai est égal au délai du chemin optimal. La colonne « Écart délai opt. (%) » indique l'écart relatif du délai par rapport au délai optimal.

À l'exception du scénario Haifa - Santiago, les deux méthodes approximées fournissent un délai plus faible ou égal à la politique myope, sans trop augmenter le nombre de mesures.

La [figure 5.4](#) présente le délai obtenu après optimisation de la mesure et du routage avec la politique à horizon de temps reculant, sur le scénario Haifa - Santiago. Dans ce cas, la politique ne mesure que 0.2 chemins sur les 4 en moyenne. Autrement dit, chaque chemin n'est mesuré que 5% du temps, et pour autant le délai obtenu est très proche du délai optimal (à 0.7%). On remarque également sur la figure que le délai par rapport au chemin direct est significativement réduit.

	Mesures moyennes			Délai opt. (%)			Écart délai opt. (%)		
	H	RH	MP	H	RH	MP	H	RH	MP
P-S	0.15	0.16	0.05	96	97	90	0.5	0.5	0.7
H-S	0.09	0.2	0.002	50	70	34	2	0.7	2.2
S-H	0.52	0.56	0.43	98	98	98	2.3	2.3	2.3
P-T	0.06	0.02	0.0	90	88	68	0.1	0.1	0.3

TABLE 5.3 – Évaluation de l'heuristique d'aggrégations de chemins (H), de l'approche à horizon de temps reculant (RH, horizon = 3) et de la politique myope (MP) sur 4 scénarios avec des mesures NLNOG.

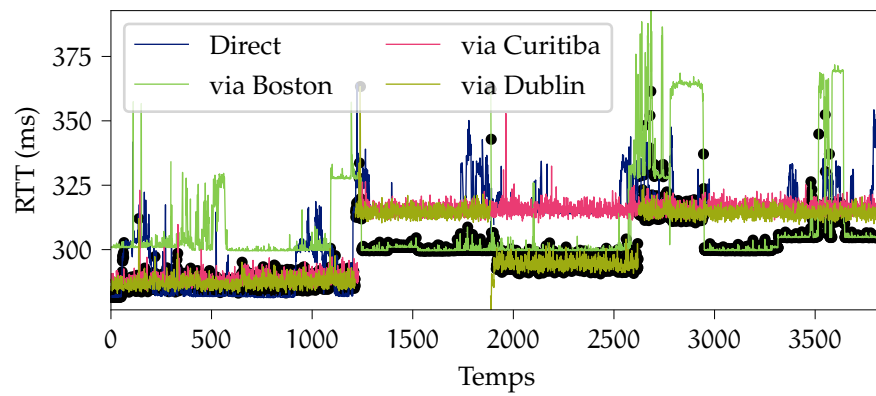


FIGURE 5.4 – Simulation de la politique à horizon de temps reculant sur un scénario à 4 chemins. En noir le délai après optimisation de la mesure et du routage.

CONCLUSION

La première contribution de cette thèse est la modélisation du délai aller-retour (RTT) sur Internet à l'aide du HDP-HMM, ou modèle de Markov caché infini. Le modèle de Markov caché est un modèle de série temporelle à temps discret et à états cachés. Le modèle suppose que les observations sont distribuées différemment selon un état caché sous-jacent, et qu'il existe une dépendance temporelle entre les états cachés successifs. Dans le cas des mesures de délai, l'état caché peut-être vu, principalement, comme une combinaison d'un chemin réseau « physique » (suite de routeurs) et d'un niveau de trafic. Un changement de chemin physique cause un changement de délai, qui résulte en un changement d'état. De la même façon, un changement dans le niveau de trafic cause un changement dans la variance du délai, qui résulte en un changement d'état. Enfin, d'autres facteurs, telles que la charge processeur des hôtes participant à la mesure, peuvent également causer des changements d'état.

Les modèles de Markov cachés ont déjà été utilisés pour modéliser la performance des réseaux dans le passé, mais leur application sur des données réelles peut-être difficile. En effet, l'estimation des paramètres d'un modèle de Markov caché nécessite de fixer en avance le nombre d'états cachés. Or, le nombre d'états est souvent inconnu a priori lorsque les données proviennent de mesures réelles. Le modèle de Markov caché infini est une version *non paramétrique* et *bayésienne* du modèle de Markov caché, qui permet d'inférer directement le nombre d'états à partir des données. De plus, nous considérons un modèle lui-même non paramétrique, le DPMM ou modèle de mélange infini, pour modéliser la distribution des observations dans un état donné. Cela nous permet d'obtenir un modèle très générique, qui fait peu d'hypothèses sur la distribution du délai dans un état, au contraire des travaux plus anciens qui supposent souvent une distribution particulière (loi normale, loi exponentielle, loi gamma, etc.).

Dans cette thèse nous utilisons le HDP-HMM pour modéliser automatiquement 16000 séries temporelles issues de mesures du délai sur Internet. Le modèle obtenu nous permet de segmenter les séries temporelles en une suite d'états, et d'étudier la dynamique du délai sur Internet. On remarque notamment que les états pour lesquelles la variance du délai est faible durent plus longtemps en moyenne que les états pour lesquelles la variance du délai est plus élevée. La segmentation obtenue nous permet également de détecter des changements anormaux du délai sur Internet. En effet, lorsqu'un changement se produit sur une infrastructure centrale d'Internet, tel qu'un point

d'échange, on observe un grand nombre de changements d'état synchronisés sur plusieurs séries temporelles.

Nous avons créé une API web publiquement accessible, et hébergée dans l'infrastructure de mesure RIPE Atlas, qui permet la segmentation à la demande de mesures de délai. Nous avons également montré que le HDP-HMM peut modéliser d'autres types de mesures réseau, tel que des mesures de congestion du projet CAIDA MANIC, ou des mesures de temps de chargement de pages web du projet Orange WebView. Enfin, nous avons exploité la capacité de prédiction de ce modèle pour réduire le nombre de mesures du délai dans les réseaux overlay.

La seconde contribution de cette thèse est la proposition d'une méthode de mesure parcimonieuse pour réduire le coût de mesure dans les réseaux overlay, ou overlays de routage. Un overlay de routage est constitué d'hôtes déployés à différents endroits d'Internet, et ces hôtes, ou nœuds, coopèrent pour transmettre le trafic de n'importe quels autres nœuds. Dans certains cas, le fait de relayer le trafic par un nœud intermédiaire, plutôt que de transmettre directement les paquets sur Internet, permet de réduire le délai de traversée du réseau. Pour ce faire, les nœuds doivent constamment mesurer le délai entre eux. Or, les nœuds d'un overlay de routage sont interconnectés entre eux dans une topologie de graphe complet, ce qui veut dire que le nombre de liens entre les nœuds grandit de façon quadratique avec le nombre de nœuds. Lorsque le nombre de nœuds devient grand, le coût de mesure peut devenir prohibitif. Nous proposons une méthode de mesure parcimonieuse qui permet de réduire grandement le nombre de mesures, tout en continuant à faire des choix de routage optimaux.

