Détection et estimation d'anomalies dans un réseau de communication

Sandy Rahme

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par
l'Université Paul Sabatier

Discipline ou spécialité :
Systèmes Automatiques

Présentée et soutenue par
Sandy RAHME

Le *mercredi 16 novembre 2011*

Titre :

Détection et estimation d'anomalies dans un réseau de communication

---

JURY

M. Jean-Louis Calvet	*Université Toulouse III - Paul Sabatier*	*Président*
M. Jean-Pierre Thomesse	*INP de Lorraine*	*Rapporteur*
M. Jamal Daafouz	*INP de Lorraine*	*Rapporteur*
M. Laurent Gallon	*IUT des Pays de l'Adour*	*Examinateur*

Ecole doctorale :
Systèmes (EDSYS)
Unité de recherche :
Laboratoire d'Analyse et d'Architectures des Systèmes (LAAS-CNRS)
Directeur(s) de Thèse :
M. Frédéric Gouaisbaut *Université Toulouse III - Paul Sabatier*
M. Yann Labit *Université Toulouse III - Paul Sabatier*

À ma mère,
je sais que tu es là
tout le temps...

Avant Propos

Ce travail est effectué au Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS) du Centre National de la Recherche Scientifique (CNRS) au sein des groupes Méthodes et Algorithmes pour la Commande (MAC) et Outils Logiciels pour la Communication (OLC).

Mes remerciements les plus sincères à mes directeurs de thèse Monsieur Yann Labit et Monsieur Frédéric Gouaisbaut qui m'ont épaulée avec une patience remarquable, une richesse pédagogique et un enthousiasme marquant ces trois dernières années. Vous restez mes idoles en tant que chercheurs, encadrants et enseignants.

Je tiens à remercier Monsieur Jamal Daafouz, Professeur des Universités, et Monsieur Jean-Pierre Thomesse, Professeur des Universités, de consacrer une grande partie de leur temps pour juger ce travail. Je suis très honorée que Monsieur Laurent Gallon, Maître de Conférences ait accepté de participer au jury de soutenance et que Monsieur Jean-Louis Calvet, Professeur des Universités première classe ait présidé le jury de soutenance.

Je souhaite remercier les personnes qui ont contribué directement ou indirectement aux résultats exposés dans ce manuscrit : Monsieur David Gauchard, Ingénieur en Informatique au LAAS et Monsieur Laurant Gallon, l'attaqueur de mon routeur au LAAS.

Je remercie Christèle Mouclier et Hélène Thirion pour la logistique, ainsi que les membres du LAAS et le service de documentation.

Je souhaite également exprimer mes sincères remerciements aux membres du groupe MAC pour l'accueil et l'ambiance conviviale au LAAS et surtout durant les Workshops du groupe : SkyMAC et AlbiMAC. Un grand merci aux permanents Lucie, Denis, Sophie, Isabelle, Fred, Dimitri, Didier, Germain, les deux Christophe, Vincent, Alexandre, Luca, Alain, Jean. Les amis de bureau Ixbalank, Luiz, Yoshio, merci pour l'aide et l'encouragement de tous les jours et le partage des soucis et du succès. Je ne vais jamais oublier les doctorants et docteurs MAC, avec qui j'ai passé des pauses cafés et des soirées autour des discussions les plus amusantes qui partent dans tous les sens : Mirko, Mauricio, Cristiano, Jean-François, Mathieu, Georgia, Francesco, Razvan, Tung, Florian, et sans oublier les ex-MAC : Thomas, Giorgio, Bogdan, Josep, Mounir, Akin et Martin. Merci pour les moments conviviaux à la salle *Frigo* avec les membres de OLC et TSF : Yann, Pascal, Thierry, Yassine, Itou, Ahmed, Baptiste, Fred, Roxana, Johan, Rodrigo, Jorgevski, Lionel, Nouha, les deux Guillaume, Akram, Rim et Oss.

Ma collocataire et ma soeur Layoul, je te remercie d'être là, durant les plus beaux voire les plus mauvais jours, je n'oublierais jamais ta compagnie. La famille Hankache Walid, Adoul et mon cher Géo, je vous remercie pour m'avoir entourée d'une telle affection familiale. Mes amis de tous les jours, je vous remercie pour faire partie de ma vie : Julie (bientôt Doc :)), Jako (Le danseur :)), Fares, Georges, Lalous, Zouz, Roro, Antoine, Mo (mon support technique et moral), Ghass, Wiss, Moe Ali, Serge (Smily face merci pour me garder l'esprit haut), Alain, Antonio.

Mes amies depuis toujours Grace, Thery et Nancy, même si les jours nous ont mises chacune dans un continent, je vous remercie pour votre présence à mes côtés.

Ma famille, les mots ne peuvent jamais exprimer ma reconnaissance. Mon père, ma sis Cynthia, ma grand-mère, mes tantes Diana, Mary, Viki, Lili, Nounou, vous étiez et vous restez toujours le support dont j'ai besoin dans toutes les étapes de ma vie, je vous dois ce que j'en suis. Mon autre partie Tony, merci pour la patience, le soutien et surtout l'amour dont tu m'a fais preuve tout au long de ces cinq dernières années. Vivre dans l'attente a bien donné son fruit.

Table des matières

Liste des notations

$\mathbb{R}$	ensemble des nombres réels
$\mathbb{R}^+$	ensemble des nombres réels positifs
$\mathbb{C}$	ensemble des nombres complexes
$\mathbb{R}^{n\times m}$	ensemble des matrices à n lignes et m colonnes
$\mathbb{I}_n$	matrice identité de dimensions $n \times n$
$\mathbb{O}_{n\times m}$	matrice nulle de dimensions $n \times m$
$[a,\, b]$	intervalle fermé dans $\mathbb{R}$ d'extrémités a et b
$\{1, \ldots, N\}$	ensemble des N premiers nombres entiers positifs
$\mathcal{C}$	ensemble des fonctions continues
$Re(s)$	partie réelle du nombre complexe s
$Im(s)$	partie imaginaire du nombre complexe s
$x^{(i)}(t)$	$i^{ième}$ dérivée temporelle de $x(t)$
$\dot{d}(t), \ddot{d}(t)$	dérivée première, et seconde de $d(t)$
A^T	matrice transposée de A
A^{-1}	matrice inverse de A
$A > 0\,(< 0)$	matrice définie positive (définie négative)
$det(A)$	déterminant de la matrice carrée A
$E[X]$	Espérance mathématique de la variable aléatoire X

Liste des Abréviations

TCP	Transmission Control Protocol
IP	Internet Protocol
UDP	User Datagram Protocol
ICMP	Internet Control Message Protocol
ACK	Acknowledgement ou acquittement
QdS	Qualité de Service
AQM	Active Queue Management
RED	Random Early Detection (AQM de type informatique)
PI	Proportionnel Intégral (AQM basé sur l'Automatique)
Gain-K	retour d'état statique (AQM basé sur l'Automatique)
TBF	Token Bucket Filter (mécanisme de filtrage du buffer du routeur)
DdS	attaque par Déni de Service
DDdS	attaque par Déni de Service Distribuée
TFN2k	Tribal Flood Network 2K (logiciel d'émission d'attaques)
NS	Network Simulator (ou Simulateur des Réseaux)
IOD	Independent Of Delay (ou Indépendant du retard)
DD	Delay Dependent (ou Dépendant du retard)
LMI	Linear Matrix Inequality (ou Linéarité Matricielle)

Table des figures

Introduction

De nos jours, les réseaux de communication constituent l'un des domaines technologiques les plus répandus dans la vie professionnelle autant que personnelle. Nous assistons à des interconnexions de plus en plus volumineuses qui sont soumises régulièrement à des perturbations. Citons par exemple la surcharge du réseau subite par des foules de connexions lors d'un match de football, ainsi que le pari en ligne qui est une nouvelle mode hypercroissante. Néanmoins, des utilisateurs malveillants utilisent l'interconnexion de réseaux à des fins abusives voire lucratives. Effectivement, les inondations des paquets envoyées causent le craquage ou le réamorçage de certains serveurs. De telles opérations anormales légitimes ou illégitimes sont étiquetées comme des anomalies. La détection de ces anomalies est devenue dans notre époque un aspect majeur de garantie de la *Qualité de Service* des réseaux de communication. Au cours de cette thèse, les anomalies sont observées au niveau d'un routeur dans les réseaux utilisant le protocole TCP (*Transmission Control Protocol*). Dans une thématique pluridisciplinaire, nous appliquons les outils de la théorie de l'Automatique à la problématique d'observation dans les modèles TCP/IP. Nous introduisons nos travaux sur les plans théorique et pratique afin de motiver l'ensemble de notre étude.

Dans l'univers des réseaux informatiques, TCP est l'un des principaux protocoles de transmission de données entre deux machines. Dans le protocole TCP, orienté connexion, la communication est assurée de bout-en-bout entre l'émetteur et le destinataire sans se préoccuper des machines intermédiaires. En outre, il est caractérisé par sa fiabilité grâce à un système d'accusés de réception permettant d'assurer une bonne réception mutuelle des données. Un phénomène de congestion se manifeste dans le routeur lorsque le débit de données entrant dépasse sa capacité. Une fois le buffer du routeur saturé, tous les nouveaux paquets arrivant sont éjectés, et donc potentiellement perdus. C'est pourquoi, un algorithme d'évitement de congestion ajuste le taux d'émission des sources TCP en fonction de la réception ou non-réception d'un acquittement. Pour gérer l'éjection ou le marquage de paquets au niveau du routeur, des algorithmes de gestion de la file d'attente (*Active Queue Management* ou AQM) sont associés au fonctionnement de TCP. L'AQM dote les paquets de probabilités d'éjection suivant le niveau de remplissage du buffer du routeur dans le but d'améliorer le débit et prévenir les sources de l'augmentation du niveau de congestion. Cependant, le fonctionnement normal du routeur est régulièrement perturbé par des flux de données appelés anomalies.

Les anomalies dans les réseaux de communication sont causées par des problèmes physiques ou techniques comme la panne d'électricité ou les échecs de serveur de fichier, des changements brusques causés par le trafic légitime comme la surcharge du réseau, les foules subites. Les comportements à risque des utilisateurs internes, et les comportements intentionnellement malveillants sont les plus dangereux à détecter comme le sont les attaques de *Déni de Service* (DdS) et *Déni de Service Distribué* (DDdS). Dans des réseaux de communication de plus en plus interconnectés, détecter ce genre d'anomalies devient crucial et la supervision est devenue depuis plusieurs années un sujet de recherche de grande importance. En informatique, les systèmes de détection sont divisés en deux : les *Systèmes de Détection des Intrusions* (IDS) qui utilisent des modèles d'attaques bien connues pour identifier des

signatures, et les *Systèmes de Détection des Anomalies* (ADS) qui cherchent principalement les activités qui dévient significativement des opérations normales. Le travail de cette thèse se situe dans le cadre général des systèmes ADS dans une recherche novatrice de reconstruction des flux survenant au niveau du routeur.

Notre étude s'est focalisée sur un routeur en congestion communiquant N flux TCP aux destinataires. Le comportement du protocole TCP en phase d'évitement de congestion est représenté par un modèle mathématique. En supposant le trafic suffisamment fluide, la variation de la fenêtre de congestion moyenne des sources est représentée par une équation différentielle comprenant des retards. Dans ce modèle, l'émission des flux dépend de la probabilité d'éjection des paquets obtenue par le mécanisme d'AQM en fonction de la longueur de la file d'attente au niveau du routeur. Pour notre étude, nous considérons l'anomalie comme un trafic passant dans le routeur. Un signal est par conséquent ajouté à la dynamique de la longueur de la file d'attente afin de modéliser le trafic supplémentaire inconnu par le routeur. Nous obtenons ainsi un système dynamique perturbé par une entrée inconnue. Ceci entre dans le cadre général des systèmes à entrées inconnues qui sont étudiés par la théorie de la commande, aussi bien pour construire des correcteurs robustes à ces perturbations que pour détecter et observer des signaux exogènes. Plus spécifiquement, l'objet de notre travail est de développer des observateurs permettant ainsi de reconstruire le profil des anomalies. Différentes techniques d'observation sont alors étudiées :

- Les observateurs à entrées connues sont conçus afin de reconstruire des profils d'anomalies pouvant se mettre sous de formes polynômiales. Ces formes couvrent une large gamme de signaux d'anomalies comme les constantes, en rampes et variables. L'anomalie et ses dérivées successives sont introduites dans le vecteur d'état du système linéarisé autour d'un point d'équilibre. L'analyse de stabilité des observateurs de *Luenberger* à retards construits pour le système résultant est étudiée suivant les deux méthodes de stabilisation dépendante du retard (DD) et indépendante du retard (IOD).
- Une approche que nous suggérons pour la détection des anomalies inconnues est l'approche par *modes glissants*. Les avantages des modes glissants sont la robustesse vis-à-vis des incertitudes paramétriques et des défauts et la convergence en un temps fini. Le principe d'application des modes glissants aux observations est de forcer l'erreur à atteindre et maintenir une valeur nulle en temps fini. L'observation de premier ordre est construit pour le modèle de TCP, ensuite une conception plus générique d'observateur d'ordre supérieur est abordée pour les modes glissants.

Les performances des observateurs sont vérifiées à l'aide des logiciels de modélisation : Matlab/Simulink et le simulateur des réseaux NS-2. Sur le plan pratique, une étape intéressante a consisté à prendre en compte des caractéristiques plus réalistes pour le trafic Internet. Nous introduisons dans ce mémoire une approche récente qui permet de rejouer des traces capturées par des équipements de métrologie dans le simulateur NS-2, de façon à reproduire les sources de trafic réalistes et les comportements des utilisateurs. Pour des expérimentations réelles sur le trafic TCP, une analyse hors-ligne de la trace permet d'extraire les flux TCP ainsi que leurs caractéristiques comme le temps d'aller-retour moyen des paquets TCP, les capacités utiles des liens, et les tailles des paquets TCP émis. Ainsi, le modèle de trafic Internet est remplacé dans NS-2 par un outil de simulation de trafic sous de conditions réalistes.

Dans le cadre du travail que nous avons exposé, ce mémoire est organisé comme suit.

Dans le chapitre 1, nous présentons un état de l'art sur le protocole TCP et les techniques d'observation des défauts pour les systèmes à retards. En premier lieu, nous détaillons le fonctionnement du protocole TCP, les types d'anomalies, surtout de dénis de Service, et les moyens informatiques pour les détecter. Ensuite, les principales étapes menant au modèle fluide adopté sont introduites. Après quelques généralités sur la stabilité des systèmes à retards, nous proposons les techniques de l'Automatique comme outils de détection et reconstruction des défauts : les observateurs à entrée connue et les observateurs à entrée inconnue.

Dans le chapitre 2, nous étudions des méthodologies d'observation des anomalies dans le modèle TCP/IP qui peuvent se mettre sous la forme polynômiale. Pour le système augmenté de l'anomalie et ses dérivées, nous développons tout d'abord des observateurs linéaires pour les systèmes linéaires à retards. Une fonctionnelle de Lyapunov-Krasovskii permet d'élaborer des conditions de stabilité par l'approche indépendante du retard (IOD). Des observateurs d'ordres réduits sont proposés pour observer la partie inconnue de l'état qui comprend la fenêtre de congestion, l'anomalie et ses dérivées. Par ailleurs, la synthèse d'un observateur dépendant du retard (DD) pour le modèle d'état complet est complétée par une approche robuste prenant en compte des incertitudes sur une plage de valeurs du retard.

Le chapitre 3 est consacré aux techniques des modes glissants pour la reconstruction des anomalies. Nous rappelons les principes des modes glissants, ensuite les algorithmes de glissement d'ordres un et supérieur. Nous proposons plusieurs algorithmes d'observation dans le cadre de détection d'anomalies dans le modèle TCP/IP. L'observateur glissant d'ordre 1 engendre normalement des oscillations de très hautes fréquences appelées réticence grâce à la commande discontinue ajoutée à la dynamique d'observation afin d'assurer le glissement. L'observateur glissant d'ordre 1 conçu en premier lieu pour le modèle TCP/IP induit une réticence qui ne peut pas être réduite par des fonctions continues à grands gains. Par conséquent, nous avons eu recours aux filtres passe-bas pour reconstruire les anomalies. L'observateur d'ordre 2 basé sur l'algorithme du super-twisting est ensuite proposé pour améliorer les performances d'observation des d'anomalies.

La validation de chacune de nos méthodologies à l'aide du simulateur Matlab/Simulink étant présente dans les chapitres précédents, nous nous orientons vers le simulateur de réseaux NS-2 dans le chapitre 4. Pour chaque type d'observateur élaboré, des topologies de réseaux comprennent des routeurs contrôlés par différents mécanismes d'AQM : informatiques comme le RED et ceux basés sur la théorie de commande comme le Proportionnel Intégral (PI) et le retour d'état statique. Pour la reconstruction des anomalies, les suivis des débits constants et en rampes sont comparés entre les observateurs. Une étude est aussi menée sur les faux négatifs et positifs qui permettent d'analyser la qualité de détection ainsi que la rapidité de détection des apparitions et disparitions des anomalies. Nous insistons sur la réduction du taux d'émission de faux négatifs qui sont les plus importants à analyser puisqu'ils traduisent le délai mis par l'observateur à détecter la présence d'une anomalie. Dans un second temps, nous entamons la partie du rejeu de traces de trafics TCP réels en détaillant la procédure adoptée. Les étapes de capture du trafic, du traitement hors-ligne

de la trace jusqu'au rejeu des flux dans NS-2 sont présentées. Chacun des observateurs est évalué en détection et reconstruction des anomalies sous des conditions réalistes des flux TCP.

Nous concluons dans une dernière partie les différents points abordés tout au long de cette thèse. Nous proposons finalement des pistes de travaux futurs à court terme concernant l'amélioration des techniques d'observation, ainsi qu'à long terme en ouvrant la voie à des techniques de supervision du modèle TCP/IP apportées par l'Automatique.