Notre méthode de mesure parcimonieuse exploite la stabilité temporelle des chemins sur Internet, ainsi que la modélisation markovienne du délai avec le HDP-HMM présentée précédemment. Nous formulons le problème comme un problème d'optimisation du délai pénalisé par le coût de mesure. Nous utilisons les processus de décision markoviens pour formuler et résoudre de façon optimale le problème pour un nombre arbitraire de chemins stochastiques. Nous fournissons également des solutions analytiques dans le cas de 1 et 2 chemins stochastiques. Lorsque le coût de calcul de la solution optimale devient prohibitif, nous fournissons deux heuristiques permettant d'obtenir une politique de mesure parcimonieuse en un temps raisonnable.

Nous avons montré à travers ces diverses applications que le HDP-HMM est un modèle robuste et flexible pour modéliser une grande variété de séries temporelles de mesures réseau. L'utilisation de ce modèle est complémentaire aux mesures de chemin sur Internet, puisqu'il permet de détecter de nombreux changements de délai invisible dans les mesures de chemin. Cependant, l'interprétation « physique » des états cachés mérite d'être approfondie.

6.1 TRAVAUX FUTURS

Cette thèse fait le lien entre des concepts statistiques plutôt théoriques, et des concepts réseau plus appliqués. Dans cette section nous détaillons quelques directions pour approfondir nos travaux, dans chacun de ces domaines.

6.1.1 Apprentissage en ligne

Dans cette thèse, l'apprentissage des modèles et des politiques est réalisé hors-ligne. Lorsque nous segmentons une série temporelle, nous devons disposer de toutes les observations (sur une semaine par exemple). Lorsque nous apprenons la politique optimale du MDP, nous devons disposer des modèles des chemins appris au préalable. Afin de permettre l'apprentissage « au fil de l'eau », deux approches sont envisageables : l'apprentissage en ligne du HDP-HMM, et l'apprentissage par renforcement de la politique de mesure.

L'apprentissage en ligne du HDP-HMM consiste à mettre à jour les paramètres du modèle à mesure que de nouvelles observations sont ajoutées, et que, potentiellement, de nouveaux états apparaissent. Il est également nécessaire d'oublier automatiquement les anciens états qui ne sont plus représentatifs des données plus récentes. Une solution à ce problème est proposée avec le HDP-HMM adaptatif en ligne (*AdOn HDP-HMM*) [12]. Il serait également possible d'étudier les méthodes d'inférence séquentielle, telles que l'échantillonneur de Gibbs particulière pour le HDP-HMM [104], bien que le problème de l'inférence en ligne ne soit pas directement traité.

L'apprentissage des politiques de décision au fil de l'eau, c'est-à-dire lorsque le modèle MDP n'est pas connu dès le départ, est traité par le domaine de l'apprentissage par renforcement [101]. L'agent commence par appliquer des décisions au hasard afin d'explorer son environnement, et de découvrir les états possibles. Notre approche MDP pourrait être généralisée grâce au processus de décision markovien partiellement observé infini (*iPOMDP*) [31], qui intègre apprentissage en ligne du HDP-HMM et apprentissage en ligne de la politique.

6.1.2 Partitionnement des modèles

Les HMMs permettent de résumer les séries temporelles à un petit nombre de paramètres interprétables. La matrice de transition donne des informations sur la durée moyenne dans chaque état, et sur les probabilités de passer d'un état à l'autre. Les lois d'observations nous renseignent sur la distribution du délai, comme par exemple, sa moyenne et sa variance. Ces paramètres peuvent être utilisés pour regrouper les états similaires à l'intérieur d'une série temporelle, et pour regrouper les séries temporelles similaires entre-elles.

Sur certaines séries temporelles, il semble possible d'identifier des « macro-états » qui regroupent des états avec de fortes probabilités de transition entre eux. Ces séries temporelles ont des matrices de transitions qui présentent des « blocs » (probabilités importantes) autour de la diagonale. Un algorithme capable d'identifier automatiquement de telles matrices, ainsi que les blocs correspondants, pourrait être utilisé pour définir ces macro-états.

Dans [94] les auteurs explorent des méthodes de partitionnement afin d'identifier des groupes de séries temporelles similaires. Le partitionnement est réalisé sur la base de caractéristiques extraites des séries temporelles : la densité spectrale de puissance, l'entropie, le nombre de changements de plus de 15 ms et quelques statistiques (mode, moyenne, écart-type ...). Les auteurs identifient deux partitions dans les données : des séries temporelles relativement stables avec des pics occasionnels, et des séries avec de grandes variations.

Une méthodologie alternative de partitionnement des séries temporelles est possible sur la base des HMMs. Il est possible de regrouper les modèles similaires entre eux. L'algorithme *VHEM-H₃M* [25] permet de réaliser un tel partitionnement. Cet algorithme est également capable de calculer le HMM « moyen » représentatif de chaque classe. Des résultats préliminaires montrent que l'on peut identifier plus de deux classes dans les mesures de délai RIPE Atlas.

6.1.3 Détection d'incidents

Finalement, l'analyse à grande échelle des mesures réseau à l'aide du HDP-HMM mérite d'être approfondie. Nous avons montré qu'il est possible d'identifier des « pics » dans la fréquence de changements d'état lorsqu'un incident se produit sur Internet. Cependant nous ne prenons pas en compte l'importance des changements, nous ne détectons pas automatiquement ces pics, et nous ne localisons pas la source de ces changements.

Prendre en compte les caractéristiques de l'état précédant un changement, et celles de l'état suivant, permettrait de pondérer les changements lors du calcul de la fréquence de changements. En effet, un changement entre un état de faible variance vers un état de plus haute variance est différent d'un changement de moyenne. De plus, cela permettrait de quantifier l'augmentation ou la diminution du délai moyen suite à un incident.

Afin de détecter automatiquement des changements significatifs dans la fréquence de changement, une méthode de détection de changement simple comme la méthode CUSUM pourrait être utilisée. Des méthodes de tomographie [67] pourraient être utilisées afin de localiser la source de ces changements. Ce problème est notamment étudié dans [92, ch. 6].