Chapitre 1

Contexte

Les réseaux de communication sont soumis à des comportements anormaux provenant de problèmes physiques ou de comportements malveillants, parfois intentionnels. Ces opérations anormales sont souvent référées à des anomalies dites légitimes ou illégitimes. Certaines sont reliées aux problèmes de performance comme des échecs de serveur de fichier, des tempêtes de broadcast (*broadcast storm*), etc. et d'autres concernent la sécurité pouvant mener au déni de service (DdS) d'une machine ou même d'une partie du réseau.
Notre travail durant cette thèse consiste à utiliser la théorie de l'Automatique pour reconstruire le trafic d'anomalies passant par le modèle TCP/IP. Dans ce chapitre, nous nous sommes focalisés sur le cadre du travail. En premier lieu, nous présentons brièvement le fonctionnement du protocole TCP, une description des anomalies et les moyens pour les détecter et les classer. Nous proposons ensuite des techniques de contrôle comme outils de détection et reconstruction du profil d'anomalies. Pour cela une représentation mathématique de la dynamique du modèle TCP/IP est nécessaire. Les principales étapes menant au modèle adopté sont introduites, ainsi que la modélisation des anomalies dans ce modèle. En supposant le trafic suffisamment fluide, la dynamique du modèle TCP/IP peut être représentée par un modèle de type systèmes à retards. Ayant défini le cadre de notre travail, certaines notions comme la stabilité des systèmes à retards sont présentées. Dans le domaine de détection des défauts, les observateurs peuvent être classés, selon la connaissance du profil des défauts, en observateurs à entrée inconnue et à entrée inconnue. Différentes techniques d'observation sont étudiées pour chacune des deux approches, avant la présentation des objectifs détaillés de ce travail.

1.1 Généralités sur le modèle TCP/IP

Durant les dernières décennies, nous avons assisté à l'intégration progressive dans l'Internet de technologies sophistiquées générant une quantité d'applications de plus en plus volumineuse. Afin de normaliser les communications entre ordinateurs, un modèle TCP/IP a été imposé comme modèle de référence en lieu et place du modèle OSI (*Open Systems Interconnection*) qui était mis au point par l'*Organisation Internationale des Standards* (ISO) [Tanenbaum 1994], [Stevens 1994]. TCP/IP désigne communément une architecture réseau où deux protocoles sont étroitement liés : un protocole de transport, TCP (*Transmission Control Protocol*) et le protocole de réseau IP (*Internet Protocol*) (voir Figure 1.1). C'est grâce à des contraintes militaires que le Ministère Américain de la Défense a créé le modèle de référence TCP/IP ayant besoin de concevoir un réseau pouvant résister à toutes les conditions, en particulier à une guerre nucléaire.
Dans un monde connecté par différents types de médias de communication tels que les fils de cuivre, micro-ondes, fibres optiques et liaisons satellites, ce modèle assurait une transmission de paquets capable d'aboutir à coup sûr et sous n'importe quelle condition.

Afin de pouvoir appliquer le modèle TCP/IP indépendamment du système d'exploitation, le système de protocoles TCP/IP a été décomposé en plusieurs modules effectuant les uns après les autres des tâches précises [Hassan 2004]. Le modèle OSI est un modèle qui comporte sept couches comme le présente la Figure 1.1, tandis que le modèle TCP/IP n'en comporte que quatre décrites ci-dessous :

- **La couche Accès réseau** "regroupe" la couche physique et la couche liaison de données du modèle OSI. Elle contient toutes les spécifications concernant la transmission de données sur un réseau physique, qu'il s'agisse de réseau local, de connexion à une ligne téléphonique ou n'importe quel type de liaison à un réseau. Elle prend en charge l'acheminement des données sur la liaison, leur synchronisation, la conversion des signaux (analogique/numérique) et le contrôle des erreurs à l'arrivée.

- **La couche Internet ou Réseau** est la clé de voûte de l'architecture. Elle résout le problème de l'acheminement de paquets dans n'importe quel réseau et indépendamment les uns des autres jusqu'à destination. Le point critique de cette couche est le routage basé sur les adresses IP des machines. D'autres protocoles sont gérés comme ICMP (*Internet Control Message Protocol*) utilisé pour transférer des messages de diagnostic liés aux transmissions IP et IGMP (*Internet Group Management Protocol*) utilisé pour gérer les groupes multicast.

- **La couche Transport** assure le transport des messages parvenus de la couche application entre les terminaux source et destination d'une application donnée. Cette couche a recours à deux principaux protocoles de transport : TCP et UDP (*User Datagram Protocol*). TCP est un protocole fiable qui procure à ses application un service orienté connexion garantissant la livraison des messages et assure un contrôle de flux. UDP procure aux applications un service sans connexions où la transmission de données se fait en absence de toute procédure de mise en présence et sans aucun contrôle de flux.

- **La couche Application** est responsable de l'exécution de différentes applications réseau. Dans ce but, elle a recours à plusieurs protocoles de haut niveau, comme par exemples TFTP (*Trivial File Transfer Protocol*) pour les transmissions de fichiers, SMTP (*Simple Mail Transfer Protocol*) pour les applications de messagerie électronique, HTTP (*HyperText Transfer Protocol*) pour le Web. Le point important pour cette couche est le choix du protocole de transport à utiliser. C'est la couche de plus haut niveau d'abstraction, elle utilise les couches inférieures pour communiquer en transparence avec d'autres machines.

Chaque couche ajoute un entête au paquet de données lorsque ce dernier passe par la couche. Ainsi le paquet de données est appelé message au niveau de la couche Application, il est ensuite encapsulé sous forme de segment dans la couche Transport. Nous parlons de datagramme dans la couche Internet et de trame dans la couche Accès réseau. Il est important de noter que le terme paquet TCP est utilisé de façon interchangeable avec le terme segment TCP. C'est l'appellation que nous allons adopter pour la modélisation du comportement du protocole TCP.

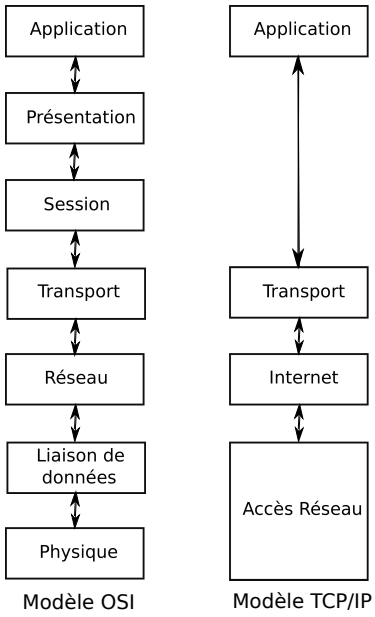


FIG. 1.1: Modèles OSI et TCP/IP.

1.2 Protocole TCP

1.2.1 Fonctionnement

TCP est un des principaux protocoles de la couche transport du modèle TCP/IP. Dans le protocole orienté connexion, la communication entre l'émetteur et le destinataire est principalement assurée avec abstraction des machines intermédiaires, d'où l'appellation de réseau de bout-en-bout. La fiabilité fournie par TCP consiste à remettre les segments sans perte ni duplication alors même qu'il utilise IP qui est un protocole de remise non fiable [Stevens 1994], [Hassan 2004].
La fiabilité est assurée à partir du mécanisme d'accusé de réception (*acquittement ou Acknowledgement* ACK) présenté brièvement dans la Figure 1.2. Après chaque émission d'un segment, la source attend que l'ACK correspondant lui est parvenu du destinataire, traduisant ainsi la réception du segment émis. De plus, la source se sert d'un temporisateur à chaque envoi d'un segment qui calcule le délai d'attente de l'ACK. Lorsque la temporisation expire sans qu'il n'ait reçu un ACK, la source considère que le segment est perdu. Celle-ci s'appelle la perte suite à une expiration du temps ou *Time Out* (TO). D'autre part, suite à des problèmes dans le réseau provoquant la perte de l'accusé, la temporisation peut expirer alors que le paquet a atteint sa destination. La source renvoie de nouveau le segment, cependant la destination éliminera les doublons parce qu'elle garde une trace des segments reçus. Une autre situation critique dans le fonctionnement de TCP est la réception de paquets en désordre. Ceci déclenche également un ACK dupliqué envoyé par le destinataire à la source. Pour éviter de retransmettre un paquet qui n'a pas été perdu mais qui arrive en retard par rapport à un autre paquet, l'émetteur attend *trois ACK dupliqués* avant de déclencher une retransmission. Ainsi, il y a une probabilité élevée que le paquet ait vraiment été perdu. Cet indice est appelé perte par réception de trois accusés de réception (3DupAck).
Ces deux derniers phénomènes sont interprétés comme des indices de *congestion* sur le

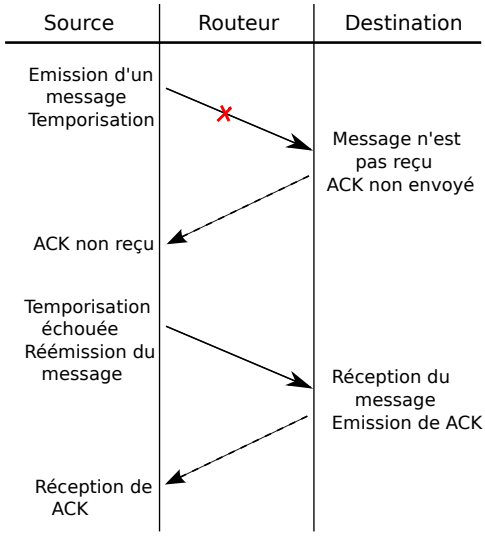


FIG. 1.2: Technique de l'accusé de réception.

parcours vers le destinataire. Ainsi, un algorithme est employé par l'émetteur pour réguler son taux d'envoi en fonction du degré de congestion qu'il perçoit. C'est l'objectif de la présentation du paragraphe suivant.

1.2.2 L'algorithme de contrôle de congestion

La méthode de la fenêtre d'émission autorise la source à envoyer plusieurs paquets à la suite [Stevens 1994]. Le problème est de savoir combien de paquets peut-on envoyer sans saturer le récepteur ou les routeurs. L'objectif est de garder un débit relativement élevé tout en évitant la congestion du réseau. D'une manière globale, l'émetteur accroît sa vitesse d'envoi d'une manière additive dès qu'il perçoit que le parcours de bout-en-bout est libre, et il la réduit de manière multiplicative au moindre signe de congestion. L'algorithme de contrôle de congestion de TCP est donc appelé *Additive Increase Multiplicative Decrease* (AIMD). Enfin, les phases de *Slow Start* [Jacobson 1988] et *Fast Recovery* [Jacobson 1990] ont été proposées pour rendre le protocole TCP plus efficace. L'algorithme résultant définit la version de TCP la plus répandue actuellement. Il est formé de deux phases principales : la phase du Slow Start et la phase de l'évitement de congestion.

a- Phase du Slow Start

- Soit W la fenêtre de congestion émise par la source. Initialement $W = 1$.
- Le seuil de Slow Start appelé *ssthresh* (*slow start threshold*) est initialement égal à la moitié de la taille maximale de W.
- Suite à chaque accusé de réception l'émetteur augmente sa taille exponentiellement.
  - Si W atteint *ssthresh*, alors fin de la phase Slow Start.
  - S'il y a perte de paquets et $W < ssthresh$, alors $ssthresh = W$ et recommencement de Slow Start.

b- Phase d'évitement de congestion

- Si le flux est transmis avec succès, sur réception de l'ACK, l'émetteur augmente son taux d'envoi, ainsi $W = W + 1$.
- En cas d'indice de congestion :
  - Si la source n'a pas de réponse du destinataire, alors perte par Time Out (TO), $W = 1$ et la phase de Slow Start se déclenche.
  - Si la source reçoit trois acquittements identiques (3DupAck), $W = W/2$. C'est la phase de Fast Recovery et l'évitement de congestion recommence directement.

Cet algorithme conduit à la courbe présentée dans la Figure 1.3 traduisant la variation de la fenêtre de congestion selon les trois phases : Slow Start, Évitement de congestion et Fast Recovery.

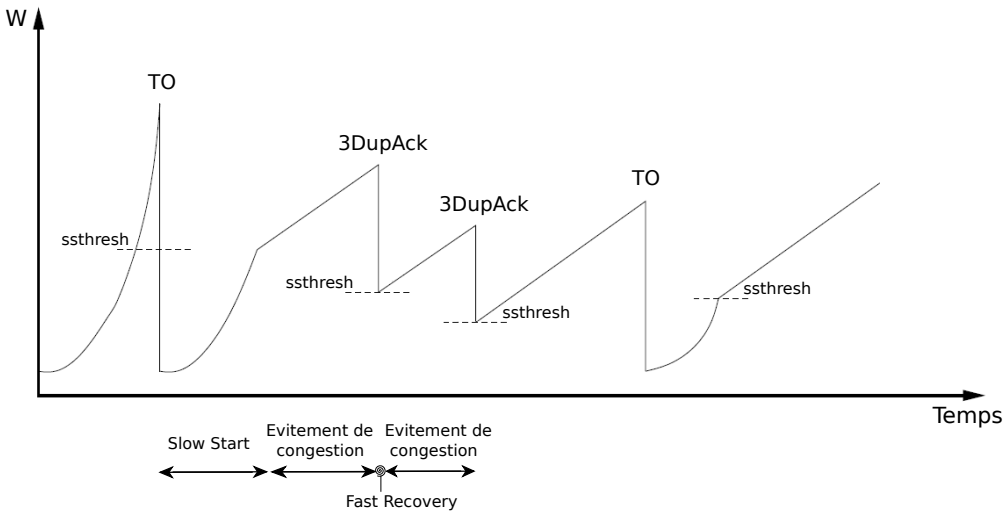


FIG. 1.3: Evolution de fenêtre de congestion selon les phases du protocole TCP.

Depuis le développement de l'algorithme de contrôle de congestion basique, connu sous le nom de *TCP Tahoe* en 1988, plusieurs versions ont apporté des améliorations sur le fonctionnement de TCP. Citons *TCP Reno, Vegas, NewReno, SACK* [Hassan 2004]. Les principaux changements ont touché la phase du Slow Start, la phase de l'évitement de congestion et surtout la phase du Fast Recovery traduisant les réponses aux multiples pertes. Le Tableau comparatif 1.1 résume les comportements des différentes versions de TCP dans chaque phase.

Exemple de simulation sur le modèle TCP/IP

Dans cette partie, nous allons illustrer l'évolution de la fenêtre de congestion par un exemple simple sur un simulateur de réseaux NS-2 [Fall 2010]. Comme nous le montrons dans la Figure 1.4, 20 sources TCP envoient des paquets vers un routeur de capacité $10Mbps$.
Dans la Figure 1.5a nous pouvons observer les phases d'évitement de congestion où la fenêtre de congestion augmente linéairement jusqu'à la perte de paquets. En parallèle la file d'attente au niveau du routeur dans la Figure 1.5b se remplit progressivement jusqu'à atteindre la limite maximale du buffer (fixée à 800 paquets), par la suite les paquets suivants

	Tahoe	**Reno**	**NewReno**	**SACK**	**Vegas**
Évolution de W dans Slow Start	$W+1$	$W+1$	$W+1$	$W+1$	augmente après un futur RTT précis
Évolution de W durant l'évitement de congestion	$W+\frac{1}{W}$	$W+\frac{1}{W}$	$W+\frac{1}{W}$	$W+\frac{1}{W}$	augmente linéairement selon la différence(Diff) entre les débits actuels et estimés d'une connexion
Passage de Slow Start à Évitement de congestion lorsque	$W=$ *ssthresh*	$W=$ *ssthresh*	$W=$ *ssthresh* (*ssthresh* peut être estimé)	$W=$ *ssthresh*	si Diff<débit fixé
Fast Recovery	-	termine avec réception de ACK partiel (partie de segments acquités)	continue avec ACK partiel et renvoie la partie non acquitée	continue avec SACK partiel indiquant quelle partie acquitée	renvoie avec ACK si RTT >TO

TAB. 1.1: Les versions de TCP.

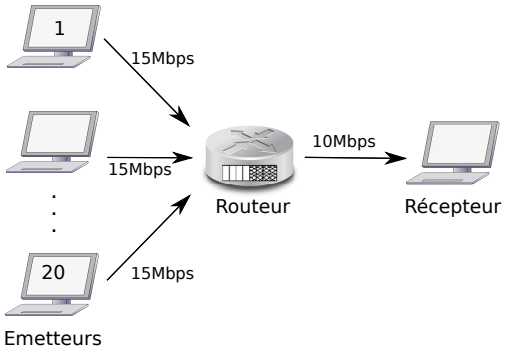
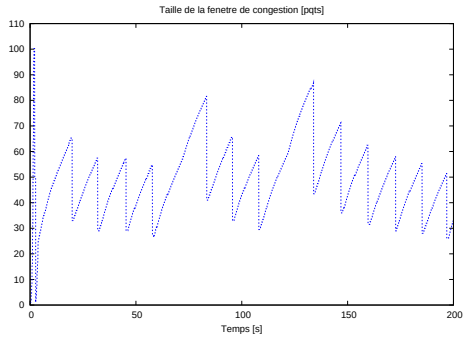
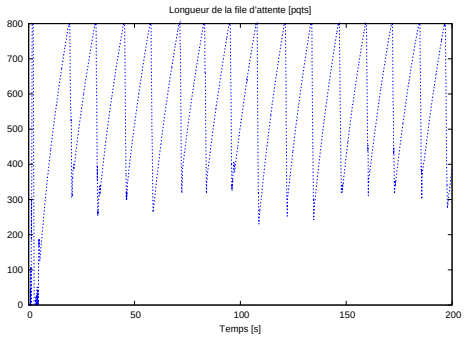


FIG. 1.4: La topologie du modèle TCP/IP en phase de congestion.

sont éjectés. Ce mode de gestion du buffer est appelé *DropTail* où le routeur jette tous les paquets lui parvenant une fois sa capacité maximale atteinte. Évidemment, ce mécanisme induit de fortes oscillations de la file d'attente (AQM) avec des saturations fréquentes. C'est pourquoi, des algorithmes de gestion de la file d'attente au niveau des routeurs ont été par la suite élaborés pour éviter ce genre de phénomène. Ils seront le sujet de discussion dans la partie 1.4.



(a)



(b)

FIG. 1.5: Evolution de : (a) la fenêtre de congestion d'une source TCP, (b) la file d'attente au niveau du routeur.

1.3 Qualité de Service (QdS)

La QdS est généralement un sujet large qui touche aux différents aspects de construction, gestion et utilisation des ressources du réseau. Cette notion permet aux fournisseurs de services (entreprises, opérateurs) de s'engager formellement auprès de leurs clients sur les caractéristiques de transport des données applicatives sur leurs infrastructures IP. Les travaux sur la QdS ont été effectués dans un contexte d'architecture de couches individuelles. Le traitement le plus complet se situe dans les couches de Transport et Réseau. C'est à ce niveau que se situent les problèmes les plus critiques.
Comme la couche de Transport est l'entité principale entre les fournisseurs et les utilisateurs d'une application qui garantit une transmission de données fiable, il faut envisager

son rôle de garant de la *Qualité de Service* fournie par la couche Réseau [Juanole 1995], [Tanenbaum 1994]. En effet, la couche Transport réalise la jonction entre les souhaits de l'utilisateur et les dispositions de la couche Réseau. Au niveau de la couche de transport, les paramètres principaux de la QdS sont les suivants :

- le temps d'établissement de la connexion qui représente le laps de temps entre la demande de connexion par l'utilisateur et la réception de confirmation. Ce délai doit évidemment être court.
- la probabilité d'échec d'une connexion qui doit s'établir dans un délai maximal suite à un problème comme la congestion du réseau.
- le débit de la liaison qui traduit le nombre d'octets pouvant être transmis pendant une seconde,
- le temps de transit qui est le temps écoulé entre l'émission et la réception d'un message,
- les taux d'erreurs de transmission doit être très faibles comme c'est la tâche de la couche de Transport.

Les autres couches participent également à la QdS. Les services fournies par la couche Application ont besoin des paramètres de QdS tels que le temps de réponse, le taux d'erreurs tolérable et le coût de la communication. Par contre, aucun moyen n'est fourni pour manipuler ces paramètres. Au niveau de la couche Accès Réseau, la QdS est traduite par des paramètres entre l'utilisateur et l'émetteur. Des paramètres sont négociés entre ces derniers comme le temps de transit, le débit utile. Certains paramètres sont choisis comme la priorité et la protection de la connexion ; et d'autres sont fixés comme le délai d'établissement d'une connexion réseau et sa probabilité d'échec, le taux d'erreurs, la probabilité d'échec de transfert, la priorité et la protection de la connexion. Dans la couche d'Accès réseau, les paramètres de la QdS dépendent d'une part du type de communication entre les machines : le débit, le temps de transit, le taux d'erreurs, la probabilité de rupture, la priorité et la protection pour deux machines liées directement ; et le temps de transit, le taux d'erreurs, la priorité et la protection lorsque deux machines ne gèrent pas directement la communication entre elles. D'autres part, des paramètres diffèrent selon les supports physiques de l'interconnexion comme la disponibilité du service, le débit, les erreurs et le temps de transit.

Les performances des applications TCP dépendent d'une façon critique sur la gestion de file d'attente dans les liens du réseau. Le management de la file d'attente au niveau des routeurs (ou AQM) représente un aspect important d'assurance de la QdS.

1.4 Active Queue Management (AQM)

Le contrôle de la file d'attente est défini comme l'algorithme qui gère la longueur de la file d'attente en éjectant ou marquant les paquets lorsque c'est nécessaire ou approprié (Figure 1.6). L'AQM marque aléatoirement les paquets avant que le buffer du routeur soit plein dans le but d'améliorer le débit et prévenir de l'augmentation du niveau de congestion [Floyd 1993], [Hassan 2004]. Ce mécanisme de prévention informe, par retour implicite, les émetteurs de l'apparition d'une congestion. Ainsi, les sources réduisent leur taux d'émission pour diminuer le niveau de congestion. Le marquage aléatoire des paquets arrivant à la file d'attente du routeur empêche le phénomène de réduction des taux d'émission de plusieurs sources en même temps appelé "la synchronisation globale" [Braden 1998].

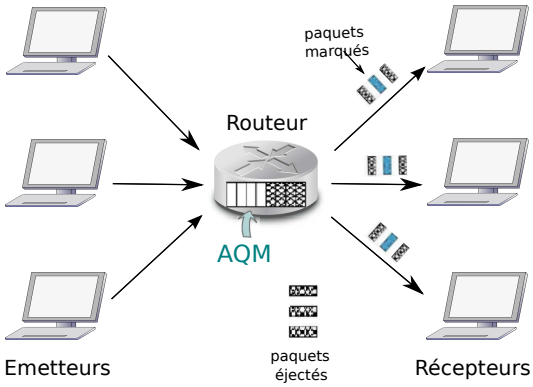


FIG. 1.6: Gestion de la file d'attente par l'AQM.

RED est l'un des AQM les plus utilisés pour la gestion de la file d'attente [Floyd 1993]. La Figure 1.7 montre la fonction d'éjection de paquets utilisée par RED. Un routeur ayant RED comme AQM accepte tous les paquets jusqu'à ce que la file d'attente atteigne un seuil minimal Min_{th}. Une fois ce seuil atteint, la probabilité d'éjection de paquets suit une fonction linéaire jusqu'à ce que la file d'attente moyenne atteigne un seuil Max_{th} à partir duquel tous les paquets sont perdus (probabilité d'éjection égale à 1).

Depuis que RED a été proposé en 1993, de nombreuses approches d'AQM, telles que

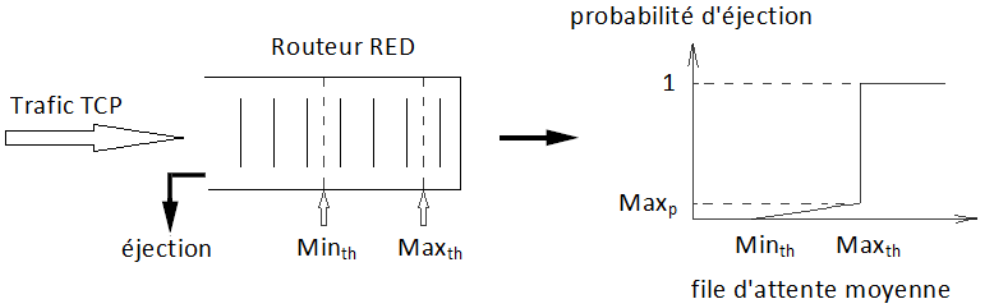


FIG. 1.7: Routeur RED et probabilité d'éjection de paquets.

Adaptive-RED (ARED), *Stabilized-RED* (SRED), BLUE, et *Adaptive Virtual Queue* (AVQ) ont été proposées, et leurs performance sont évaluées dans [Ali Ahammed 2010], [May 2000] et [Ryu 2004].

ARED [Feng 1999] tente de maintenir les paramètres de fonctionnement de RED en ajustant dynamiquement la probabilité maximale d'éjection des paquets relativement à la longueur de la file d'attente du routeur.

SRED [Ott 1999] éjecte les paquets avec une probabilité qui dépend de l'estimation du nombre de flux et de la longueur instantanée de la file d'attente. SRED stabilise l'utilisation du buffer indépendamment du niveau de charge du réseau.

BLUE [Feng 2002] utilise la perte de paquets et les événements liés aux liens plutôt que la longueur de la file d'attente pour le contrôle de congestion. BLUE augmente la probabilité d'éjection de paquets suite à un débordement du buffer et diminue la probabilité d'éjection

lorsque le lien devient inoccupé.
AVQ [Kunniyur 2001] utilise une file virtuelle pour réguler l'utilisation du buffer au lieu des valeurs instantanées de la file d'attente. AVQ régularise la taille de la file virtuelle et la capacité de lien virtuelle proportionnellement au taux d'arrivée de paquets et à la probabilité d'éjection.

Ces AQM que nous venons de présenter sont basés sur les outils informatiques. Comme détaillés dans la section 1.2.2, les mécanismes de contrôle sont intégrés dans le protocole TCP pour fournir un transfert fiable des données. La variation de la taille de la fenêtre de congestion peut être exprimée en fonction des acquittements. Ainsi, l'ensemble du système formé par le nombre total de connexions TCP circulant à travers l'Internet représente l'un des plus grands systèmes de contrôle que l'homme a jamais réalisés, en termes de portée géographique et le nombre d'entrées et de sorties. Dans le but d'analyser plus finement l'impact de telles stratégies, des modèles mathématiques ont été proposés récemment pour les flux TCP contrôlés par des mécanismes d'AQM [Misra 1999], [Kelly 1998], [Misra 2000], [Liu 2003]. Des algorithmes basés sur la théorie de la commande ont été conçus pour améliorer les performances et la robustesse du protocole TCP en régularisant la file d'attente à une certaine valeur désirée notée q_{ref}.

1.4.1 Construction d'AQM utilisant la théorie de la commande

Implémenté au niveau d'un routeur, un AQM détermine la probabilité d'éjection des paquets en fonction de la longueur moyenne de la file d'attente. De ce fait, il peut être représenté par une commande par rétroaction pour des modèles représentant TCP tels que le système [Hollot 2001a], [Liu 2003] comme il est montré dans la Figure 1.8. Les auteurs de [Hollot 2002] ont été les premiers à proposer un correcteur Proportionnel Intégral (PI) pour le modèle TCP/IP, ensuite des architectures de commandes utilisant les principes de la théorie de la commande étaient proposées pour des modèles TCP comme le retour d'état statique [Ariba 2009], la commande robuste [Wang 2003], [Manfredi 2004], [Li 1997], la commande prédictive [Witrant 2005], la commande non linéaire [Hollot 2001c], [Michiels 2005] et la commande par commutation de contrôleurs [Cao 2009].

C'est à partir du contrôle de la file d'attente au niveau du routeur et des réponses des sources TCP à la variation du flux que nous pouvons conclure sur le bon fonctionnement du modèle TCP/IP. Si les réactions des sources sont différentes de celles attendues, alors le modèle est dans le cas d'un fonctionnement anormal que nous allons essayer de détecter et quantifier au niveau du routeur.

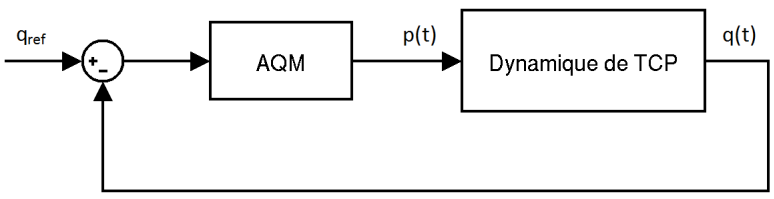


FIG. 1.8: AQM : un contrôle de TCP par rétroaction.

1.5 Anomalies dans le réseau

Les réseaux de communication ont énormément grandi en complexité de sorte que garantir la Qualité de Service (QdS) et particulièrement la sécurité sont devenues de plus en plus difficiles. Effectivement, les réseaux sont fortement sensibles à une large variété de perturbations, souvent désignées comme des anomalies. Les anomalies peuvent avoir un impact significatif sur le comportement normal du réseau [Aussibal 2007], [Lakhina 2004]. Les anomalies peuvent être causées par plusieurs événements : les problèmes physiques ou techniques comme la panne d'électricité ou les échecs de serveur de fichier, les changements brusques causés par le trafic légitime comme la surcharge du réseau, les foules subites, ainsi que les comportements risqués des utilisateurs internes, et les comportements intentionnellement malveillants comme des attaques de déni de service, les intrusions, et d'autres.

1.5.1 Attaques de Déni de Service

Les attaques sont actuellement les anomalies les plus dangereuses comme les Dénis de Service (DdS), les saturations de mémoire, les attaques de mots de passe, et d'autres. Les Dénis de Service représentent la classe d'attaques la plus dangereuse [Lakhina 2004], [Specht 2004]. De telles attaques bloquent pour un utilisateur l'accès à la machine ou retardent le temps de réponse en rendant l'accès inacceptable. Le Déni de Service peut survenir suite à une attaque programmée lorsqu'une personne malveillante surcharge intentionnellement une ressource ou un système ; ou accidentellement lorsqu'un utilisateur déclenche une procédure inappropriée. Dans les deux cas, des mesures de protection doivent être envisagées pour résoudre le problème. En général, il est difficile de prévenir ou d'éviter les attaques DdS. Cependant, le contrôle et la protection peuvent retarder les attaques mais une fois déclenchées, ce sont les moyens de détection qui peuvent restreindre les dégâts parvenus sur le réseau.
Avec un DdS traditionnel, une machine attaque une autre machine. La machine attaquante peut avoir recours à d'autres machines dans le but d'inonder la victime par le flux de paquets ou même un réseau. Ce qui forme un Déni de Service distribué (DDdS) illustré dans la Figure 1.9. La distribution au travers de plusieurs machines rend très difficile la défense contre une telle attaque. Ainsi, l'identification et le blocage de ces attaques parvenant de plusieurs adresses IP est une tâche extrêmement compliquée.

Exemples d'attaques Déni de Service

Nous allons présenter quelques exemples de déni de Service. Nous pourrons nous reporter aux travaux de [Jung 2002], [Hussain 2003], [Chen 2006] pour une présentation plus exhaustive.

- Inondation de SYN (ou *SYN Flooding*) : C'est une attaque qui est lancée lors de l'ouverture de la connexion TCP. La source attaquante envoie des paquets de type SYN demandant l'autorisation d'une connexion de la victime mais avec une adresse erronée correspondant à une machine incapable de répondre. Par conséquent, la machine victime continue à recevoir des SYN et à y répondre sans pouvoir établir une connexion. Un routeur doit bloquer ce type d'attaque en autorisant juste un nombre limité de connexions.

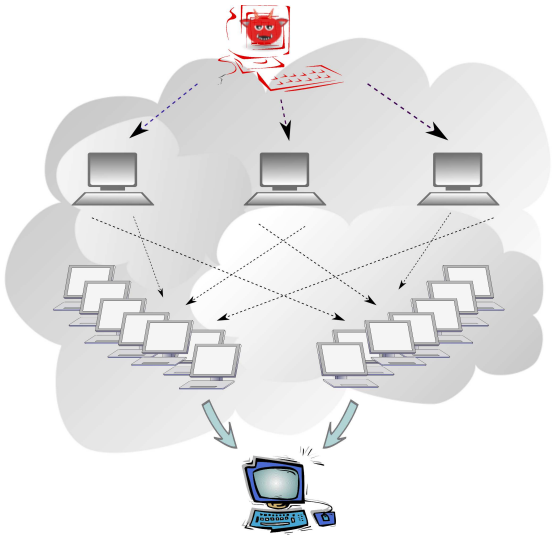


FIG. 1.9: Description d'une attaque DDdS.

- Smurf : La machine attaquante envoie un trafic ICMP de type *écho request* (ping) à une adresse *broadcast* en changeant l'adresse IP source par celle de la machine victime. Alors, cette dernière va recevoir un trafic intense de réponses ICMP de type écho reply. Un routeur doit arrêter ces attaques en interdisant les réponses aux paquets comprenant des adresses de broadcast.
- Inondation par UDP (ou *UDP Flooding*) : Ce déni de service génère une grande quantité de paquets UDP (UDP Packet Storm) soit à destination d'une machine soit entre deux machines. Cette attaque exploite le mode non orienté connexion du protocole UDP. Elle entraîne ainsi une congestion importante du réseau ainsi qu'une saturation des ressources des victimes.
- Teardrop : C'est une attaque qui exploite les défaillances dans le remontage de fragments de paquet IP. La source d'attaques envoie des fragments IP à une machine. Avant d'arriver à la cible, Teardrop crée des séries de paquets IP décomposés en paquets plus petits qui ressemblent au paquet IP original mais avec chevauchement des champs. Lorsque ces fragments sont rassemblés au niveau de l'hôte de destination, certains systèmes vont craquer ou se réamorcer. Cette attaque est supposée être non destructive, mais elle pourrait causer la perte des données non sauvegardées par la cible avant le lancement de l'attaque.

1.5.2 Outils d'attaques de Déni de Service

Il existe de nombreux outils pour lancer des attaques de type Déni de Service. Les plus connus sont : Trinoo [Dittrich 1999], TFN2K (ou *Tribal Flood Network 2000*) [Barlow 2000], et Stacheldraht [Mirkovic 2004]. Tous ces outils contiennent les mêmes fonctionnalités pour lancer des attaques. TFN2K est le logiciel le plus complet en terme de fonctionnalités et possibilités. Il génère différents types d'attaques, des DdS et les DDdS en utilisant le protocole ICMP indétectable par la machine attaquée. Trinoo est l'un des premiers outils comprenant des aspects limités par rapport à TFN2K. Il utilise TCP et UDP pour protocoles de transport. Stacheldraht est spécialisé sur les attaques DDdS. Il combine les fonctionnalités de Trinoo

et TFN2K en ajoutant d'autres techniques comme le cryptage des communications. Il utilise les protocoles TCP et ICMP.

1.6 Détection des anomalies

Le défi majeur dans la détection automatique des anomalies est la diversité des événements que peuvent couvrir les anomalies. Un système de détection général d'anomalies devrait donc pouvoir détecter des anomalies avec des structures différentes, distinguer les différents types et regrouper les anomalies semblables. Cependant, de nouvelles anomalies de réseau apparaissent continuellement, et continueront à survenir avec le temps. Un système de détection ne devrait donc pas être limité à une gamme déterminée d'anomalies. Cela devient la préoccupation principale de la plupart des approches récentes liées à la détection d'anomalies. La littérature est abondante sur les aspects de détection et classification des anomalies [Chandola 2009]. Nous allons présenter quelques approches étudiées.

Détection par type d'anomalie

C'est l'approche la plus facile à réaliser puisqu'elle concerne des types spécifiques d'anomalies. La plupart des travaux de détection et classification des anomalies a été limitée sur certains types comme par exemple les *Port Scans*, les attaques DdS, et les foules subites [Jung 2002] et d'autres [Hussain 2003], [Hodge 2004].

Détection par méthode en ligne ou hors ligne

Les processus de détection peuvent suivre des modes en ligne ou hors ligne [Aussibal 2007], [Cleary 2000]. L'analyse hors ligne permet une étude plus complète des données et l'extraction des informations qui sont inaccessibles en ligne. Ce type d'analyse nécessite des systèmes de mesure et monitoring qui doivent être mis en place pour extraire en temps réel les paquets circulant sur les liens sans que le trafic soit perturbé (comme les cartes DAG [Cleary 2000]). D'autre part, les approches en ligne se révèlent meilleures en termes de réduction des quantités de données sauvegardées à traiter. En outre, elles peuvent nécessiter des équipements spécifiques qui sont coûteux, capables de traiter les données échantillonnées à fréquence élevée dans le but de rester le plus fidèle possible à la représentation exacte des données circulant. Cette technique s'avère intéressante lors de la capture de trafic réseau de coeur.

Un autre critère de classement des techniques de détection des anomalies se rapporte à la méthodologie utilisée pour effectuer l'analyse du trafic. La détection peut être classée selon la base adoptée pour le traitement des données. Nous pouvons donc différencier entre les techniques basées sur des seuils, sur le profil ou à base d'ondelettes.