BIBLIOGRAPHIE

- [1] Emile ABEN. *Does the Internet Route Around Damage? A Case Study Using RIPE Atlas*. 17 nov. 2015. URL : <https://labs.ripe.net/Members/emileaben/does-the-internet-route-around-damage>.
- [2] Emile ABEN et Maxime MOUCHET. *What Happens When You Let Statisticians Loose on RIPE Atlas Data?* 1^{er} fév. 2019. URL : <https://labs.ripe.net/Members/emileaben/what-happens-when-you-let-statisticians-loose-at-ripe-atlas-data>.
- [3] Emile ABEN et Stephen D. STROWES. *Does The Internet Route Around Damage in 2018?* 11 avr. 2018. URL : <https://labs.ripe.net/Members/emileaben/does-the-internet-route-around-damage-in-2018>.
- [4] H. AKAIKE. « A new look at the statistical model identification ». In : *IEEE Transactions on Automatic Control* 19.6 (déc. 1974), p. 716-723. URL : <https://doi.org/10.1109/tac.1974.1100705>.
- [5] David ANDERSEN, Hari BALAKRISHNAN, Frans KAASHOEK et Robert MORRIS. « Resilient overlay networks ». In : *SIGOPS Oper. Syst. Rev.* 35.5 (oct. 2001), p. 131-145. URL : <https://doi.org/10.1145/502059.502048>.
- [6] Konstantin ANDREEV, Bruce M MAGGS, Adam MEYERSON et Ramesh K SITARAMAN. « Designing overlay multicast networks for streaming ». In : *Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures*. 2003, p. 149-158. URL : <https://doi.org/10.1145/777412.777437>.
- [7] Charles E. ANTONIAK. « Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems ». In : *The Annals of Statistics* 2.6 (nov. 1974), p. 1152-1174. URL : <https://doi.org/10.1214/aos/1176342871>.
- [8] David ARTHUR et Sergei VASSILVITSKII. « K-Means++ : The Advantages of Careful Seeding ». In : *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '07. Society for Industrial et Applied Mathematics, 2007, p. 1027-1035. URL : <https://dl.acm.org/doi/10.5555/1283383.1283494>.
- [9] Peter AUER, Nicolò CESA-BIANCHI, Yoav FREUND et Robert E. SCHAPIRE. « The Nonstochastic Multiarmed Bandit Problem ». In : *SIAM J. Comput.* 32.1 (jan. 2003), p. 48-77. URL : <https://doi.org/10.1137/s0097539701398375>.

- [10] Brice AUGUSTIN, Xavier CUVELLIER, Benjamin ORGOGOZO, Fabien VIGER, Timur FRIEDMAN, Matthieu LATAPY, Clémence MAGNIEN et Renata TEIXEIRA. « Avoiding traceroute anomalies with Paris traceroute ». In : *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*. 2006, p. 153-158. URL : <https://doi.org/10.1145/1177080.1177100>.
- [11] Brice AUGUSTIN, Timur FRIEDMAN et Renata TEIXEIRA. « Measuring load-balanced paths in the Internet ». In : *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. 2007, p. 149-160. URL : <https://doi.org/10.1145/1298306.1298329>.
- [12] Ava BARGI, Richard Yi DA XU et Massimo PICCARDI. « AdOn HDP-HMM : An Adaptive Online Model for Segmentation and Classification of Sequential Data ». In : *IEEE Transactions on Neural Networks and Learning Systems* 29.9 (sept. 2017), p. 3953-3968. URL : <https://doi.org/10.1109/TNNLS.2017.2742058>.
- [13] Matthew J BEAL, Zoubin GHAHRAMANI et Carl E RASMUSSEN. « The Infinite Hidden Markov Model ». In : *Advances in neural information processing systems*. 2002, p. 577-584. URL : <https://papers.nips.cc/paper/1956-the-infinite-hidden-markov-model.pdf>.
- [14] Salem BELHAJ et Moncef TAGINA. « Modeling and Prediction of the Internet End-to-end Delay using Recurrent Neural Networks ». In : *Journal of Networks* 4.6 (août 2009), p. 528-535. URL : <https://doi.org/10.4304/jnw.4.6.528-535>.
- [15] Richard BELLMAN. « A Markovian Decision Process ». In : *Indiana Univ. Math. J.* 6 (4 1957), p. 679-684. URL : <https://doi.org/10.1512/iumj.1957.6.56038>.
- [16] Mathieu BESANÇON, David ANTHOFF, Alex ARSLAN, Simon BYRNE, Dahua LIN, Theodore PAPAMARKOU et John PEARSON. « Distributions.jl : Definition and Modeling of Probability Distributions in the JuliaStats Ecosystem ». In : *arXiv e-prints*, arXiv :1907.08611 (juill. 2019), arXiv :1907.08611. arXiv : [1907.08611 \[stat.CO\]](https://arxiv.org/abs/1907.08611).
- [17] Christopher M. BISHOP. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg : Springer-Verlag, 2006. URL : <https://dl.acm.org/doi/book/10.5555/1162264>.
- [18] David BLACKWELL et James B. MACQUEEN. « Ferguson Distributions Via Polya Urn Schemes ». In : *The Annals of Statistics* 1.2 (mars 1973), p. 353-355. URL : <https://doi.org/10.1214/aos/1176342372>.
- [19] Mohamed BOUCADAIR et Christian JACQUENET. *Emerging Automation Techniques for the Future Internet*. IGI Global, 2019. URL : <https://doi.org/10.4018/978-1-5225-7146-9>.

- [20] Raouf BOUTABA, Mohammad A. SALAHUDDIN, Noura LIMAM, Sara AYOUBI, Nashid SHAHRIAR, Felipe ESTRADA-SOLANO et Oscar M. CAICEDO. « A comprehensive survey on machine learning for networking : Evolution, applications and research opportunities ». In : *Journal of Internet Services and Applications* 9.1 (juin 2018), p. 1-99. URL : <https://doi.org/10.1186/s13174-018-0087-2>.
- [21] O. BRUN, L. WANG et E. GELENBE. « Big Data for Autonomic Intercontinental Overlays ». In : *IEEE Journal on Selected Areas in Communications* 34.3 (2016), p. 575-583. URL : <https://doi.org/10.1109/jsac.2016.2525518>.
- [22] Olivier BRUN, Hassan HASSAN et Josselin VALLET. « Scalable, self-healing, and self-optimizing routing overlays ». In : *2016 IFIP Networking Conference (IFIP Networking) and Workshops*. IEEE. 2016, p. 64-72. URL : <https://doi.org/10.1109/ifipnetworking.2016.7497209>.
- [23] Christopher Thraves CARO, Josu DONCEL et Olivier BRUN. « Optimal path discovery problem with homogeneous knowledge ». In : *Theory of Computing Systems* 64.2 (2020), p. 227-250. URL : <https://doi.org/10.1007/s00224-019-09928-w>.
- [24] Brent CHUN, David CULLER, Timothy ROSCOE, Andy BAVIER, Larry PETERSON, Mike WAWRZONIAK et Mic BOWMAN. « Planetlab : an overlay testbed for broad-coverage services ». In : *ACM SIGCOMM Computer Communication Review* 33.3 (2003), p. 3-12. URL : <https://doi.org/10.1145/956993.956995>.
- [25] Emanuele COVIELLO, Antoni B. CHAN et Gert R. G. LANCKRIET. « Clustering hidden Markov models with variational HEM ». In : *Journal of Machine Learning Research* 15.1 (jan. 2014), p. 697-747. URL : <https://dl.acm.org/doi/abs/10.5555/2627435.2627457>.
- [26] Alberto DAINOTTI, Antonio PESCAPÉ, Pierluigi Salvo ROSSI, Francesco PALMIERI et Giorgio VENTRE. « Internet traffic modeling by means of Hidden Markov Models ». In : *Computer Networks* 52.14 (oct. 2008), p. 2645-2662. URL : <https://doi.org/10.1016/j.comnet.2008.05.004>.
- [27] Alun DAVIES. *Announcing RIPE Atlas VM Anchors*. 8 nov. 2018. URL : https://labs.ripe.net/Members/alun_davies/announcing-ripe-atlas-vm-anchors.
- [28] A. P. DEMPSTER, N. M. LAIRD et D. B. RUBIN. « Maximum Likelihood from Incomplete Data Via the EM Algorithm ». In : *Journal of the Royal Statistical Society : Series B (Methodological)* 39.1 (sept. 1977), p. 1-22. URL : <https://doi.org/10.1111/j.2517-6161.1977.tb01600.x>.