Détection à base de seuils

Les algorithmes basés sur les seuils nécessitent la définition de seuils, au-dessus ou en dessous desquels, le trafic est considéré comme anormal. Les valeurs assignées à chacun des seuils dépendent des paramètres de trafic, comme par exemple le nombre de paquets et d'octets par unité de temps [Denning 1987]. Cependant, une importante variation de volume n'est

pas une caractéristique intrinsèque aux anomalies de trafic. Cela rend les méthodes basées sur l'étude des variations inadaptées à un grand nombre de perturbations dans le réseau. En outre, le choix des seuils au-delà duquel les anomalies sont détectées n'est pas une tâche évidente pour séparer un trafic régulier d'un trafic anormal [Borgnat 2009], [Borgnat 2007].

Détection à base de profil

La méthodologie basée sur le profil, comme son nom l'indique, nécessite la définition d'un profil normal du comportement du réseau. Puis, lors de l'analyse du trafic actuel, l'algorithme compare avec les valeurs attendues [Krishnamurthy 2003]. Comme pour les approches à seuils, cette approche exige également la spécification des informations actuelles du réseau pour différer des valeurs prévues. Une spécification incorrecte peut alors amener à des faux positifs.

Remarque 1. Un faux positif ou fausse alarme est un résultat d'une prise de décision inverse à la réalité mais ayant un effet positif. Par contre, un faux négatif provient d'une interprétation négative de la décision opposée à la réalité.

Détection à base d'ondelettes

La méthode basée sur les ondelettes consiste à diviser le signal du trafic réseau en ses composantes fréquentielles. Chaque composant est ensuite testé pour les anomalies, dont la durée correspond à son échelle. Ainsi, dans un domaine à basse fréquence, l'anomalie détectée est de longue durée. D'autre part, à des fréquences élevées, les anomalies spontanées sont détectées (par exemple, le bruit), ainsi que les anomalies de courte durée [Barford 2002]. Ces approches sont très efficaces dans la détection des anomalies.

Détection par des techniques plus sophistiquées

La diversité des anomalies dans le réseau a mené au développement de méthodes sophistiquées qui combinent les approches acquises avec de nouvelles méthodes comme les algorithmes basés sur l'intelligence artificielle [Patcha 2007] et sur les techniques de traitement de l'image [Chandola 2009].

Détection à base de signatures

Depuis que les profils des utilisateurs sont définis en fonction de leurs activités, de nombreux Systèmes de Détection d'Intrusions (IDS) ont été élaborés [Denning 1987]. Les IDS sont directement liés à un type particulier d'anomalie grâce à l'identification des comportements spécifiques comme un morceau de code, une séquence de bits dans un paquet, ou une séquence d'appels de fonction malveillants. Une signature est un ensemble de motifs qui sont utilisés pour identifier un cas spécifique d'anomalies dans le contexte des IDS. Les méthodes basées sur les signatures identifient les intrusions à des modèles prédéfinis comme comportements anormaux [Jung 2002]. Un des principaux avantages de l'utilisation de tels IDS, c'est qu'ils sont faciles à développer et à comprendre. En outre, l'exigence de connaissances préalables sur toutes les anomalies qui doivent être détectées par un système, limite le temps de réponse de ce système. Ainsi, le nombre de fausses alarmes augmente avec le nombre des

anomalies non détectées. D'où la nécessité de la mise à jour continuelle des attaques récentes par des développeurs des IDS.

Détection de tout type d'anomalies (ADS)

Ce type de détection n'est pas entièrement dépendant des anomalies comme les IDS à base de signatures. Ces systèmes apprennent le comportement normal du trafic et des systèmes et examinent continuellement les comportements pour identifier les incidents ou les anomalies potentiellement dangereuses. Cette approche reconnaît les anomalies selon leurs impacts sur le réseau, plutôt que de connaître leur configuration dans un incident spécifique passé. Ainsi, ces types de systèmes peuvent détecter un ensemble plus large d'anomalies.
Les systèmes détectant les anomalies sont caractérisées par deux phases : la phase d'apprentissage et la phase de détection [Chandola 2009]. Dans la phase d'apprentissage, le comportement du système est observé en absence d'attaques pour créer un profil de comportement normal. Dans la phase de détection, le profil est comparé avec le comportement actuel du système, et les écarts sont considérés comme des attaques potentielles.
La sélection incorrecte des paramètres qui doivent être vérifiés afin de distinguer une anomalie d'un comportement normal est l'un des inconvénients majeurs des ADS. Ceci est traduit par une augmentation des faux négatifs. Un autre inconvénient potentiel survient lorsque les malfonctionnements sont détectés et des alertes sont générées, il est alors très difficile de corréler ces alertes à un type spécifique d'anomalies. Cependant, détecter les anomalies inconnues est l'avantage principal de la détection d'anomalies. Ainsi, au lieu de définir un grand nombre de signatures pour les différents scénarios d'attaques, il suffit de définir un profil pour une activité normale. Alors, toute activité qui s'écarte de ce profil est anormale. Le compromis entre les deux méthodes est que la détection basée sur les signatures ne peut pas détecter de nouvelles attaques, mais le taux de faux positifs correspondant est plus faible [Barford 2002], [Ye 2000].

Les systèmes de détection des anomalies présentés dans cette section résultent de l'étude du comportement du réseau en observant soit les variations du trafic résultant de l'anomalie soit la structure des paquets anormaux (IDS). Le travail de cette thèse se situe dans le cadre général des ADS. Le but est la détection et la reconstruction des anomalies dans le modèle TCP/IP en utilisant la théorie de la commande appliquée sur un modèle mathématique régissant le comportement normal du protocole TCP. En outre, ce système est soumis à de grandes variations concernant les tailles des fichiers transférés, le nombre de connexions, les mises à jour des routes des paquets ou de défaillances des lignes. Dans un modèle mathématique décrivant la dynamique du modèle TCP/IP, ces phénomènes peuvent être efficacement représentés comme des perturbations.

1.7 Modélisation d'une architecture TCP/IP

Dans la littérature, un certain nombre de représentations mathématiques du modèle TCP/IP a été développé en se focalisant sur des aspects particuliers du protocole ou en ajoutant un nouveau niveau de généralité dans la modélisation de contrôle de congestion du modèle TCP/IP. Ainsi un modèle détaillant les différents types de pertes de paquets est proposé dans [Padhye 1998], un modèle stochastique avec un processus général de pertes est

présenté dans [Altman 2005], et des travaux de modélisation fluide ont été proposés dans [Misra 2000], [Low 2002], [Misra 1999] sans tenant compte des pertes de paquets par Time Out. D'autres recherches se sont spécialisées sur des versions spécifiques de TCP comme : Reno et sa famille, Tahoe [Kumar 1998] ; SACK et Vegas [Wierman 2003].

Lors de communications TCP entre deux machines éloignées, plusieurs types de réseaux peuvent être rencontrés. Si pour une partie de ce réseau, le niveau de pertes de paquets est important, les performances de la connexion entre les 2 machines vont être dégradées vu que le temps de transmission et le temps d'aller-retour d'un paquet TCP et de l'ACK dépendent du parcours du réseau complet. Une proposition sera d'isoler un tronçon du réseau de manière à analyser localement sur un routeur la détection de pertes et la retransmission de paquets. Cette méthode est dite de *splitting* [Maki 2005]. L'idée de base consiste à scinder une communication TCP en plusieurs connexions par l'intermédiaire de proxies permettant de transférer le trafic entre les connexions (c.f. Figure 1.10). Des études expérimentales [Maki 2005] ont montré l'efficacité d'un tel mécanisme qui ne nécessite aucune mesure du réseau ou de modification des protocoles de communication TCP et IP.

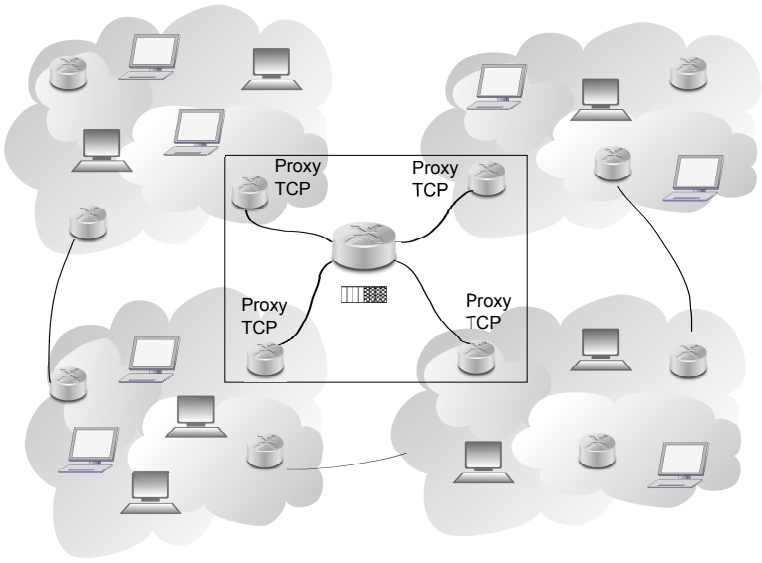


FIG. 1.10: Méthode du *splitting*.

En suivant ce principe, nos travaux sont centralisés au niveau d'un routeur supposé lié en aval à un nombre N de flux TCP comme montré dans la Figure 1.11. Pour représenter la dynamique de TCP au niveau du routeur, nous avons adopté le modèle proposé dans [Misra 2000], [Ariba 2008], [Cao 2009]. Vu sa simplicité, ce modèle est le plus utilisé par la communauté scientifique pour effectuer une analyse quantitative du problème de congestion dans le cadre de la théorie de la commande [Hollot 2002]. Les dynamiques de la fenêtre de congestion et de la longueur de la file d'attente du routeur sont élaborées grâce à des méthodes d'analyse différentielle stochastique en considérant le trafic TCP comme fluide, ce qui implique la continuité de la dynamique de la fenêtre de congestion. D'autre part, les flux envoyés par les sources sont considérés comme homogènes. Toutes les connexions sont supposées avoir le même temps d'aller-retour moyen $R(t)$. Dans ce modèle, les pertes suivent un processus de Poisson qui permet d'avoir une description relativement simple

du comportement de la dynamique de la queue au niveau du routeur. Les résultats des simulations sur un tel modèle ont montré une précision dans la capture de la dynamique TCP [Misra 1999].

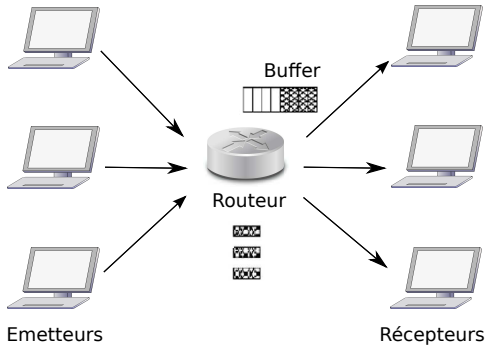


FIG. 1.11: La topologie du modèle TCP/IP.

Étapes de calcul du modèle :

Considérons N connexions TCP traversant un routeur. Pour un seul flux TCP i ($i = 1, \ldots, N$) ayant un temps d'aller-retour $R_i(t)$ donné par :

$$R_i(t) = T_{p_i} + \frac{q(t)}{C},$$

avec T_{p_i} le délai de propagation correspondant au flux i et $q(t)$ la longueur de la file d'attente du routeur. Le délai dans la queue du routeur est représenté par $\frac{q(t)}{C}$ où C est la capacité de lien sortant.

a- Dynamique de la fenêtre de congestion pour un seul flux :

Nous supposons que le nombre de paquets éjectés est modélisé par un processus de Poisson. Basée sur l'algorithme de contrôle de congestion décrit dans la section 1.2.2, la variation de la taille de la fenêtre de congestion W_i de la source i peut être alors décrite par l'équation :

$$dW_i(t) = \frac{dt}{R_i(q(t))} - \frac{W_i(t)}{2} dI_{i3DupAck}(t) + (1 - W_i(t)) dI_{iTO}(t), \tag{1.1}$$

où les indices des pertes par $3DupAck$ et TO sont symbolisés par le processus de Poisson dI où :

$$dI = \begin{cases} 1 & \text{en cas de pertes,} \\ 0 & \text{sinon.} \end{cases}$$

L'équation (1.1) reflète l'algorithme AIMD de contrôle de la fenêtre de congestion : le premier terme explique l'augmentation linéaire d'un paquet durant le temps d'aller-retour (R) qui représente le temps écoulé pour que la source reçoive l'accusé de réception émis par le destinataire. Le deuxième terme traduit la division par 2 après la perte de paquets par 3

acquittements dupliqués ($3DupAck$), et le troisième terme représente le retour à 1 dû à une perte par Time Out (TO).

Notre travail sera axé sur une version simplifiée où la perte de paquets par TO et la phase du Slow Start sont ignorées [Misra 1999]. Les raisons de ces hypothèses peuvent se résumer par le fait que l'algorithme de Fast Recovery permet à TCP de réagir avant que le TO s'exécute. Donc, l'algorithme de Slow Start est arrêté par les déclenchements du Fast Recovery. De plus, la phase de Slow Start n'apparaît qu'en déclenchant une communication. Par conséquent, il est supposé dans le modèle que les pertes se produisent dans le réseau en raison des conditions de congestion inévitable lorsque le routeur ne dispose plus de mémoire suffisante pour les paquets qui arrivent. La dynamique de $W_i(t)$ dans (1.1) sera donc [Misra 2000] :

$$dW_i(t) = \frac{dt}{R_i(q(t))} - \frac{W_i(t)}{2} dI_{i3DupAck}(t). \tag{1.2}$$

Selon le processus de Poisson, les informations de pertes de paquets par $3DupAck$ parviennent à la source avec un taux $\lambda_i(t)$. Le calcul des espérances mathématiques des deux côtés de (1.2) en considérant $E[dI_{i3DupAck}] = \lambda_i(t)dt$ donne :

$$E[dW_i(t)] = E\left[\frac{dt}{R_i(q(t))}\right] - \frac{E[W_i(t)\lambda_i(t)dt]}{2}. \tag{1.3}$$

L'équation (1.3) sera simplifiée en supposant que la fenêtre de congestion est indépendante du processus d'éjection de paquets, d'où :

$$dE[W_i(t)] = E\left[\frac{dt}{R_i(q(t))}\right] - \frac{E[W_i(t)]\lambda_i(t)}{2} dt. \tag{1.4}$$

La source réalise l'éjection d'un paquet approximativement après un délai τ du temps d'aller-retour [Misra 2000]. Il est modélisé par le système d'équations :

$$\begin{aligned} t &= R(q(t')) + t, \\ t' &= t - \tau. \end{aligned}$$

Si $\bar{q}$ est la longueur moyenne de la file d'attente dont dépend la probabilité d'éjection d'un paquet entrant dans la file d'attente $p(t)$, le taux moyen d'éjection de paquets dans un flux TCP est donné par :

$$\lambda_i(t) = p(t-\tau)E\left[\frac{W_i(t-\tau)}{R_i(\bar{q}(t-\tau))}\right].$$

Alors, (1.4) devient :

$$dE[W_i(t)] = \frac{dt}{R_i(\bar{q}(t))} - \frac{\bar{W}_i}{2} p(t-\tau)\frac{\bar{W}_i(t-\tau)}{R_i(\bar{q}(t-\tau))} dt.$$

D'où,

$$\frac{d\bar{W}_i(t)}{dt} = \frac{1}{R_i(\bar{q}(t))} - \frac{\bar{W}_i\bar{W}_i(t-\tau)}{2R_i(\bar{q}(t-\tau))} p(t-\tau). \tag{1.5}$$

b- Dynamique de la longueur de la file d'attente :

Le comportement de la longueur de la file d'attente $q(t)$ peut être décrit par la version différentielle de l'équation de Lindley [Misra 2000], [Misra 1999] :

$$\frac{dq(t)}{dt} = -1_{q(t)}C + \sum_{i=N}^{N} \frac{W_i}{R_i(q)}.$$

Le premier terme modélise la décroissance de la longueur de la file lorsqu'elle est supérieure à zéro due au passage des paquets par le retour ($1_{q(t)} = 1$ pour $q(t) > 0$). Le deuxième terme exprime l'arrivée des paquets des N flux.
En appliquant les espérances mathématiques, nous obtenons :

$$\begin{aligned}\frac{d\bar{q}(t)}{dt} &= E[-1_{q(t)}]C + \sum_{i=N}^{N} E\left[\frac{W_i}{R_i(q)}\right] \\ &= E[-1_{q(t)}]C + \sum_{i=N}^{N} \frac{\bar{W}_i}{R_i(\bar{q})},\end{aligned}$$

Ayant considéré le routeur en phase d'évitement de congestion, nous avons $q(t) > 0$ avec une probabilité très proche de 1 ; nous aurons :

$$\frac{d\bar{q}(t)}{dt} = -C + \sum_{i=N}^{N} \frac{\bar{W}_i}{R_i(\bar{q})}. \tag{1.6}$$

c- Modèle simplifié avec N flux :

En considérant un réseau à N flux TCP, les équations représentant le comportement peuvent être simplifiées en présence des flux identiques ayant la même route et le même RTT. Ces flux vont donc avoir le même comportement moyen. Les équations (1.5) et (1.6) sont modifiées pour représenter d'une façon simplifiée et unifiée les dynamiques moyennes de la taille de la fenêtre de congestion (ou $W(t)$) et de la longueur moyenne de la file d'attente (ou $q(t)$) [Hollot 2001b] :

$$\begin{cases} \dot{W}(t) &= \frac{1}{R(t)} - \frac{W(t)W(t-R(t))}{2R(t-R(t))}p(t-R(t)), \\ \dot{q}(t) &= \frac{W(t)}{R(t)}N - C, \\ R(t) &= \frac{q(t)}{C} + T_p. \end{cases} \tag{1.7}$$

Les variations $W(t)$ et $q(t)$ sont exprimées au niveau de paquets en fonction des paramètres caractéristiques du réseau. Le temps d'aller-retour moyen de paquets ou Round Trip Time $R(t)$ est exprimé en secondes [s] et C la capacité du lien en [paquets/s].
Dans notre modèle (1.7), le signal $p(t)$ constitue l'entrée obtenue à partir des mécanismes de contrôle de la longueur de la file d'attente du routeur ou *Active Queue Management* (AQM). La longueur de la file d'attente $q(t)$ peut être mesurée à chaque instant, elle représente donc la sortie du modèle.

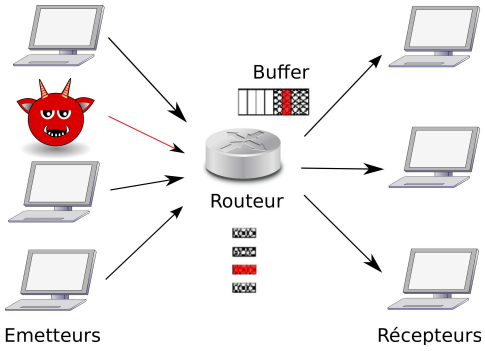


FIG. 1.12: Représentation d'une anomalie dans la topologie du modèle TCP/IP.

1.7.1 Modélisation de l'anomalie

Le système adopté pour notre travail (1.7) modélise le comportement normal du modèle TCP. Cependant, notre but est la détection et la reconstruction des anomalies perturbant le réseau en analysant le comportement de la file d'attente au niveau du routeur (Figure 1.12). Nous considérons l'anomalie comme un trafic affectant le protocole TCP. Dans (1.7), un signal $d(t)$ peut être ajouté à la dynamique de la longueur de la file $q(t)$ afin de représenter le trafic supplémentaire inconnu.

$$\dot{q}(t) = \frac{W(t)}{R(t)}N - C + \mathbf{d(t)}.$$

Ce terme $d(t)$ engendre une augmentation inattendue de la longueur de la file d'attente menant à une augmentation de la probabilité d'éjection des paquets calculée par l'AQM considéré ; donc la décroissance de la fenêtre de congestion des flux TCP émis. Toute modification du comportement du système permet de conclure sur la présence de phénomènes anormaux dans le modèle TCP/IP.
Remarquons que la dynamique adoptée pour le protocole TCP (1.7) est représentée par une équation non linéaire à retard. Vu la complexité du modèle non linéaire, une façon de simplifier sera de le linéariser autour d'un point d'équilibre. Les détails de la linéarisation sont présentés dans l'Annexe A.

A notre connaissance, l'observation des anomalies dans le modèle TCP/IP est à peine adoptée dans la littérature. Dans [Fliess 2005], une détection des défauts par la méthode de platitude est développée pour le modèle fluide non linéaire de TCP [Misra 2000]. Dans [Ariba 2009], un observateur de Luenberger classique est conçu pour un modèle étendu afin d'estimer des anomalies à débit constant.

Afin de présenter nos résultats, nous rappelons succinctement certaines notions importantes relatives aux systèmes à retards ainsi que les approches d'observation nécessaires.

1.8 Les systèmes à retards

Le retard est un phénomène intrinsèque dans la modélisation de la plupart des processus physiques. En outre, même si un processus ne contient pas de retard, des retards apparaissent souvent dans la boucle de commande du fait des délais de réaction des capteurs ou des actionneurs, des temps de transmissions des informations ou des temps de calcul.

1.8.1 Représentation des systèmes à retards

Nous nous sommes intéressés à l'étude des techniques de détection des défauts dans les systèmes linéaires invariants dans le temps (LTI) comprenant de retards. Ces systèmes peuvent être représentés sous la forme [Gu 2003], [Richard 2003], [Niculescu 2001] :

$$\begin{cases} \dot{x}(t) & = \sum_{l=1}^{q} D_l \dot{x}(t-w_l) + \sum_{i=1}^{k}(A_i x(t-h_i) + B_i u(t-h_i)) \\ & \quad + \sum_{j=1}^{r} \int_{t-\tau_j}^{t} (G_j(\theta)x(\theta) + H_j(\theta)u(\theta))d\theta \\ y(t) & = \sum_{i=0}^{k} C_i x(t-h_i) + \sum_{j=1}^{r} \int_{t-\tau_j}^{t} (N_j(\theta)x(\theta))d\theta \\ x(t_0) & = \phi. \end{cases} \tag{1.8}$$

$x(t)$ est l'état du système. $x(t_0)$ est la condition initiale qui appartient à l'ensemble des fonctions continues $\mathcal{C} : [-\delta, 0] \to \mathbb{R}^n$ avec $\delta = \max_{i,j,l}\{h_i, \tau_j, w_l\}$.
Les matrices A_i sont associées aux retards discrets h_i. La somme de l'intégrale correspond aux retards distribués, pondérés par G_j. Les matrices D_l représentent la contribution des dérivées retardées à la dynamique du système, appelé système neutre. Les matrices B_i et H_j sont les matrices de commande.
Un système neutre est un modèle où des retards apparaissent également sur le terme $\dot{x}(t)$: l'évolution du système dépend alors des variations passées de la dérivée. La partie distribuée apparaît lorsque le système dépend de l'ensemble de son passé, de façon continue durant l'intervalle de temps $[t-\tau_j, t]$. Enfin, il y a des retards discrets, pour lesquels le système évolue en fonction de son passé à des instants précis. Ce sont les retards discrets qui apparaissent dans la modélisation du protocole TCP (1.7).

L'effet du retard sur la stabilité des systèmes comprenant un espace d'état retardé et/ou entrée retardée, est un problème d'intérêt récurrent [Gu 2003], [Fridman 2006], [Richard 2003]. En effet, la présence des retards peut induire de complexes comportements (oscillations, instabilité) dans le système en boucle fermée. Un petit retard peut parfois être suffisant pour déstabiliser certains systèmes ; par contre, certains retards peuvent stabiliser un système [Richard 2003]. L'analyse de la stabilité des systèmes à retards est étudiée dans ce mémoire en utilisant une extension de la théorie de stabilité de Lyapunov au cas des systèmes à retards.

1.8.2 Stabilité selon la seconde méthode de Lyapunov

Pour la stabilité des systèmes à retards discrets, nous allons rappeler quelques résultats en nous focalisant sur les approches liées à la seconde méthode de Lyapunov.

Considérons le système général suivant :

$$\begin{cases} \dot{x}(t) &= f(t, x(t), x_t) \\ x_{t_0}(\theta) &= \phi(\theta), \qquad \text{pour } \theta \in [-\tau, 0], \end{cases} \tag{1.9}$$

où $x_t(\theta) = x(t+\theta)$. Supposons que ce système admet un seul point d'équilibre : $x_t = 0$. Si ce n'est pas le cas, un changement de variable peut ramener le point d'équilibre à l'origine. En général pour des systèmes sans retards, cette méthode consiste à développer une fonction définie positive V telle que sa dérivée soit négative le long des trajectoires du système (1.9) ($\frac{dV}{dt} < 0$ pour $x \neq 0$). Cette fonction n'est pas valable pour les systèmes à retards puisque les variations de V dépendent des valeurs passées x_t. Deux extensions ont alors été développées pour les équations différentielles à retards : la méthode de Lyapunov-Krasovskii et la méthode de Lyapunov-Razumikhin que nous allons introduire brièvement.

1.8.2.1 Fonctions de Razumikhin

Dans cette approche, nous considérons une fonction de Lyapunov $V(t, x(t))$. Toutefois le théorème suivant montre qu'il est inutile de vérifier que $\dot{V}(t, x(t)) \leq 0$ le long de toutes les trajectoires du système. Effectivement, ce théorème peut se restreindre aux solutions qui ont tendance à quitter un voisinage de $V(t, x(t)) \leq c$ du point d'équilibre [Gu 2003].

Théorème 1.1. *Soient u, v, et $w : \mathbb{R}^+ \to \mathbb{R}^+$ des fonctions croissantes. $u(\theta)$ et $v(\theta)$ sont strictement positives pour tout $\theta > 0$ avec $u(0) = v(0) = 0$. Supposons que la fonction f dans (1.9) est bornée.*
S'il existe une fonction continue $V : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^+$ telle que :

a. $u(||\phi(0)||) \leq V(t, \phi(t)) \leq v(||\phi||)$,

b. $\dot{V}(t, \phi) \leq -w(||\phi(0)||)$ *pour toutes les trajectoires de (1.9) vérifiant :*

$$V(t+\theta, \phi(t+\theta)) \leq V(t, \phi(t)), \quad \forall \theta \in [-\tau, 0], \tag{1.10}$$

alors la solution de (1.9) est uniformément stable.
De plus si $w(\theta) > 0$ pour tout $\theta > 0$ et s'il existe une fonction $p : \mathbb{R}^+ \to \mathbb{R}^+$ strictement croissante avec $p(\theta) > \theta$ pour tout $\theta > 0$ telle que les 2 conditions précédentes sont aussi vérifiées mais avec

$$V(t+\theta, x(t+\theta)) \leq p(V(t, x(t))), \quad \forall \theta \in [-\tau, 0],$$

alors la fonctionnelle V est appelée fonction de Lyapunov-Razumikhin et la solution de (1.9) est asymptotiquement stable.

Preuve. Le lecteur peut se référer à [Hale 1993] (page 151) pour la preuve du théorème. □

L'approche de Lyapunov-Razumikhin permet de prendre en compte des retards variables sans restriction sur la dérivée du retard. Pour des retards constants, l'existence d'une fonction de Lyapunov-Razumikhin entraîne celle d'une fonctionnelle de Lyapunov-Krasovskii [Driver 1977]. Cependant, la condition de stabilité de Lyapunov-Razumikhin (1.10) est difficile à traiter. De plus, les fonctionnelles de Lyapunov-Krasovskii sont plus utilisées dans la littérature puisqu'elles conduisent à des résultats moins conservatifs que les fonctions de Lyapunov-Razumikhin.

1.8.2.2 Fonctionnelles de Krasovskii

Cette méthode consiste à établir une fonctionnelle $V(t, x_t)$ décroissante le long des solutions de (1.9).

Théorème 1.2. *Soient u, v, et $w : \mathbb{R}^+ \to \mathbb{R}^+$ des fonctions continues croissantes. $u(\theta)$ et $v(\theta)$ sont strictement positives pour tout $\theta > 0$ avec $u(0) = v(0) = 0$. Supposons que f dans (1.9) est bornée.*
S'il existe une fonctionnelle continue $V : \mathbb{R} \times \mathcal{C} \to \mathbb{R}^+$ telle que :

a. $u(||\phi(0)||) \leq V(t, \phi(t)) \leq v(||\phi||)$,

b. $\dot{V}(t, \phi) \leq -w(||\phi(0)||)$ *pour tout* $t > t_0$ *le long des trajectoires de (1.9),*

alors la solution de (1.9) est uniformément stable.
De plus si $w(\theta) > 0$ pour tout $\theta > 0$, alors la solution de (1.9) est uniformément asymptotiquement stable.

Preuve. Se référer à [Hale 1993] (page 132) pour la preuve du théorème. □

Pour les systèmes linéaires à retards, il est possible de chercher une fonctionnelle de Lyapunov-Krasovskii générique complète [Hale 1993], [Niculescu 2001]. La désignation de "complète" signifie que la fonctionnelle assure les conditions nécessaires et suffisantes pour l'étude de la stabilité. Si nous considérons le système LTI suivant :

$$\begin{cases} \dot{x}(t) &= Ax(t) + A_d x(t-h), \\ x(\theta) &= \phi(\theta), \quad \theta \in [-h, 0], \end{cases} \tag{1.11}$$

où $x \in \mathbb{R}^n$ est l'état du système, $\phi \in \mathcal{C}([-h, 0])$ est la condition initiale et $h \in \mathbb{R}^+$ le retard considéré constant. La fonctionnelle de Lyapunov-Krasovskii stabilisant asymptotiquement le système (1.11) peut se mettre sous la forme :

$$\begin{aligned} V(x_t) \;=\; & x(t)^T P x(t) + 2x(t)^T \int_{-h}^{0} Q(\zeta) x(t+\zeta) d\zeta + \int_{-h}^{0} x(t+\zeta)^T S(\zeta) x(t+\zeta) d\zeta \\ & + \int_{-h}^{0} \int_{-h}^{0} x(t+\zeta)^T R(\zeta, \eta) x(t+\eta) d\eta d\zeta \;\geq\; \varepsilon ||x(t)||^2 \end{aligned} \tag{1.12}$$

où $P = P^T \in \mathbb{R}^{n \times n}$ est une matrice constante, Q, R et $S \in \mathbb{R}^{n \times n}$ sont des fonctions matricielles continuellement différentiables telles que $R(\zeta, \eta) = R(\eta, \zeta)^T$ et $S(\zeta) = S(\zeta)^T$ et ε est un scalaire positif.
L'utilisation de l'approche de Lyapunov-Krasovskii permet d'obtenir des conditions de stabilité sous forme d'Inégalités Linéaires Matricielles (LMI)[Boyd 1994]. Pour des fonctionnelles plus simples dont les matrices P, Q, R et S sont constantes [Fridman 2005], [Fridman 2006], [Gu 2003], des conditions peuvent être testées numériquement pour obtenir des matrices satisfaisantes.

Les fonctionnelles de Lyapunov-Krasovskii seront les outils de base pour l'analyse de stabilité des observateurs conçus dans ce mémoire pour la détection et reconstruction des anomalies dans le réseau TCP/IP.

1.9 Observation des défauts

1.9.1 Observation et filtrage

Les techniques d'observation et de filtrage des systèmes dynamiques assurent l'estimation des signaux non mesurables ou le rejet des perturbations dans le système considéré. Le concept de l'observateur est différent de celui du filtre. Les observateurs estiment les signaux inconnus en assurant la stabilité de l'erreur entre le modèle de l'observateur et le système. En d'autres termes, les équations différentielles de l'erreur d'observation doivent tendre vers zéro. Il est important de noter que l'observateur doit être capable d'estimer l'état du système quelque soit sa valeur. De ce fait, l'erreur d'observation doit être indépendante de l'état du système. Les observateurs les plus utilisés sont les observateurs de Luenberger [Luenberger 1971] dont les paramètres sont calculés à partir des conditions de stabilisation de l'erreur.
Par contre les filtres, pouvant avoir la même structure d'un observateur, visent à garantir une atténuation minimale de la différence entre le signal désiré et l'estimé. Par exemple, les filtres de kalman, les plus connus dans la littérature [Sorenson 1985], estiment les états d'un système dynamique en présence de mesures bruitées considérés comme des bruits blancs. Les paramètres d'observation peuvent être obtenus à partir d'une résolution de l'équation de Ricatti.
La classe des systèmes qui peuvent être traités par la théorie de l'observation n'est pas aussi large que pour le filtrage. De plus, le problème résultant de l'application des filtres est plus simple qu'avec les observateurs. Cependant, ces derniers sont plus adaptés que les filtres à la théorie de commande. Les observateurs entrent dans notre domaine d'intérêt et plus précisément les observateurs contribuant à la détection des défauts.

La détection et l'isolation des défauts (*Fault Detection and Isolation* FDI) est un domaine de recherche de grande importance. Un système FDI a pour objectifs la génération d'une alarme en cas de défaut ainsi que l'identification de sa nature et son emplacement. En principe, la sortie mesurée du système est comparée à la sortie d'un observateur FDI conçu à partir d'un modèle du système, ainsi l'écart formé appelé résidu à partir duquel une conclusion peut être faite sur la présence ou l'absence d'un défaut. Plusieurs ouvrages sont consacrés à la détection des défauts basé sur des modèles mathématiques linéaires [Chen 1999], [Gertler 1998], [Hou 1996].
Deux approches de synthèse d'observateurs peuvent être étudiées pour l'observation des défauts : les observateurs à entrées connues et les observateurs à entrées inconnues.

1.9.2 Observateurs à entrées connues

La première approche suppose connaître partiellement le comportement des défauts. D'où l'appellation d'observateurs standards (à entrées connues). Les variations des défauts peuvent être intégrées dans la représentation du système de façon à avoir un modèle augmenté de forme similaire au système initial. L'avantage de cette approche est restreint à l'observation classique des états inconnus du système. Les inconvénients résident dans la nécessité d'une connaissance préalable des défauts que nous voulons "observer", ainsi dans la limitation sur une classe déterminée de défauts.

Considérons le système modélisé par la représentation d'état :

$$\begin{cases} \dot{x}(t) &= Ax(t) + Bu(t) + Rf(t), \\ y(t) &= Cx(t), \end{cases} \tag{1.13}$$

où $f(t)$ représente le défaut dans le système et R une matrice constante. Si la dynamique de f est connue, un vecteur d'état augmenté de la dynamique de f sera considéré. Par conséquent, un observateur permettra d'estimer simultanément le vecteur d'état x et le défaut f [Friedland 1996], [Luenberger 1971].

1.9.3 Observateurs à entrées inconnues (UIO)

La deuxième approche considère les défauts complètement inconnus, ce qui mène à la synthèse d'un observateur à entrées inconnues (*Unknown Input Observer* UIO) [Hammouri 2010], [Corless 1998]. Plusieurs travaux ont été consacrés à la synthèse d'observateurs linéaires [Guan 1991], [Hou 1992], [Chen 1996], et non linéaires pour les systèmes à entrées inconnues [Edwards 2004], [Yang 1997], [Frank 1994]. Des conditions théoriques nécessaires et suffisantes pour l'existence des UIO linéaires et non linéaires ont été établies : la fonction de matrice de transfert entre l'entrée non mesurable et les sorties doit être à phase minimale et de degré relatif égal à un, des conditions de recouvrement (en anglais *matching conditions*) des observateurs doivent être aussi vérifiées.

Exemple de synthèse d'un UIO

Prenons le système (1.13). L'objectif consiste à concevoir un observateur qui s'écrit sous la forme :

$$\begin{cases} \dot{z}(t) &= Dz(t) + Hu(t) + Ey(t), \\ \hat{x}(t) &= z(t) - My(t). \end{cases} \tag{1.14}$$

Un observateur à entrées inconnues existe pour ce système si et seulement si les deux conditions de rang suivantes sont vérifiées [Guan 1991] :

$$rang(CR) = rang(R) = m$$

$$rang \begin{pmatrix} sI - A & R \\ C & 0 \end{pmatrix} = n + m \quad \forall s \in \mathbb{C},\ Re(s) \geq 0$$

En introduisant la matrice $P = I + MG$ et l'erreur d'observation $e(t) = x(t) - \hat{x}(t)$, nous obtenons :

$$\begin{aligned} \dot{\tilde{x}}(t) &= P\dot{x}(t) - \dot{z}(t), \\ &= D\tilde{x}(t) + (PA - DP - EC)x(t) + (PB - H)u(t) + PRf(t). \end{aligned} \tag{1.15}$$

L'observateur est utilisé pour générer des résidus permettant de conclure sur la présence de défauts et l'estimer. Le résidu doit converger vers zéro afin d'être insensible au défaut ou

$\lim_{t\to\infty} r(t) = 0$.

$$\begin{aligned} r(t) &= y(t) - \hat{y}(t) \\ &= Py(t) - Cz(t). \end{aligned} \tag{1.16}$$

Pour que $\lim_{t\to\infty} r(t) = 0$ pour toutes les entrées connues et inconnues, et pour tout état initial, il est nécessaire que les conditions suivantes soient satisfaites :

i. D est une matrice de Hurwitz ;

ii. $PA - DP = EC$;

iii. $H = PB$;

iv. $PR = 0$.