- [29] Amogh DHAMDHERE, David D. CLARK, Alexander GAMERO-GARRIDO, Matthew LUCKIE, Ricky K. P. MOK, Gautam AKIWATE, Kabir GOGIA, Vaibhav BAJPAI, Alex C. SNOEREN et Kimberly CLAFFY. « Inferring Persistent Interdomain Congestion ». In : *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication. SIGCOMM '18*. Association for Computing Machinery, 2018, p. 1-15. URL : <https://doi.org/10.1145/3230543.3230549>.
- [30] Ai DONG, Zhijiang DU et Zhiyuan YAN. « Round Trip Time Prediction Using Recurrent Neural Networks With Minimal Gated Unit ». In : *IEEE Communications Letters* 23.4 (avr. 2019), p. 584-587. URL : <https://doi.org/10.1109/lcomm.2019.2899603>.
- [31] Finale DOSHI-VELEZ. « The Infinite Partially Observable Markov Decision Process ». In : *Advances in Neural Information Processing Systems* 22. Curran Associates, Inc., 2009, p. 477-485. URL : <http://papers.nips.cc/paper/3780-the-infinite-partially-observable-markov-decision-process.pdf>.
- [32] Constantine DOVROLIS, Krishna GUMMADI, Aleksandar KUZMANOVIC et Sascha D MEINRATH. « Measurement lab : Overview and an invitation to the research community ». In : *ACM SIGCOMM Computer Communication Review* 40.3 (2010), p. 53-56. URL : <https://doi.org/10.1145/1823844.1823853>.
- [33] Stuart DREYFUS. « Richard Bellman on the Birth of Dynamic Programming ». In : *Operations Research* (fév. 2002). URL : <https://doi.org/10.1287/opre.50.1.48.17791>.
- [34] Nick FEAMSTER, Hari BALAKRISHNAN, Jennifer REXFORD, Aman SHAIKH et Jacobus van der MERWE. « The Case for Separating Routing from Routers ». In : *Proceedings of the ACM SIGCOMM Workshop on Future Directions in Network Architecture. FDNA '04*. Association for Computing Machinery, 2004, p. 5-12. URL : <https://doi.org/10.1145/1016707.1016709>.
- [35] Thomas S. FERGUSON. « A Bayesian Analysis of Some Nonparametric Problems ». In : *The Annals of Statistics* 1.2 (mars 1973), p. 209-230. URL : <https://doi.org/10.1214/aos/1176342360>.
- [36] R. FONTUGNE, J. MAZEL et K. FUKUDA. « An empirical mixture model for large-scale RTT measurements ». In : *2015 IEEE Conference on Computer Communications (INFOCOM)*. 2015, p. 2470-2478. URL : <https://doi.org/10.1109/INFOCOM.2015.7218636>.
- [37] Romain FONTUGNE, Cristel PELSSER, Emile ABEN et Randy BUSH. « Pinpointing Delay and Forwarding Anomalies Using Large-Scale Traceroute Measurements ». In : *Proceedings of the 2017 Internet Measurement Conference. IMC '17*. Association for

- Computing Machinery, 2017, p. 15-28. URL : <https://doi.org/10.1145/3131365.3131384>.
- [38] Emily B. Fox et al. *Software*. URL : <https://homes.cs.washington.edu/~ebfox/software>.
- [39] Emily B. FOX, Erik B. SUDDERTH, Michael I. JORDAN et Alan S. WILLSKY. « A sticky HDP-HMM with application to speaker diarization ». In : *The Annals of Applied Statistics* 5.2A (juin 2011), p. 1020-1056. URL : <https://doi.org/10.1214/10-aos395>.
- [40] Samuel J. GERSHMAN et David M. BLEI. « A tutorial on Bayesian nonparametric models ». In : *Journal of Mathematical Psychology* 56.1 (fév. 2012), p. 1-12. URL : <https://doi.org/10.1016/j.jmp.2011.08.004>.
- [41] Jim GETTYS et Kathleen NICHOLS. « Bufferbloat : Dark buffers in the Internet ». In : *Queue* 9.11 (2011), p. 40-54. URL : <https://doi.org/10.1145/2063166.2071893>.
- [42] Vasileios GIOTAS, Christoph DIETZEL, Georgios SMARAGDAKIS, Anja FELDMANN, Arthur BERGER et Emile ABEN. « Detecting Peering Infrastructure Outages in the Wild ». In : *Proceedings of the Conference of the ACM Special Interest Group on Data Communication. SIGCOMM '17*. Association for Computing Machinery, 2017, p. 446-459. URL : <https://doi.org/10.1145/3098822.3098855>.
- [43] *Global RIPE Atlas Network Coverage*. URL : <https://atlas.ripe.net/results/maps/network-coverage> (visité le 22/05/2020).
- [44] Ian GOODFELLOW, Yoshua BENGIO et Aaron COURVILLE. *Deep Learning*. The MIT Press, 2016. URL : <https://dl.acm.org/doi/book/10.5555/3086952>.
- [45] Graham GOODWIN, María M SERON et José A DE DONÁ. *Constrained control and estimation : an optimisation approach*. Springer Science & Business Media, 2006. URL : <https://doi.org/10.1007/b138145>.
- [46] Kaylea HAYNES, Paul FEARNHEAD et Idris A. ECKLEY. « A computationally efficient nonparametric approach for changepoint detection ». In : *Statistics and Computing* 27.5 (juill. 2016), p. 1293-1305. URL : <https://doi.org/10.1007/s11222-016-9687-5>.
- [47] J. A. HERNANDEZ et I. W. PHILLIPS. « Weibull mixture model to characterise end-to-end Internet delay at coarse time-scales ». In : *IEE Proceedings - Communications* 153.2 (avr. 2006), p. 295. URL : <https://doi.org/10.1049/ip-com:20050335>.
- [48] Thomas HOLTERBACH, Emile ABEN, Cristel PELSSER, Randy BUSH et Laurent VANBEVER. « Measurement Vantage Point Selection Using A Similarity Metric ». In : *ANRW '17 : Proceedings of the Applied Networking Research Workshop* (2020), p. 1-3. URL : <https://doi.org/10.1145/3106328.3106334>.