Les détails de calcul des matrices D, E et M sont présents dans [Guan 1991].
Pour obtenir une estimation de $f(t)$, nous pouvons utiliser la décomposition en valeurs singulières de R :

$$R = U_R \Delta_R V_R^T$$

Avec $x(t) = \tilde{x}(t)$, nous obtenons :

$$\Delta_R V_R^T f(t) = U_R^T(\dot{\hat{x}} - Az(t) + AMy(t) - Bu(t)).$$

Si P est inversible,

$$\Delta_R V_R^T f(t) = U_R^T((P^{-1}D - A)z(t) + (AM + P^{-1}E)y(t) + (P^{-1}H - B)u(t)).$$

Toutes les entrées inconnues peuvent être estimées si Δ_R est de rang plein en colonnes.

1.10 Observation des défauts pour les systèmes à retards

Comme pour les systèmes sans retards, nous pouvons classer les observateurs selon les types des défauts étudiés. Par chaque classe de défauts, nous présentons les résultats de la littérature les plus pertinents.

1.10.1 Observateurs à entrées connues

Les défauts à dynamique connue peuvent être intégrés dans le vecteur d'état du système. Donc, le problème d'observation de défauts se ramène à l'observateur d'état. Dans la littérature pour les systèmes à retards, des observateurs d'état sont conçus en faisant intervenir les retards dans leurs équations différentielles [Boutayeb 2001], [Darouach 2001]. Dans tels types d'observateurs, les retards sont considérés connus. En outre, des observateurs peuvent être construits sans retard interne [Darouach 1999], [Wang 2001] nécessitent la connaissance de la sortie du système aux temps courants et retardés du même temps que les termes retardés dans l'équation de la dynamique.

Observateurs à retard interne

Considérons le système linéaire invariant comprenant un seul retard constant dans l'état

$$\begin{cases} \dot{x}(t) &= Ax(t) + A_d x(t-h) + Bu(t), \\ y(t) &= Cx(t), \end{cases} \tag{1.17}$$

où A, A_d, B et C sont des matrices de dimensions appropriées.
Dans [Darouach 1999], un observateur est proposé pour le système (1.17) de la forme :

$$\begin{cases} \dot{z}(t) &= Fz(t) + Gz(t-h) + Dy(t) + Ey(t-h) + TBu(t), \\ \hat{x}(t) &= My(t) + Nz(t), \end{cases} \tag{1.18}$$

où les matrices F, G, D, E, T, M et N sont des matrices déterminées telles que $\hat{x}$ converge asymptotiquement vers x. Ces matrices satisfont :

$$\begin{aligned} D &= TAM, \\ F &= TAN, \\ E &= TA_dM, \\ G &= TA_dN, \end{aligned}$$

où T est déterminée partir de

$$det\left(\begin{bmatrix} C \\ T \end{bmatrix}\right) \neq 0 \quad \text{et} \quad \begin{bmatrix} C \\ T \end{bmatrix}^{-1} = [M,\ N].$$

Comme $\begin{bmatrix} C \\ T \end{bmatrix}$ est une matrice non singulière, T est obtenue à partir de l'équation

$$T = R - KC,$$

sachant que K est une matrice arbitraire et R une matrice de rang plein en lignes.
Les détails de calcul de l'observateur sont présents dans [Darouach 1999].

Observateurs sans retard interne

L'observateur sans retard est obtenu en posant $G = 0$. Cette simplification est contrainte aux conditions :

$$rang\begin{pmatrix} C \\ CA_d \end{pmatrix} = n,$$

$$rang\begin{pmatrix} sI - \bar{A} \\ \bar{C} \end{pmatrix} = n - m \quad \forall s \in \mathbb{C},\ Re(s) \geq 0,$$

où

$$\bar{A} = RAN - RA_dN(CA_dN)^+CAN,$$
$$\bar{C} = (I - (CA_dN)(CA_dN)^+)CAN,$$

et $(CA_dN)^+$ l'inverse généralisé de (CA_dN).
Les details de construction de tels observateurs sans retard interne sont proposés dans [Darouach 1999] et [Darouach 2001].

Dans ce travail de thèse, notre première contribution porte sur la synthèse d'observateurs pour les systèmes à retards soumis à une classe spécifique de défauts. Nous nous sommes intéressés à l'approche de Lyapunov-Krasovskii qui sera prise en compte dans le chapitre 2.

1.10.2 Observateurs à entrées inconnues

De récents résultats concernant la détection des défauts par les UIO ont été élaborés pour les systèmes à retards. Dans [Yang 1998], un observateur est conçu pour les systèmes à retards avec des hypothèses conservatives. Dans [Ding 2001], [Zhong 2003], [Bai 2007], la détection robuste pour des systèmes linéaires à retards connus est proposée par une optimization H_∞. Dans cette approche, les systèmes comprennent en même temps des entrées inconnues représentant les perturbations et des défauts. L'effet des entrées inconnues sur les résidus est borné pourtant la sensibilité des résidus aux défauts augmente dans un domaine de fréquences. Dans [You 2004], [Jiang 2005] des approches d'observateurs adaptatifs sont étudiées pour estimer les défauts pour les systèmes à retards, dans [Meskin 2009] une approche géométrique est proposée.

Les observateurs à modes glissants [Floquet 2007b], [Spurgeon 2008], [Kalsi 2010] sont largement utilisés dans la catégorie des UIO non linéaires pour la détection des défauts. La particularité de ces observateurs est leurs propriétés de robustesse vis-à-vis des perturbations et incertitudes [Edwards 2004], et au lieu de générer des résidus, les observateurs à modes glissants sont capables de "reconstruire" les défauts. Ceci est avantageux dans certaines circonstances où la grandeur de la faute est répliquée. Nous présentons dans la partie suivante les principes des modes glissants que nous allons adopter pour notre problématique de détection des anomalies inconnues dans le modèle TCP/IP.

1.10.3 Observateurs à modes glissants

Les modes glissants pour la commande et l'observation ont attiré l'attention des chercheurs ces dernières décennies. Les intérêts que portent ces techniques se résument par :

- leur simplicité d'élaboration ;
- leur convergence en temps fini ;
- leur robustesse vis-à-vis de certaines incertitudes paramétriques et perturbations ;
- la large gamme de domaines de leurs applications tels que la robotique, la mécanique ou l'électrotechnique.

L'idée des modes glissants est de contraindre les trajectoires du système d'atteindre après un temps fini une surface déterminée appelée "*surface de glissement*" et de maintenir la dynamique autour de cette surface [Utkin 1992], [Drakunov 1995]. Le principal avantage d'une telle stratégie est la réduction de la dimension du problème à la dynamique glissante. En effet, l'étude se réduit, dans le cas de l'observation, à l'espace des états inconnus du système qui est complémentaire aux sorties. De plus, il garantit des propriétés de robustesse par rapport à une classe de perturbations et d'incertitudes paramétriques qui satisfont les conditions de recouvrement [Utkin 1992].

C'est à partir des propriétés désirées du système que la surface de glissement est définie. Une

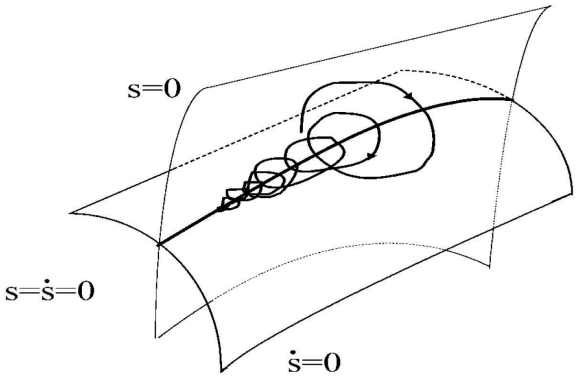


FIG. 1.13: Ensemble de glissement du second ordre.

loi de commande discontinue est ensuite synthétisée de façon à rendre la surface invariante et attractive. L'introduction de cette action discontinue agit sur la dérivée par rapport au temps d'une variable choisie comme variable de glissement. Dans le cas de l'observation, la variable de glissement est l'erreur d'observation, qui est la différence entre l'état du système et l'estimation, censée être nulle. L'ordre de la dérivée de cette variable définit donc l'ordre des modes glissants [Floquet 2000].

Ainsi soit s la surface de glissement. $\mathcal{S}_r$ est l'ensemble de glissement d'ordre r défini par les $(r_i - 1)$ premières dérivées par rapport au temps telles que :

$$\mathcal{S}_r = \left\{ x \in \mathbb{R}^n : s = \dot{s} = \ldots = s^{(r-1)} = 0 \right\}. \tag{1.19}$$

Un exemple d'un ensemble de glissement du second ordre est montré dans la Figure 1.13. Dans la pratique, ce régime glissant idéal n'existe pas étant donné que cela impliquerait que la commande puisse commuter avec une fréquence infinie. Le caractère discontinu de la commande engendre de fortes oscillations autour de la surface désignées par *réticence* (chattering en anglais) comme montré dans la Figure 1.14. La réticence s'avère être un inconvénient majeur surtout pour les modes glissants d'ordre un. Dans de telles conditions, il est difficile d'envisager des applications pratiques puisque leur implantation implique une usure relativement rapide des organes de commande du processus. Pour contourner cet obstacle, les modes glissants d'ordre supérieur ont été proposés afin de réduire ou même d'exclure tout phénomène de chattering, tout en conservant les propriétés de robustesse et de convergence en temps fini. Les applications des modes glissants d'ordre supérieur ont trouvé un succès dans les domaines de la mécanique, la robotique, les machines électriques. Les designs de commande ou d'observateurs peuvent être trouvés abondamment dans la littérature comme

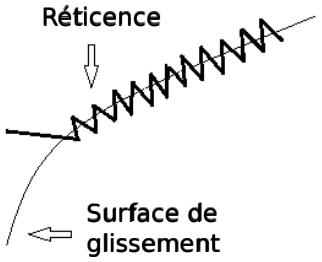


FIG. 1.14: Réticence autour de la surface de glissement.

dans [Defoort 2009], [Riachy 2008], [Pisano 2008].

Dans ce manuscrit, l'approche des modes glissants sera exploitée pour l'estimation des entrées inconnues à partir d'une méthode dite *la commande équivalente* [Utkin 1992]. Cette méthode est un moyen de déterminer le comportement du système lorsqu'il est restreint à l'ensemble de glissement $\mathcal{S}_r$ [Utkin 1992]. Une fois la stabilité de l'observateur est assurée, les défauts sont construits à partir des équations traduisant l'invariance de l'ensemble de glissement $s = \dot{s} = \ldots = s^{(r-1)} = 0$.

1.11 Objectifs

Avant de conclure ce chapitre, nous revenons aux objectifs qui ont motivé les travaux de cette thèse. Nous confrontons la problématique de détection et reconstruction des anomalies parvenant au modèle TCP/IP. Dans la théorie de commande, ces anomalies sont considérées comme des défauts dans une représentation mathématique de la dynamique du modèle TCP/IP. Théoriquement, il s'agit de concevoir un observateur adéquat pour le modèle de TCP adopté. Notre première contribution en termes de synthèse d'observateurs à entrées inconnues se limite à une classe spécifique de défauts. Nous allons présenter les anomalies sous forme polynômiale pouvant recouvrir une large gamme de profils de signaux d'anomalies dans la dynamique de TCP. Pour les entrées inconnues, notre seconde contribution se situe dans le cadre des observateurs à modes glissants pour un double objectif : l'estimation de l'état inconnu représentant la fenêtre de congestion des sources TCP, ensuite la reconstruction des anomalies parvenant au modèle TCP/IP.

Une fois la vérification des performances est achevée à l'aide des logiciels de modélisation : Matlab/Simulink et le simulateur des réseaux NS [Fall 2010], l'idéal sera de tester sur un trafic réel. L'implémentation d'un observateur sur un routeur réel est une tâche extrêmement difficile, une solution envisageable consiste à rejouer un trafic ayant des caractéristiques réelles sur un simulateur. Pour cela, nous avons besoin des outils de métrologie pour des mesures et capture de trafic TCP à partir d'une machine considérée comme routeur.

La métrologie ou la science des mesures est un outil récemment utilisé dans le domaine des réseaux pour la caractérisation et la modélisation du trafic, ainsi que pour l'analyse du comportement du trafic et du réseau, l'ingénierie, l'optimisation de la QdS et la sécurité. Les

systèmes de mesure permettent de faire une analyse en-ligne ou hors-ligne d'une trace du trafic capturé. Dans le cadre d'une analyse en-ligne, toute l'analyse doit être effectuée dans le laps de temps correspondant au passage du paquet dans la sonde de mesure. Dans une telle approche, les analyses sur de longues périodes et les statistiques restent très limitées à cause du faible temps de calcul autorisé. Une analyse hors-ligne permet à la sonde de sauvegarder une trace du trafic pour une analyse ultérieure. Cette dernière approche demande ainsi d'énormes ressources, ce qui représente une limitation pour des traces de très longue durée. Par contre, une analyse hors-ligne permet des analyses extrêmement complètes et difficiles, capables d'étudier des propriétés non triviales du trafic. L'endroit idéal pour positionner des sondes de mesures passives est indéniablement dans les routeurs.

Pour nos expérimentations réelles sur le trafic TCP, une analyse hors-ligne de la trace permet d'extraire les flux TCP ainsi que leurs caractéristiques comme le temps d'aller-retour moyen des paquets TCP, les capacités utiles des liens, les tailles des paquets TCP émis ainsi que les temps de début d'émission de chacun des flux. Ainsi, le modèle de trafic Internet est remplacé dans un simulateur par un outil de rejeu de traces de métrologie [Aussibal 2007], [Owezarski 2004]. Le simulateur NS-2 contient les fonctionnalités nécessaires au rejeu de flux TCP ayant des spécifications obtenues à partir de l'analyse hors-ligne de la trace du trafic réel extraite. Les étapes de capture du trafic, du traitement hors-ligne de la trace ensuite le rejeu des flux dans NS-2 basé sur la méthodologie dans [Owezarski 2004], seront détaillées dans le chapitre 4.

1.12 Conclusion

Au cours de ce chapitre, nous avons posé le cadre du travail de thèse. Le fonctionnement du protocole TCP a été brièvement présenté selon le modèle de couches gérant la communication entre deux machines dans le réseau Internet. Cependant, des trafics anormaux provenant des utilisations légitimes ou illégitimes peuvent perturber le réseau et même provoquer un déni de service. La recherche dans le domaine de réseaux informatiques a élaboré plusieurs outils de détection et/ou classification des anomalies. La plupart des ouvrages traite ce problème en se limitant sur des types spécifiques d'anomalies, d'où les Systèmes de Détection des Intrusions.
Notre travail se base sur la théorie de commande pour élaborer des observateurs capables de détecter et reconstruire les profils des anomalies traversant un routeur. Deux types d'observateurs seront étudiés : le premier modèle classique de Luenberger où une structure particulière de l'anomalie est prise en compte. Ce type d'observation fait l'objet du chapitre 2. Le second type d'observateur est classé dans la catégorie des observateurs à entrée inconnue où les modes glissants interviendront pour détecter et reconstruire les profils d'anomalies. La méthodologie de conception de tels observateurs sera présentée dans le chapitre 3. Des simulations sur Matlab/Simlulink sont introduites avec chacune des méthodologies proposées dans les Chapitres 1 et 2 pour une première étape de validation de la théorie. Le chapitre 4 est ensuite dédié aux simulations sous NS-2. Pour chaque type d'observateur élaboré dans les chapitres précédents, des topologies de réseaux sont étudiées en tenant en compte différents AQM informatiques (comme le RED) et ceux basés sur la théorie de commande (comme le PI et le retour d'état). De plus, des expérimentations réelles concernant le rejeu de traces

TCP sous NS-2 sont détaillées. Dans les perspectives, une architecture de supervision du modèle TCP/IP sera proposée afin de garantir une meilleure QdS en limitant l'impact des anomalies sur un routeur touché par un flux anormal.

Chapitre 2

Observation des anomalies polynômiales

Dans ce chapitre, nous proposons des méthodologies d'observation pour détecter et reconstruire les anomalies dans le protocole TCP pouvant se mettre sous une forme polynômiale. Dans le chapitre 1, après une introduction sur l'architecture du modèle TCP/IP, les types des anomalies et leurs moyens de détection existants, un modèle mathématique du comportement dynamique de TCP en phase de congestion est présenté. Les dérivées successives du signal d'anomalie sont introduites dans le vecteur d'état du système afin de les observer. Nous développons pour le système augmenté des observateurs de Luenberger suivant les deux approches de stabilisation des systèmes à retards : dépendant et indépendant du retard (DD et IOD). Nous proposons ensuite la synthèse des observateurs d'ordres réduits pour une partie de l'état comprenant les états inconnus du modèle TCP/IP : la fenêtre de congestion, l'anomalie et ses dérivées. Suivant le choix des gains d'observation, la synthèse de ces observateurs conduit à des critères IOD ou DD. La synthèse d'un observateur pour le modèle d'état complet suivant une fonctionnelle de Lyapunov-Krasovskii DD est complétée par une approche polytopique prenant en compte des incertitudes sur une plage de valeurs du retard. Pour chacun des observateurs conçus dans ce Chapitre, des simulations préliminaires sous Simulink/Matlab sont étudiées suivant l'ordre du polynôme considéré pour la présentation de l'anomalie.

2.1 Introduction

Les besoins en Qualité de Service (QdS) proposés dans l'Internet nécessitent la sécurité de transmission du trafic pour diverses applications. La supervision du réseau et plus particulièrement la détection d'anomalies représente un aspect important de la QdS. En effet, le trafic peut présenter des variations fortes, légitimes ou illégitimes. Les plus dangereuses sont les attaques par Déni de Service ou attaques par Déni de Service Distribuées (DdS ou DDdS [Lakhina 2004]) dont le but est de rendre indisponible durant une certaine période les services ou ressources d'un système informatique (un serveur web par exemple). Comme nous l'avons décrit dans le Chapitre 1, des travaux de recherche ont élaboré des techniques de détection d'anomalies ou d'intrusions (Intrusion ou Anomaly Detection System, IDS ou ADS [Hussain 2003]) capables de détecter différents types d'anomalies. Nous proposons d'aborder cette problématique à l'aide des outils de l'Automatique [Hollot 2001b], [Ariba 2009], [Fliess 2005] où le trafic TCP est observé au niveau d'un routeur touché par une anomalie. Après une première phase de modélisation, nous réalisons des observateurs pour la détection d'une classe d'anomalies.

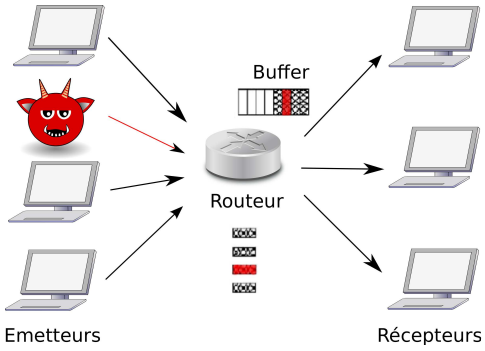


FIG. 2.1: La topologie de TCP.

2.2 Modèle dynamique de TCP

Dans la topologie TCP présentée dans la Figure 2.1, les comportements de la fenêtre de congestion $W(t)$ et de la file d'attente $q(t)$ au niveau d'un routeur sont décrits par le système (2.1). Ce système est décrit en détails dans le chapitre 1 section 1.7.

$$\begin{cases} \dot{W}(t) &= \frac{1}{R(t)} - \frac{W(t)W(t-R(t))}{2R(t-R(t))} p(t-R(t)), \\ \dot{q}(t) &= \frac{W(t)}{R(t)} N - C, \\ R(t) &= \frac{q(t)}{C} + T_p. \end{cases} \tag{2.1}$$

Le retard variable dans le modèle de TCP a pour origine : le délai de transmission des paquets qui est variable dans les liens entre deux machines ; et la durée d'attente dans le buffer du routeur qui varie selon le niveau de congestion du routeur. Pour des raisons de simplicité, le retard est supposé constant égal à sa valeur à l'équilibre R_0. Une expérimentation préliminaire sur le réseau permet d'étudier les plages de variation du temps d'aller-retour RTT des paquets TCP dans un réseau en phase d'évitement de congestion.

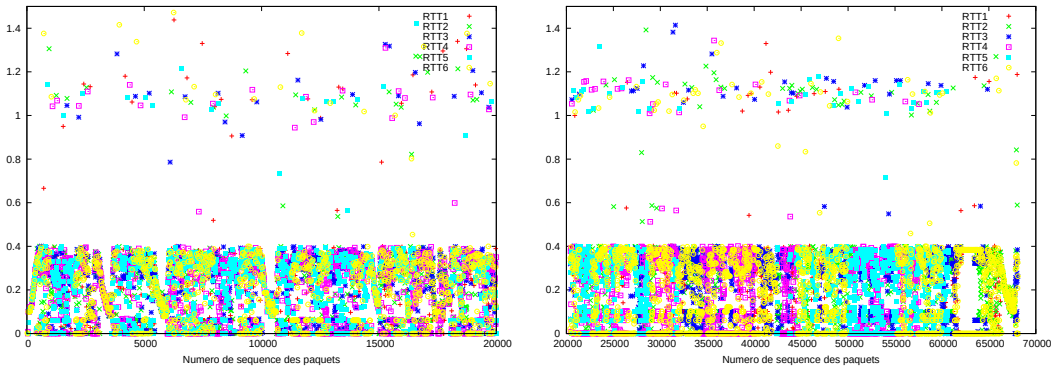
2.2.1 Etude expérimentale sur le RTT

Dans cette expérience, des traces réelles de trafic sont collectées au niveau de la *plateforme expérimentale pour l'émulation et les tests de réseaux* (LaasNetExp) [Owezarski 2008] située au laboratoire LAAS-CNRS. La plateforme sera présentée en détails dans le chapitre 4. Une machine spécifique est choisie comme routeur à partir duquel les données entrantes et les données sortantes sont captées par l'analyseur de réseau *Wireshark* [Orebaugh 2007].

La capacité du routeur est réglée afin de créer une congestion sur le routeur permettant de mieux analyser les RTT des flux TCP. Le mécanisme de *Token Bucket Filter* (TBF) est utilisé pour limiter le débit de données au niveau du routeur à 200Kbps. C'est un mécanisme de filtrage basé sur un bucket abstrait qui contient des jetons, dont chacune peut représenter un paquet de taille prédéterminée [Clark 1992]. Un flux peut être transmis jusqu'à un certain taux maximal suivant la quantité des jetons adéquate dans le bucket. Quant à l'éjection de paquets, le routeur suit le mécanisme de *Drop Tail* qui est l'algorithme de base pour la gestion de la file d'attente. Contrairement aux AQM comme RED qui différencient entre les paquets à éjecter, lorsque la file d'attente est remplie à sa capacité maximale, Drop Tail

éjecte tous les paquets nouvellement arrivés jusqu'à ce que la file d'attente ait suffisamment de place pour accepter des paquets entrants.

Durant la capture, 6 flux TCP ont été envoyés pendant 20 minutes d'une machine du LAAS vers une destination à travers le routeur. Différents flux de débits variables entre 0 et 25Kbps passaient aussi par le routeur. Nous avons choisi de charger davantage le routeur par un flux UDP de 75Kbps pour le forcer à entrer en phase de congestion. Une étude de l'impact de la congestion sur les RTT des paquets TCP est menée en les comparant avant et durant la charge UDP. Une analyse de la trace de capture permet de calculer les RTT pour chacun des flux TCP. Nous pouvons remarquer dans les graphes de la Figure 2.2 qu'avant et durant la charge UDP les RTT sont dispersés d'une manière très dense jusqu'à 0.4s.



(a) Avant la charge UDP. (b) Durant la charge UDP.

FIG. 2.2: Répartition des RTT des paquets TCP (en secondes).

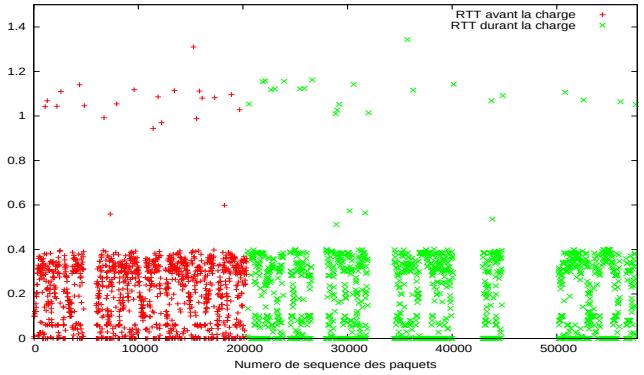


FIG. 2.3: RTT des paquets de l'un des flux (en secondes).

Nous allons étudier quantitativement pour chacun des 6 flux TCP les variations des RTT moyens avant et durant le lancement du trafic UDP. Plus spécifiquement, nous montrons le graphe des RTT pour l'un des flux dans la Figure 2.3. Nous remarquons que durant l'émission des paquets UDP, la plupart les paquets TCP sont émis avec des RTT répartis sur la même plage qu'avant. Parfois l'émission de TCP est presque interrompue où quelques paquets circulent avec de grands RTT. Ce phénomène est relié au mécanisme du TBF au niveau du

routeur qui éjecte tous les paquets une fois sa capacité maximale atteinte. Le phénomène de synchronisation est l'un des inconvénients du TBF où toutes les sources correspondant aux paquets éjectés sont obligées d'atténuer leurs taux d'émission. Pendant des intervalles de temps distincts, la source émet juste quelques paquets avec un RTT grand. Le nombre de ces paquets est à négliger devant le nombre de paquets ayant des RTT inférieurs à 0.4s. Dans le Tableau 2.1, nous présentons les valeurs moyennes des RTT, ainsi que le taux de variation des RTT en présence des paquets UDP par rapport au cas avant l'envoi des UDP.

Flux TCP	Avant UDP	Durant UDP	Taux de variation
1	191.465	202.625	+5.82%
2	188.338	199.292	+5.81%
3	194.576	203.798	+4.74%
4	196.573	199.524	+1.5%
5	190.964	189.396	−0.8%
6	194.276	198.918	+2.3%

Tab. 2.1: RTT moyens exprimés en ms pour chacun des flux.

Dans le Tableau 2.1, les RTT moyens ne présentent pas de grandes variations entre la présence et l'absence des trafics UDP. En outre dans les deux cas étudiés, l'homogénéité entre les flux est montrée vu que les paquets TCP prennent le même chemin. Pourtant, le décalage entre la valeur moyenne des RTT de chaque source et la valeur maximale (0.4s) est de 100%. Cette variation a pour origine le Drop Tail qui induit de fortes variations de la longueur de la file d'attente du routeur. Dans la théorie de contrôle, le Drop Tail représente la commande Tout ou rien. Par conséquent, avec cette commande le choix du point d'équilibre nécessaire pour notre étude sur les observateurs est délicat. Un AQM comme RED, PI ou le retour d'état permet de stabiliser la file d'attente du routeur autour du point d'équilibre. Ce qui réduit ainsi les variations du RTT autour du point d'équilibre comme ayant $R(t) = \frac{q(t)}{C} + T_p$. Pour l'élaboration des observateurs, nous considérons que le RTT est constant mais incertain entre des valeurs minimales et maximales autour de sa valeur à l'équilibre.

2.2.2 Modèle linéarisé

Le système (2.1) est non linéaire à retards. La non linéarité dépend des valeurs actuelles et retardées de la fenêtre de congestion et de l'entrée retardée. De tels systèmes étant compliqués à manipuler, nous proposons de linéariser le modèle (2.1) autour d'un point d'équilibre qui peut être obtenu à partir du système d'équations (2.2). Le calcul de la linéarisation du système sans perturbations est détaillé dans l'Annexe A.

$$\begin{cases} W_0^2 p_0 &= 2, \\ W_0 &= \frac{R_0 C}{N}, \\ R_0 &= \frac{q_0}{C} + T_p. \end{cases} \tag{2.2}$$

Le nombre de connexions N est considéré connu et constant. Dans le système (2.1), le retard RTT ou $R(t)$ apparaît dans $W(t)$, $p(t)$ et dans $R(t)$ lui-même. Pour la simplification de la procédure de linéarisation (c.f. Annexe A), le retard est considéré constant égal à sa valeur à l'équilibre. Le système linéaire autour du point d'équilibre se met alors sous la forme :

$$\begin{cases} \delta\dot{W}(t) &= -\frac{N}{R_0^2 C}(\delta W(t) + \delta W(t-h(t))) - \frac{1}{R_0^2 C}(\delta q(t) - \delta q(t-h(t))) \\ & \quad -\frac{R_0^2 C^2}{2N^2}\delta p(t-h(t)), \\ \delta\dot{q}(t) &= \frac{N}{R_0}\delta W(t) - \frac{1}{R_0}\delta q(t). \end{cases} \tag{2.3}$$

La longueur de la file d'attente $q(t)$ est mesurable au niveau du routeur, $\delta q(t)$ représente ainsi le signal de sortie. L'entrée $u(t) = \delta p(t)$ est la variation de la probabilité d'éjection des paquets.
Nous proposons d'ajouter $d(t)$ à la dynamique de $\delta q(t)$ pour représenter les anomalies perturbant le modèle TCP/IP avec $d_0 = 0$, le point d'équilibre associé à l'anomalie. En prenant le vecteur d'état comme étant $x(t) = \begin{bmatrix} \delta W(t) \\ \delta q(t) \end{bmatrix}$, la représentation d'état s'écrit comme suit :

$$\begin{cases} \dot{x}(t) &= Ax(t) + A_d x(t-h) + Bu(t-h) + B_d d(t), \\ y(t) &= Cx(t), \\ x_0(\theta) &= \phi(\theta), \text{ pour } \quad \theta \in [-h, 0]. \end{cases} \tag{2.4}$$

où

$$A = \begin{bmatrix} -\frac{N}{R_0^2 C} & -\frac{1}{R_0^2 C} \\ \frac{N}{R_0} & -\frac{1}{R_0} \end{bmatrix}, A_d = \begin{bmatrix} -\frac{N}{R_0^2 C} & \frac{1}{R_0^2 C} \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} -\frac{R_0 C^2}{2N^2} \\ 0 \end{bmatrix},$$

$$B_d = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \text{ et } C = [0 \quad 1].$$

Après la présentation du modèle linéaire (2.3), une extension du vecteur l'état est proposée afin de prendre en compte les dérivées supérieures des anomalies sous forme polynômiale.

2.2.3 Modèle augmenté

En mathématiques, et plus précisément en analyse, des fonctions peuvent être approximées par des polynômes selon le Théorème de Weierstrass [Bernstein 1912].

Théorème 2.1. *Théorème de Weierstrass :*
Supposons une fonction f définie et continue sur l'intervalle $[a, b]$. Pour tout $\varepsilon > 0$, il existe un polynôme P convergeant uniformément vers f tel que :

$$\forall x \in [a, b], \quad |f(x) - P(x)| < \varepsilon.$$

L'approximation de Bernstein est une des nombreuses méthodes d'approximation polynômiale permettant d'approcher uniformément une fonction continue définie sur l'intervalle [0, 1] par une famille de polynômes, appelés polynômes de Bernstein.

Théorème 2.2. *Bernstein-Weierstrass [Bernstein 1912] :*
En se ramenant par changement de variables à l'intervalle $[0, 1]$*, la suite* (P_n) *de polynômes approximant* f *se met sous la forme :*

$$P_n(x) = \sum_{k=0}^{n} f\left(\frac{k}{n}\right) B_n^k(x),$$

avec

$$B_n^k(x) = C_n^k x^k (1-x)^{n-k}$$

représentant les polynômes de Bernstein pour un entier n*, et* $C_n^k = \frac{n!}{k!(n-k)!}$ *le nombre de combinaisons d'un ensemble de* k *éléments parmi* n*.*

Les formes polynômiales peuvent couvrir une large classe de profils carrés ou triangulaires, donc bien adaptées aux trafics d'anomalies dans le réseau TCP/IP. Dans [Aussibal 2007], des séries d'attaques sont générées par des logiciels TFN2k modifiés et TRINOO en prenant plusieurs formes : constantes ou plates qui résultent de l'enchaînement de plusieurs flooding classiques, rampes où la puissance croît progressivement (forme de dent de scie), variables qui prennent la forme carrée avec beaucoup de "bruit", enfin les formes asynchrones qui sont des formes plates générées par plusieurs TFN2k non synchronisés. Ces formes sont illustrées dans la Figure 2.4.
Dans la représentation d'état (2.3), l'anomalie $d(t)$ peut être ajoutée avec ses dérivées suc-

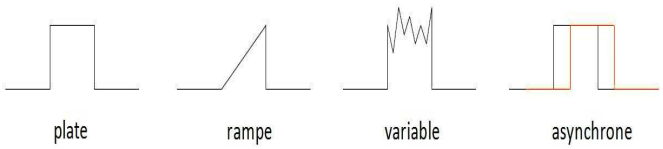


FIG. 2.4: Les formes des attaques générées par TFN modifiés.

cessives dans le vecteur d'état. Dans [Ariba 2009], un observateur à retards est conçu pour l'estimation des anomalies à débit constant en ajoutant $\dot{d}(t) = 0$ à la dynamique de TCP adoptée. Le profil constant reste un cas particulier et idéal (polynôme d'ordre 0). En tenant en compte les dérivées supérieures de l'anomalie, nous pouvons avoir plus d'informations sur la dynamique de l'anomalie. Par ailleurs, le temps de convergence de l'observateur vers les valeurs réelles de l'anomalie va diminuer par l'effet de convergence des dérivées d'ordre supérieur de l'anomalie. Les taux de faux positifs et négatifs sont de même diminués durant la détection.

Supposons que $d(t)$ est un polynôme de degré k où

$$d(t) = d_0 + d_1 t + d_2 t^2 + \ldots + d_k t^k.$$

Pour représenter l'anomalie de degré k, les (k) premières dérivées sont ajoutées au vecteur

d'état $x(t)$ telles que :

$$\begin{cases} w_1 &= d, \\ w_2 &= \dot{d} &= \dot{w}_1, \\ \vdots & \vdots & \vdots \\ w_{k+1} &= d^{(k)} &= \dot{w}_k, \\ w_{k+2} &= d^{(k+1)} &= \dot{w}_{k+1} = 0. \end{cases} \tag{2.5}$$

En considérant le vecteur d'état augmenté

$$\tilde{x}(t) = \begin{bmatrix} \delta W(t) \\ \delta q(t) \\ w_1(t) \\ \vdots \\ w_k(t) \\ w_{k+1}(t) \end{bmatrix},$$

et la sortie mesurable $\tilde{y}(t) = \delta q(t)$, nous obtenons le modèle linéaire augmenté suivant :

$$\begin{cases} \dot{\tilde{x}}(t) &= \left[\begin{array}{c|cccc} A & 0 & 0 & \dots & 0 \\ & 1 & 0 & \dots & 0 \\ \hline 0 \;\; 0 & 0 & 1 & & 0 \\ \vdots \;\; \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 \;\; 0 & 0 & 0 & \dots & 1 \\ 0 \;\; 0 & 0 & 0 & \dots & 0 \end{array}\right] \tilde{x}(t) + \left[\begin{array}{c|cccc} A_d & 0 & 0 & \dots & 0 \\ & 0 & 0 & \dots & 0 \\ \hline 0 \;\; 0 & 0 & 0 & & 0 \\ \vdots \;\; \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 \;\; 0 & 0 & 0 & \dots & 0 \\ 0 \;\; 0 & 0 & 0 & \dots & 0 \end{array}\right] \tilde{x}(t-h) \\ &\quad + \left[\begin{array}{c} B \\ \hline 0 \\ \vdots \\ 0 \\ 0 \end{array}\right] u(t-h), \\ \tilde{y}(t) &= \left[\begin{array}{c|cccc} C & 0 & \dots & 0 & 0 \end{array}\right] \tilde{x}(t) \end{cases} \tag{2.6}$$

La représentation d'état du système est alors sous la forme :

$$\begin{cases} \dot{\tilde{x}}(t) &= \tilde{A}\tilde{x}(t) + \tilde{A}_d\tilde{x}(t-h) + \tilde{B}u(t-h), \\ \tilde{y}(t) &= \tilde{C}\tilde{x}(t). \end{cases} \tag{2.7}$$

Ayant défini notre système régissant le comportement normal du modèle TCP/IP, des observateurs conçus dans la section suivante et implantés au niveau du routeur (Figure 2.5) ont comme objectifs d'estimer l'état inconnu qui est la variation de la fenêtre de congestion $\delta W(t)$, de détecter et de reconstruire les profils de perturbations $d(t)$ et leurs k dynamiques supérieures $d^{(k)}(t)$.