- [49] Shun-Yun HU et Guan-Ming LIAO. « Scalable peer-to-peer networked virtual environment ». In : *Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games* (août 2004), p. 129-133. URL : <https://doi.org/10.1145/1016540.1016552>.
- [50] Young HYUN. « Archipelago measurement infrastructure ». In : *Proceedings of the 7th CAIDA-WIDE Workshop*. 2006. URL : https://www.caida.org/publications/presentations/2006/young_wide0611_ark/young_wide0611_ark.pdf.
- [51] H ISHWARAN. « Markov chain Monte Carlo in approximate Dirichlet and beta two-parameter process hierarchical models ». In : *Biometrika* 87.2 (juin 2000), p. 371-390. URL : <https://doi.org/10.1093/biomet/87.2.371>.
- [52] Matthew J. JOHNSON et al. *pyhsmm*. URL : <https://github.com/mattjj/pyhsmm>.
- [53] Matthew J. JOHNSON et Alan S. WILLSKY. « Bayesian Nonparametric Hidden Semi-Markov Models ». In : *Journal of Machine Learning Research* 14.2 (2013), p. 673-701. URL : <http://www.jmlr.org/papers/v14/johnson13a.html>.
- [54] E. KAMRANI, H. R. MOMENI et A. R. SHARAFAT. « Modeling Internet delay dynamics for teleoperation ». In : *Proceedings of 2005 IEEE Conference on Control Applications, 2005. CCA 2005*. 2005, p. 1528-1533. URL : <https://doi.org/10.1109/cca.2005.1507349>.
- [55] Ken KEYS. « Internet-scale IP alias resolution techniques ». In : *ACM SIGCOMM Computer Communication Review* 40.1 (jan. 2010), p. 50-55. URL : <https://doi.org/10.1145/1672308.1672318>.
- [56] Ken KEYS, Young HYUN, Matthew LUCKIE et Kimberly CLAFFY. « Internet-Scale IPv4 Alias Resolution With MIDAR ». In : *IEEE/ACM Transactions on Networking* 21.2 (mai 2012), p. 383-399. URL : <https://doi.org/10.1109/tnet.2012.2198887>.
- [57] R. KILLICK, P. FEARNHEAD et I. A. ECKLEY. « Optimal Detection of Changepoints With a Linear Computational Cost ». In : *J. Am. Stat. Assoc.* 107.500 (déc. 2012), p. 1590-1598. URL : <https://doi.org/10.1080/01621459.2012.737745>.
- [58] Rebecca KILLICK et Idris ECKLEY. « changepoint : An R Package for Changepoint Analysis ». In : *Journal of Statistical Software, Articles* 58.3 (2014), p. 1-19. URL : <https://doi.org/10.18637/jss.v058.i03>.
- [59] Bikash KOLEY. « The Zero Touch Network ». In : *Google Research* (2016). URL : <https://research.google/pubs/pub45687>.

- [60] Diego KREUTZ, Fernando M. V. RAMOS, Paulo ESTEVES VERISSIMO, Christian ESTEVE ROTHENBERG, Siamak AZODOLMOLKY et Steve UHLIG. « Software-Defined Networking : A Comprehensive Survey ». In : *Proceedings of the IEEE* 103.1 (jan. 2015), p. 14-76. URL : <https://doi.org/10.1109/jproc.2014.2371999>.
- [61] Vikram KRISHNAMURTHY. *Partially observed Markov decision processes*. Cambridge University Press, 2016. URL : <https://doi.org/10.1017/cbo9781316471104>.
- [62] Farah LAYOUNI, Brice AUGUSTIN, Timur FRIEDMAN et Renata TEIXEIRA. « Origine des étoiles dans traceroute ». In : *Colloque Francophone sur l'Ingénierie des Protocoles (CFIP 2008)*. Mars 2008. URL : <https://hal.archives-ouvertes.fr/hal-00243154>.
- [63] Dahua LIN et al. *ConjugatePriors.jl*. URL : <https://github.com/JuliaStats/ConjugatePriors.jl>.
- [64] Matthew LUCKIE, Amogh DHAMDHERE, David CLARK, Bradley HUFFAKER et Kimberly CLAFFY. « Challenges in Inferring Internet Interdomain Congestion ». In : *Proceedings of the 2014 Conference on Internet Measurement Conference*. IMC '14. Association for Computing Machinery, 2014, p. 15-22. URL : <https://doi.org/10.1145/2663716.2663741>.
- [65] *MANIC : Measurement and ANalysis of Internet Congestion*. URL : <https://manic.caida.org/> (visité le 28/05/2020).
- [66] Paolo MEDAGLIANI, Stefano PARIS, Jérémie LEGUAY, Lorenzo MAGGI, Chuangsong XUE et Haojun ZHOU. « Overlay routing for fast video transfers in CDN ». In : *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE. 2017, p. 531-536. URL : <https://doi.org/10.23919/inm.2017.7987323>.
- [67] Abia MOLOISANE, Ivan GANCHEV et Máirtín O'DROMA. *Internet Tomography : An Introduction to Concepts, Techniques, Tools and Applications*. Cambridge Scholars Publishing, 2014. URL : <https://dl.acm.org/doi/book/10.5555/2523330>.
- [68] Maxime MOUCHET. « Integrate Julia and Javascript using Node.js extensions ». JuliaCon 2020. En ligne, juill. 2020. URL : <https://live.juliacon.org/talk/Q88P8U>.
- [69] Maxime MOUCHET, Emile ABEN et Jasper den HERTOEG. *An API for Summarising Events in RIPE Atlas RTT Time Series*. 14 mars 2019. URL : https://labs.ripe.net/Members/maxime_mouchet/api-for-summarising-events-in-ripe-atlas-rtt-time-series.
- [70] Maxime MOUCHET, Sandrine VATON et Thierry CHONAVEL. *Statistical Characterisation of RTT Series*. 6 fév. 2019. URL : https://labs.ripe.net/Members/maxime_mouchet/statistical-characterisation-of-rtt-series.

- [71] Akihiro NAKAO, Larry PETERSON et Andy BAVIER. « Scalable routing overlay networks ». In : *SIGOPS Oper. Syst. Rev.* 40.1 (jan. 2006), p. 49-61. URL : <https://doi.org/10.1145/1113361.1113372>.
- [72] Radford M. NEAL. « Markov Chain Sampling Methods for Dirichlet Process Mixture Models ». In : *Journal of Computational and Graphical Statistics* 9.2 (juin 2000), p. 249-265. URL : <https://doi.org/10.1080/10618600.2000.10474879>.
- [73] NLNOG RING. URL : <https://ring.nlnog.net> (visité le 23/05/2020).
- [74] Orange Web View. URL : <https://webview.orange.com/> (visité le 01/06/2020).
- [75] Konstantinos PANOUSIS, Sotirios CHATZIS et Sergios THEODORIDIS. « Nonparametric Bayesian Deep Networks with Local Competition ». In : *Proceedings of the 36th International Conference on Machine Learning*. Sous la dir. de Kamalika CHAUDHURI et Ruslan SALAKHUTDINOV. T. 97. Proceedings of Machine Learning Research. PMLR, juin 2019, p. 4980-4988. URL : <http://proceedings.mlr.press/v97/panousis19a.html>.
- [76] Cristel PELSSER, Luca CITTADINI, Stefano VISSICCHIO et Randy BUSH. « From Paris to Tokyo : On the suitability of ping to measure latency ». In : *Proceedings of the 2013 conference on Internet measurement conference*. 2013, p. 427-432. URL : <https://doi.org/10.1145/2504730.2504765>.
- [77] L. R. RABINER. « A tutorial on hidden Markov models and selected applications in speech recognition ». In : *Proceedings of the IEEE* 77.2 (fév. 1989), p. 257-286. URL : <https://doi.org/10.1109/5.18626>.
- [78] Anurag RAI, Rahul SINGH et Eytan MODIANO. « A distributed algorithm for throughput optimal routing in overlay networks ». In : *2019 IFIP Networking Conference (IFIP Networking)*. IEEE, 2019, p. 1-9. URL : <https://doi.org/10.23919/ifipnetworking.2019.8816834>.
- [79] Carl Edward RASMUSSEN. « The Infinite Gaussian Mixture Model ». In : *Advances in Neural Information Processing Systems* 12. MIT Press, 2000, p. 554-560. URL : <http://papers.nips.cc/paper/1745-the-infinite-gaussian-mixture-model.pdf>.
- [80] RIPE Routing Information Service (RIS). URL : <https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris> (visité le 24/07/2020).
- [81] Christian P. ROBERT et George CASELLA. *Monte Carlo Statistical Methods*. Springer New York, 2004. URL : <https://doi.org/10.1007/978-1-4757-4145-2>.

- [82] *Routeviews – University of Oregon Route Views Project*. URL : <http://www.routeviews.org/routeviews> (visité le 24/07/2020).
- [83] Tobias RYDÉN. « EM versus Markov chain Monte Carlo for estimation of hidden Markov models : A computational perspective ». In : *Bayesian Analysis* 3.4 (déc. 2008), p. 659-688. URL : <https://doi.org/10.1214/08-ba326>.
- [84] Kavé SALAMATIAN et Sandrine VATON. « Hidden Markov modeling for network communication channels ». In : *ACM SIGMETRICS Performance Evaluation Review* 29.1 (juin 2001), p. 92-101. URL : <https://doi.org/10.1145/384268.378439>.
- [85] Y. SATO, S. ATA, I. OKA et C. FUJIWARA. *Using mixed distribution for modeling end-to-end delay characteristics*. Rapp. tech. Graduate School of Engineering, Osaka City University, et Faculty of Modern Management Information, Osaka Seikei University., jan. 2005.
- [86] S. SAVAGE et al. « Detour : informed Internet routing and transport ». In : *IEEE Micro* 19.1 (jan. 1999), p. 50-59. URL : <https://doi.org/10.1109/40.748796>.
- [87] Antoine SAVERIMOUTOU. « Future Internet Metrology : Characterization, Quantification and Prediction of Web Browsing Quality ». Thèse de doct. IMT Atlantique, 2020.
- [88] Antoine SAVERIMOUTOU, Bertrand MATHIEU et Sandrine VATON. « Web View : A Measurement Platform for Depicting Web Browsing Performance and Delivery ». In : *IEEE Commun. Mag.* 58.3 (mars 2020), p. 33-39. URL : <https://doi.org/10.1109/mcom.001.1900178>.
- [89] Gideon SCHWARZ. « Estimating the Dimension of a Model ». In : *The Annals of Statistics* 6.2 (mars 1978), p. 461-464. URL : <https://doi.org/10.1214/aos/1176344136>.
- [90] Steven L. SCOTT. « Bayesian Methods for Hidden Markov Models : Recursive Computing in the 21st Century ». In : *J. Am. Stat. Assoc.* 97.457 (mars 2002), p. 337-351. URL : <https://doi.org/10.1198/016214502753479464>.
- [91] Jayaram SETHURAMAN. « A Constructive Definition of Dirichlet Priors ». In : *Statistica Sinica* 4.2 (juill. 1994), p. 639-650. URL : <http://www.jstor.org/stable/24305538>.
- [92] Wenqin SHAO. « Measurement-based inter-domain traffic engineering ». Thèse de doct. Télécom ParisTech, 2017.
- [93] Wenqin SHAO. *rtt*. URL : <https://github.com/WenqinSHAO/rtt>.

- [94] Wenqin SHAO, Jean-Louis ROUGIER, Francois DEVIENNE et Mateusz VISTE. « Improve round-trip time measurement quality via clustering in inter-domain traffic engineering ». In : *Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP*. Avr. 2016. URL : <https://doi.org/10.1109/noms.2016.7502970>.
- [95] Wenqin SHAO, Jean-Louis ROUGIER, François DEVIENNE et Mateusz VISTE. « Missing measurements on RIPE Atlas ». In : *ArXiv e-prints* (jan. 2017). eprint : 1701.00938. URL : <https://arxiv.org/abs/1701.00938v1>.
- [96] Wenqin SHAO, Jean-Louis ROUGIER, Antoine PARIS, François DEVIENNE et Mateusz VISTE. « One-to-One Matching of RTT and Path Changes ». In : *2017 29th International Teletraffic Congress (ITC 29)* 1 (sept. 2017), p. 196-204. URL : <https://doi.org/10.23919/ITC.2017.8064356>.
- [97] Ramesh K SITARAMAN, Mangesh KASBEKAR, Woody LICHTENSTEIN et Manish JAIN. « Overlay networks : An akamai perspective ». In : *Advanced Content Delivery, Streaming, and Cloud Services* 51.4 (2014), p. 305-328. URL : <https://doi.org/10.1002/9781118909690.ch16>.
- [98] RIPE NCC STAFF. « RIPE Atlas : A global Internet measurement network ». In : *Internet Protocol Journal* 18.3 (2015). URL : <https://ipj.dreamhosters.com/wp-content/uploads/issues/2015/ipj18-3.pdf>.
- [99] Robert STONE. « CenterTrack : An IP overlay network for tracking DoS floods. » In : *USENIX Security Symposium*. T. 21. 2000, p. 114. URL : https://www.usenix.org/legacy/events/sec2000/full_papers/stone/stone.pdf.
- [100] Stephen D. STROWES. *Atlas and Ark Data in Google BigQuery*. 17 oct. 2019. URL : <https://ripe79.ripe.net/archives/video/268>.
- [101] Richard S. SUTTON et Andrew G. BARTO. *Reinforcement Learning : An Introduction*. A Bradford Book, 2018. URL : <https://doi.org/10.1007/978-1-4615-3618-5>.
- [102] Yee Whye TEH, Michael I JORDAN, Matthew J BEAL et David M BLEI. « Hierarchical Dirichlet Processes ». In : *Journal of the American Statistical Association* 101.476 (déc. 2006), p. 1566-1581. URL : <https://doi.org/10.1198/016214506000000302>.
- [103] Ke TRAN, Yonatan BISK, Ashish VASWANI, Daniel MARCU et Kevin KNIGHT. « Unsupervised Neural Hidden Markov Models ». In : *ArXiv e-prints* (sept. 2016). eprint : 1609.09007. URL : <https://arxiv.org/abs/1609.09007v1>.

- [104] Nilesh TRIPURANENI, Shixiang (Shane) GU, Hong GE et Zoubin GHAHRAMANI. « Particle Gibbs for Infinite Hidden Markov Models ». In : *Advances in Neural Information Processing Systems* 28. Sous la dir. de C. CORTES, N. D. LAWRENCE, D. D. LEE, M. SUGIYAMA et R. GARNETT. Curran Associates, Inc., 2015, p. 2395-2403. URL : <http://papers.nips.cc/paper/5968-particle-gibbs-for-infinite-hidden-markov-models.pdf>.
- [105] Jurgen VAN GAEL, Yunus SAATCI, Yee Whye TEH et Zoubin GHAHRAMANI. « Beam sampling for the Infinite Hidden Markov Model ». In : *Proceedings of the 25th international conference on Machine learning - ICML '08*. ICML '08. ACM Press, 2008, p. 1088-1095. URL : <https://doi.org/10.1145/1390156.1390293>.
- [106] Sandrine VATON, Olivier BRUN, Maxime MOUCHET, Pablo BELZARENA, Isabel AMIGO, Balakrishna PRABHU et Thierry CHONAVEL. « Joint Minimization of Monitoring Cost and Delay in Overlay Networks : Optimal Policies with a Markovian Approach ». In : *Journal of Network and Systems Management* 27.1 (jan. 2019), p. 188-232. URL : <https://hal.archives-ouvertes.fr/hal-01857738>.
- [107] Kevin VERMEULEN, Stephen D. STROWES, Olivier FOURMAUX et Timur FRIEDMAN. « Multilevel MDA-Lite Paris Traceroute ». In : *Proceedings of the Internet Measurement Conference 2018*. IMC '18. Association for Computing Machinery, 2018, p. 29-42. URL : <https://doi.org/10.1145/3278532.3278536>.
- [108] Ju WANG, Linyuan LU et Andrew A CHIEN. « Tolerating denial-of-service attacks using overlay networks : impact of topology ». In : *Proceedings of the 2003 ACM workshop on Survivable and self-regenerative systems : in association with 10th ACM Conference on Computer and Communications Security*. 2003, p. 43-52. URL : <https://doi.org/10.1145/1036921.1036926>.
- [109] Wei WEI, Bing WANG et Don TOWSLEY. « Continuous-time hidden Markov models for network performance evaluation ». In : *Performance Evaluation* 49.1-4 (sept. 2002), p. 129-146. URL : [https://doi.org/10.1016/s0166-5316\(02\)00122-0](https://doi.org/10.1016/s0166-5316(02)00122-0).
- [110] M. YANG, J. RU, X. R. LI, H. CHEN et A. BASHI. « Predicting Internet End-to-End Delay : A Multiple-Model Approach ». In : *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*. 2006, p. 1-5. URL : <https://doi.org/10.1109/INFOCOM.2006.345>.
- [111] Nancy R. ZHANG et David O. SIEGMUND. « A Modified Bayes Information Criterion with Applications to the Analysis of Comparative Genomic Hybridization Data ». In : *Biometrics* 63.1 (oct. 2006), p. 22-32. URL : <https://doi.org/10.1111/j.1541-0420.2006.00662.x>.

- [112] Wei ZHANG et Jingsha HE. « Modeling End-to-End Delay Using Pareto Distribution ». In : *Second International Conference on Internet Monitoring and Protection (ICIMP 2007)* (juill. 2007), p. 21.
URL : <https://ieeexplore.ieee.org/abstract/document/4271767>.

PUBLICATIONS

ARTICLES DANS UNE REVUE

- [1] Maxime MOUCHET, Sandrine VATON, Thierry CHONAVEL, Emile ABEN et Jasper DEN HERTOOG. « Large-Scale Characterization and Segmentation of Internet Path Delays with Infinite HMMs ». In : *IEEE Access* 8 (jan. 2020), p. 16771-16784. URL : <https://hal.archives-ouvertes.fr/hal-02326926>.
- [2] Sandrine VATON, Olivier BRUN, Maxime MOUCHET, Pablo BELZARENA, Isabel AMIGO, Balakrishna PRABHU et Thierry CHONAVEL. « Joint Minimization of Monitoring Cost and Delay in Overlay Networks : Optimal Policies with a Markovian Approach ». In : *Journal of Network and Systems Management* 27.1 (jan. 2019), p. 188-232. URL : <https://hal.archives-ouvertes.fr/hal-01857738>.

COMMUNICATIONS DANS UN CONGRÈS AVEC ACTE

- [3] Maxime MOUCHET, Martin RANDALL, Marine SÉGNERÉ, Isabel AMIGO, Pablo BELZARENA, Olivier BRUN, Balakrishna PRABHU et Sandrine VATON. « Scalable Monitoring Heuristics for Improving Network Latency ». In : *IEEE/IFIP Network Operations and Management Symposium (IEEE/IFIP NOMS 2020)*. Budapest, Hongrie, avr. 2020. URL : <https://hal.laas.fr/hal-02413636>.
- [4] Maxime MOUCHET, Sandrine VATON et Thierry CHONAVEL. « Statistical Characterization of Round-Trip Times with Nonparametric Hidden Markov Models ». In : *IFIP/IEEE IM 2019 Workshop : 4th International Workshop on Analytics for Network and Service Management (AnNet 2019)*. Washington DC, États-Unis, avr. 2019. URL : <https://hal.archives-ouvertes.fr/hal-02010099>.
- [5] Maxime MOUCHET, Sandrine VATON et Thierry CHONAVEL. « Poster Abstract : A flexible infinite HMM model for accurate characterization and segmentation of RTT timeseries ». In : *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. Paris, France, avr. 2019, p. 1055-1056. URL : <https://hal.archives-ouvertes.fr/hal-02300968>.
- [6] Maxime MOUCHET, Sandrine VATON, Olivier BRUN, Pablo BELZARENA, Isabel AMIGO et Balakrishna PRABHU. « Optimisation conjointe de la métrologie active et du routage : une approche mar-

kovienne ». In : *Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication*. Roscoff, France, mai 2018. URL : <https://hal.archives-ouvertes.fr/hal-01785870>.

COMMUNICATIONS DANS UN CONGRÈS SANS ACTE

- [7] Maxime MOUCHET. « Integrate Julia and Javascript using Node.js extensions ». JuliaCon 2020. En ligne, juill. 2020. URL : <https://live.juliacon.org/talk/Q88P8U>.
- [8] Maxime MOUCHET, Thierry CHONAVEL et Sandrine VATON. « Statistical models for RTTs ». Neutral Interconnect Days 2019. La Haye, Pays-Bas, sept. 2019.
- [9] Maxime MOUCHET, Thierry CHONAVEL et Sandrine VATON. « Learning network states from RTT measurements ». RIPE 77. Amsterdam, Pays-Bas, oct. 2018. URL : <https://ripe77.ripe.net/archives/video/2250/>.

ARTICLES DE BLOG

- [10] Maxime MOUCHET, Emile ABEN et Jasper den HERTOEG. *An API for Summarising Events in RIPE Atlas RTT Time Series*. 14 mars 2019. URL : https://labs.ripe.net/Members/maxime_mouchet/api-for-summarising-events-in-ripe-atlas-rtt-time-series.
- [11] Maxime MOUCHET, Sandrine VATON et Thierry CHONAVEL. *Statistical Characterisation of RTT Series*. 6 fév. 2019. URL : https://labs.ripe.net/Members/maxime_mouchet/statistical-characterisation-of-rtt-series.
- [12] Emile ABEN et Maxime MOUCHET. *What Happens When You Let Statisticians Loose on RIPE Atlas Data?* 1^{er} fév. 2019. URL : <https://labs.ripe.net/Members/emileaben/what-happens-when-you-let-statisticians-loose-at-ripe-atlas-data>.

Titre : Modélisation Robuste du Délai Internet et Schémas de Mesure Intelligents pour l'Automatisation des Réseaux Overlay

Mots clés : Métrologie Parcimonieuse, Réseaux Informatiques, Mesure d'Internet, Modèles Bayésiens Non-Paramétriques, Processus de Décision Markoviens

Résumé : La mesure de la performance des réseaux informatiques est un élément crucial pour l'administration, la sécurisation et l'optimisation des réseaux. Des changements inattendus de la performance peuvent indiquer des incidents matériels, ou des attaques informatiques. De plus, une bonne connaissance de la performance des différents éléments du réseau permet de choisir intelligemment le chemin entre deux hôtes afin d'optimiser la performance de bout en bout. Dans cette thèse nous considérons la mesure du délai aller-retour sur Internet. Nous proposons d'abord un modèle statistique pour le délai sur Internet, basé sur les modèles de Markov cachés infinis, qui permet d'identifier les changements significatifs de délai et de prédire ses valeurs futures. Ce modèle est testé sur plusieurs dizaines de milliers de séries temporelles provenant de l'infrastructure de mesure d'Internet RIPE Atlas, et sur un jeu de données labélisé d'instantané de changements. Nous montrons que ce modèle permet de détecter des incidents sur des infrastructures centrales d'Internet, tels que les points d'échanges. De plus, le modèle est déployé au sein de l'infrastructure de mesure RIPE Atlas, où une API HTTP publique permet la segmentation à la demande de séries de mesures de délai.

Notre seconde contribution est la conception d'un schéma de mesure intelligent pour la mesure du délai dans les réseaux overlays. Les réseaux overlays sont un type particulier de réseau où les nœuds établissent des liens virtuels entre eux, dans une topologie de graphe complet. Chaque nœud peut relayer les paquets d'un autre nœud, permettant ainsi d'influencer le chemin physique emprunté par les paquets. L'objectif est alors de choisir à tout instant le chemin virtuel (suite de nœuds relais) qui permet de minimiser une certaine métrique (par exemple, le délai aller-retour). Nous proposons une méthode qui permet d'atteindre cet objectif sans avoir besoin de mesurer en permanence la performance entre chacun des liens virtuels. Notre méthode exploite les modèles de Markov cachés infinis et les processus de décision Markoviens afin de fournir une politique de mesure qui effectue le compromis optimal entre nombre de mesures et performance du routage. Nous montrons sur une topologie de 30 nœuds que notre méthode permet d'obtenir un routage quasi optimal en ne mesurant que 10% du temps. Finalement, nous proposons des heuristiques permettant d'obtenir des politiques de mesures parcimonieuses en un temps raisonnable.

Title: Robust Modelling of Internet Delay and Smart Monitoring Schemes for the Automation of Overlay Networks

Keywords: Parsimonious Monitoring, Computer Networks, Internet Measurement, Nonparametric Bayesian Models, Markov Decision Processes

Abstract: It is crucial to computer networks administration, security and optimization to monitor the performance of the network. Unexpected changes in performance can indicate hardware problems, or malicious attacks. In addition, a good understanding of the performance of the network components allows for intelligent path selection between the hosts to optimize end-to-end performance. In this thesis we consider the measurement of the round-trip time (RTT) on the Internet. We first propose a statistical model for the delay on the Internet, based on infinite hidden Markov models, which can detect significant changes in the delay and predict its future values. This model is validated on tens of thousands of time series from the RIPE Atlas Internet measurement infrastructure, and on a labeled change point dataset. We show that this can detect incidents on key Internet infrastructures, such as exchange points (IXP). Furthermore, the model is deployed within the RIPE Atlas measurement infrastructure, and a public HTTP API enables on-demand segmentation of delay measurements.

Our second contribution is the design of an intelligent measurement scheme for the monitoring of delay in overlay networks. Overlay networks are a specific type of network where nodes establish virtual links with each other, in a complete graph topology. Each node can relay packets from the other nodes, thereby influencing the physical path taken by the packets. The objective is then to choose at all times the virtual path (series of relay nodes) which minimize a certain metric (for example, the round-trip delay). We provide a method that achieves this goal without the need to continuously measure the performance between each of the virtual links. Our method exploits the infinite hidden Markov models and Markovian decision processes to provide a measurement policy that makes the optimal trade-off between number of measurements and routing performance. We show on a topology of 30 nodes that our method allows to obtain an almost optimal routing by measuring only 10% of the time. Finally, we propose heuristics allowing to obtain sparse measurement policies in a reasonable time.