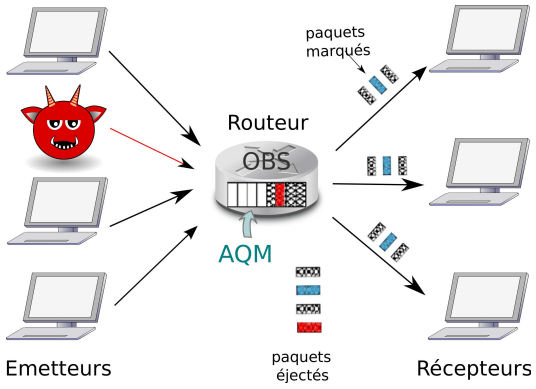


FIG. 2.5: Gestion de la file d'attente et observation du trafic au niveau du routeur.

2.3 Observateurs de Luenberger pour le modèle TCP/IP

2.3.1 Observabilité du modèle

Un système linéaire est observable, si pour une loi de commande $u(t)$ et une sortie $y(t)$ telles que $t_0 \leq t \leq T$, l'état initial $x(t_0)$ peut être déterminé. Pour un système à retards tel que (2.8)

$$\begin{cases} \dot{x}(t) &= Ax(t) + A_d x(t-h) + Bu(t-h), \\ y(t) &= Cx(t), \\ x(\theta) &= \phi(\theta), \quad \theta \in [-h, 0], \end{cases} \tag{2.8}$$

où $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$ et $\phi \in \mathcal{C}([-h, 0])$ la condition initiale, l'extension de l'observabilité est l'observabilité initiale énoncée comme ce qui suit :

Définition 1. *[Sename 2001] Pour toute entrée $u(t) = 0$ sur $[0,\ T]$ et pour toutes conditions initiales x_0 et $\phi(t)$ avec $t \in [-h, 0]$, le système (2.8) est observable si $y(t)$ n'est pas identiquement nulle sur $[0,\ T]$.*

Dans la littérature il existe plusieurs types d'observabilité pour les systèmes à retards comme l'hyperobservabilité, l'observabilité sur un anneau, la $\mathbb{R}^n$-observabilité et l'observabilité spectrale [Richard 2003]. Nous sommes concernés par étudier l'observabilité sur un anneau qui est une technique récente [Morse 1976], [Lee 1981]. Définissons un opérateur ∇ tel que $\nabla x = x(t-h)$, le système (2.8) s'écrit alors :

$$\begin{cases} \dot{x}(t) &= A(\nabla)x(t) + B(\nabla)u(t), \\ y(t) &= C(\nabla)x(t), \end{cases} \tag{2.9}$$

avec les matrices polynômiales en ∇ $A(\nabla) : \mathbb{R}^n[\nabla] \to \mathbb{R}^n[\nabla]$ et $B(\nabla) : \mathbb{R}^m[\nabla] \to \mathbb{R}^n[\nabla]$. où :

$$\begin{aligned} A(\nabla) &= A + \nabla A_d \\ B(\nabla) &= \nabla B \\ C(\nabla) &= C. \end{aligned}$$

Le système (2.9) représente un espace vectoriel sur un anneau de polynômes en ∇. La matrice d'observabilité de (2.9) s'écrit sous la forme :

$$\mathcal{O} = \begin{bmatrix} C \\ CA(\nabla) \\ \vdots \\ CA^{n-1}(\nabla) \end{bmatrix}$$

La condition d'observabilité qui garantit la reconstruction de l'état de $x(t)$ à tout instant t est désignée par *la forte observabilité*.

Définition 2. *Le système (2.9) est fortement observable sur l'anneau $\mathbb{R}^n[\nabla]$ s'il satisfait les caractéristiques suivantes :*

- $rang(\mathcal{O}) = n$,
- $Im(\mathcal{O}) = \mathbb{R}^n$.

Dans cette partie, nous allons étudier l'observabilité du modèle initial de TCP (2.3) pour conclure sur l'observabilité du modèle linéaire augmenté (2.7). Le modèle de (2.3) sur l'anneau a la même forme que (2.9) avec :

$$A(\nabla) = \begin{bmatrix} -\frac{N}{R_0^2 C} - \frac{N}{R_0^2 C}\nabla & -\frac{1}{R_0^2 C} + \frac{1}{R_0^2 C}\nabla \\ \frac{N}{R_0} & -\frac{1}{R_0} \end{bmatrix}, \tag{2.10}$$

$$B(\nabla) = \nabla B = \begin{bmatrix} -\frac{R_0 C^2}{2N^2}\nabla \\ 0 \end{bmatrix}, \tag{2.11}$$

$$C(\nabla) = [0\ 1]. \tag{2.12}$$

$$\tag{2.13}$$

Lemme 2.1. *Le modèle TCP/IP (2.3) est fortement observable sur l'anneau $\mathbb{R}^2[\nabla]$.*

Preuve. En tenant en compte les matrices (2.10), la matrice d'observabilité s'écrit sous la forme :

$$\mathcal{O} = \begin{bmatrix} C \\ CA(\nabla) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{N}{R_0} & -\frac{1}{R_0} \end{bmatrix}.$$

Le fait que $rang(\mathcal{O}) = n = 2$ et

$$Im(\mathcal{O}) = \left\{ V \in \mathbb{R}^2[\nabla] \,|\, V = \alpha_1(\nabla) \begin{bmatrix} 0 \\ \frac{N}{R_0} \end{bmatrix} + \alpha_2(\nabla) \begin{bmatrix} 1 \\ -\frac{1}{R_0} \end{bmatrix} \right\} = \mathbb{R}^2[\nabla]$$

permettent de conclure. □

En appliquant les caractéristiques sur le modèle augmenté, nous pouvons démontrer l'observabilité faible en assurant un rang plein de la matrice d'observabilité.

Lemme 2.2. *La paire* $(\tilde{A}(\nabla), \tilde{C})$ *du système augmenté* (2.7) *est faiblement observable sur* $\mathbb{R}^{k+3}[\nabla]$.

Preuve.

$$\tilde{\mathcal{O}} = \begin{bmatrix} \tilde{C} \\ \tilde{C}\tilde{A}(\nabla) \\ \vdots \\ \tilde{C}\tilde{A}^{k+2}(\nabla) \end{bmatrix} = \left[\begin{array}{c|cccc} \begin{bmatrix} 0 & 1 \end{bmatrix} & 0 & 0 & \dots & 0 \\ CA(\nabla) & 1 & 0 & \dots & 0 \\ CA^2(\nabla) & - & 1 & \dots & 0 \\ \vdots & & & \ddots & \\ CA^{k+1}(\nabla) & - & - & & 1 \\ CA^{k+2}(\nabla) & - & - & - & - \end{array}\right].$$

Il faut démontrer que les $(k+2)$ colonnes sont linéairement indépendantes, en d'autres termes, si toute combinaison linéaire nulle des colonnes a nécessairement des coefficients tous nuls. Pour $i = 1, \dots, k+3$, $\beta_1 Col_1 + ... + \beta_i Col_i + ... + \beta_{k+3} Col_{k+3} = 0$ implique que les scalaires $\beta_i = 0$.
En effet nous allons étudier la combinaison linéaire sur la matrice $\tilde{\mathcal{O}}$. En appliquant la combinaison nulle sur la première, nous trouvons $\beta_2 = 0$. Ensuite, si nous prenons les deuxième et troisième lignes des 3 premières colonnes, nous obtenons :

$$\begin{array}{l} \beta_1 \dfrac{N}{R_0} + \beta_3 \\ \beta_1 \left(-\dfrac{N^2}{R_0^3 C} - \dfrac{N^2}{R_0^3 C}\nabla - \dfrac{N}{R_0^2} \right) - \beta_3 \left(\dfrac{1}{R_0} \right) \end{array}$$

Ce qui donne $\beta_1 = 0$ et $\beta_3 = 0$. Puisque les dernières $(k+2)$ colonnes de la matrice d'observabilité forment une matrice échelonnée en ligne :

$$\underbrace{\left.\begin{array}{cccc} 1 & 0 & \dots & 0 \\ - & 1 & \dots & 0 \\ \vdots & & \ddots & \\ - & - & & 1 \\ - & - & - & - \end{array}\right\}}_{(k+2)colonnes} (k+3)lignes$$

nous obtenons $\beta_4 = \dots = \beta_{k+3} = 0$. Ce qui vérifie l'indépendance des $(k+3)$ colonnes et la matrice complète est de rang complet $(rang(\mathcal{O}) = k+3, \quad \forall\, k > 0)$.
Afin d'assurer l'observabilité forte, la deuxième condition $Im(\tilde{\mathcal{O}}) = \mathbb{R}^{k+3}$ n'est pas garantie vu que les termes dans $\tilde{\mathcal{O}}$ dépendent de ∇^{k+1}, les coefficients α_1 et α_2 dépendent donc de l'inverse de ∇^{k+1} qui n'est pas défini sur l'anneau. □

2.3.2 Synthèse de l'observateur

Nous considérons un observateur de Luenberger qui prend la même forme que le système augmenté (2.7). Le terme $\tilde{y}(t)-\hat{y}(t)$ ajouté représente l'erreur sur la sortie mesurée pondérée du gain de l'observateur noté L.

$$\dot{\hat{x}}(t) = \tilde{A}\hat{x}(t) + \tilde{A}_d\hat{x}(t-h) + \tilde{B}u(t-h) + L(\tilde{y}(t) - \hat{y}(t)) \tag{2.14}$$

L'erreur d'observation est définie comme étant la différence entre l'état du système $\tilde{x}(t)$ et celui de l'observateur $\hat{x}(t)$ telle que

$$\varepsilon(t) = \hat{x}(t) - \tilde{x}(t)$$

En respectant la dynamique du système (2.7), l'erreur est représentée par l'équation différentielle à retards :

$$\dot{\varepsilon}(t) = (\tilde{A} - L\tilde{C})\varepsilon(t) + \tilde{A}_d\varepsilon(t-h) \tag{2.15}$$

L'approche de Lyapunov-Krasovskii est très souvent utilisée pour prouver la stabilité d'un système à retards tel que (2.15). La principale difficulté reste néanmoins de trouver une fonctionnelle qui vérifie les conditions de stabilité (Théorème (1.2) du chapitre 1, en particulier les conditions de négativité de la dérivée. Dans l'approche de Lyapunov-Krasovskii, nous pouvons dégager deux approches de stabilisation [Gu 2003] [Niculescu 2001]. La première méthode, appelée IOD (*Independent Of Delay*) consiste à vérifier la stabilité du système quelque soit le retard. La seconde méthode, par contre, propose d'étudier la stabilité du système en fonction du retard h, c'est l'approche DD (*Delay Dependent*).

2.3.3 Approche de synthèse IOD

La stabilité de l'erreur d'observation (2.15) est proposée dans le théorème suivant.

Théorème 2.3. *Le système (2.15) est asymptotiquement stable $\forall h \geq 0$ s'il existe des matrices symétriques $P > 0$, $Q > 0$ et W telles que l'Inégalité Matricielle Linéaire (LMI)*

$$\begin{bmatrix} \tilde{A}^T P + P\tilde{A} + Q - W\tilde{C} - \tilde{C}^T W^T & P\tilde{A}_d \\ \tilde{A}_d^T P & -Q \end{bmatrix} < 0 \tag{2.16}$$

soit satisfaite.
Le gain de l'observateur est obtenu à partir de la matrice W tel que : $L = P^{-1}W$.

Preuve. Choisissons une fonctionnelle de Lyapunov-Krasovskii de la forme :

$$V(t) = x^T(t)Px(t) + \int_{-h}^{0} x^T(s)Qx(s)ds.$$

où P et Q sont des matrices symétriques définies positives assurant la définie positivité de la fonctionnelle de Lyapunov. La dérivée de V le long des trajectoires de (2.15) mène à :

$$\begin{aligned} \dot{V}(t) = \; & \dot{\varepsilon}^T(t)P\varepsilon(t) + \varepsilon^T(t)P\dot{\varepsilon}(t) + \varepsilon^T(t)P\tilde{A}_d\varepsilon(t-h) + \varepsilon^T(t-h)\tilde{A}_d^T P\varepsilon(t) \\ & + \varepsilon^T(t)Q\varepsilon(t) - \varepsilon^T(t-h)Q\varepsilon(t-h) \end{aligned}$$

Posons $\xi = \begin{bmatrix} \varepsilon^T(t) & \varepsilon^T(t-h) \end{bmatrix}^T$,

$$\dot{V}(t) = \xi^T \begin{bmatrix} \tilde{A}^T P + P\tilde{A} - PL\tilde{C} - \tilde{C}^T L^T P + Q & P\tilde{A}_d \\ \tilde{A}_d^T P & -Q \end{bmatrix} \xi.$$

Posons $W = PL$, nous obtenons la LMI du Théorème 2.3.
Ainsi si la LMI est négative, il existe un $\delta > 0$ tel que la LMI $< -\delta\mathbb{I}$. Alors $\dot{V}(t) < -\xi^T\xi$ montre la stabilité du système augmenté (2.15). Nous prouvons que l'erreur converge vers zéro, ou en d'autres termes, $\varepsilon(t) \to 0$ lorsque $t \to \infty$. □

Exemple d'application

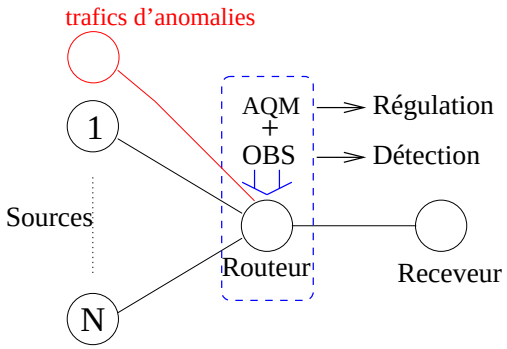


FIG. 2.6: La topologie de validation.

Dans la topologie représentée par la Figure 2.6, 60 sources TCP émettent des paquets vers la destination via un routeur avec une capacité de lien $C = 3750$ paquets/s équivalente à 15Mbps telle que la taille moyenne des paquets est de 500 octets, et $Tp = 0.2$s le délai de propagation.
Le modèle linéaire à retards (2.7) de TCP est élaboré autour des valeurs à l'équilibre $W_0 = 15$ paquets et $q_0 = 175$ paquets, $R_0 = 0.246$s. La résolution de la LMI (2.16) sous le logiciel Matlab ne montre pas la stabilité du système (2.15).
L'approche Lyapunov-Krasovskii n'étant pas applicable pour l'observation du modèle TCP augmenté indépendamment du retard, nous proposons de reconstruire uniquement l'état inconnu en partitionnant l'état en une partie connue et autre inconnue. En effet dans le modèle (2.7), la variation de la longueur de la file d'attente $\delta q(t)$ est mesurée à chaque instant au niveau du routeur. La conception d'un observateur minimal est traitée dans la partie suivante.

2.3.4 Observateur minimal

Nous allons considérer deux sous-ensembles du vecteur d'état augmenté $\tilde{x}$ dans (2.7) $x_1(t) = y(t) \in \mathbb{R}$ qui représente la sortie mesurable égale à $\delta q(t)$ et $\tilde{x}_2(t) \in \mathbb{R}^{(k+2)\times(k+2)}$ qui comprend la variation de la fenêtre de congestion $\delta W(t)$, l'anomalie $d(t)$ et ses k dérivées.

Le système original (2.7) peut alors s'écrire :

$$\begin{cases} \dot{x}_1(t) &= \tilde{A}_{11}x_1(t) + \tilde{A}_{12}\tilde{x}_2(t), \\ \dot{\tilde{x}}_2(t) &= \tilde{A}_{21}x_1(t) + \tilde{A}_{22}\tilde{x}_2(t) + \tilde{A}_{d_1}x_1(t-h) + \tilde{A}_{d_2}\tilde{x}_2(t-h) + \tilde{B}u(t-h), \\ y(t) &= x_1(t), \end{cases} \tag{2.17}$$

avec

$$\tilde{A}_{11} = -\frac{1}{R_0}, \quad \tilde{A}_{12} = [\ \tfrac{1}{R_0} \ \ 1 \ \ 0 \ \ \dots \ \ 0\ ],$$

$$\tilde{A}_{21} = \begin{bmatrix} -\frac{1}{R_0^2 C} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tilde{A}_{22} = \begin{bmatrix} -\frac{N}{R_0^2 C} & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & & 0 \\ \vdots & \vdots & & \ddots & \\ & & & & 1 \\ 0 & 0 & \dots & & 0 \end{bmatrix},$$

$$\tilde{A}_{d_1} = \begin{bmatrix} \frac{1}{R_0^2 C} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tilde{A}_{d_2} = \begin{bmatrix} -\frac{N}{R_0^2 C} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}.$$

Pour mettre en valeur les dynamiques de l'état inconnu, nous posons la nouvelle commande $\bar{u}$ telle que :

$$\bar{u} = \tilde{A}_{21}x_1(t) + \tilde{A}_{d_1}x_1(t-h) + \tilde{B}u(t-h)$$

et la nouvelle sortie $z(t)$ qui dépendent de l'état mesurable :

$$z(t) = \dot{x}_1(t) - \tilde{A}_{11}x_1(t) = \tilde{A}_{12}\tilde{x}_2(t).$$

Nous obtenons un système d'ordre réduit :

$$\begin{cases} \dot{\tilde{x}}_2(t) &= \tilde{A}_{22}\tilde{x}_2(t) + \tilde{A}_{d_2}\tilde{x}_2(t-h) + \bar{u}, \\ z(t) &= \tilde{A}_{12}\tilde{x}_2(t), \end{cases} \tag{2.18}$$

Puisque le système initial $(\tilde{A}, \tilde{C})$ est observable, le système réduit $(\tilde{A}_{22}, \tilde{A}_{12})$ est observable. Nous proposons un observateur de Luenberger pour le système (2.18) de la forme :

$$\begin{cases} \dot{\hat{x}}_2(t) &= \tilde{A}_{22}\hat{x}_2(t) + \tilde{A}_{d_2}\hat{x}_2(t-h) + \bar{u} + L(z(t) - \hat{z}(t)), \\ \hat{z}(t) &= \tilde{A}_{12}\hat{x}_2(t), \end{cases} \tag{2.19}$$

La dynamique de l'erreur d'observation définie par $\varepsilon(t) = \hat{x}_2(t) - \tilde{x}_2(t)$ est représentée par :

$$\dot{\varepsilon}(t) = (\tilde{A}_{22} - L\tilde{A}_{12})\varepsilon(t) + \tilde{A}_{d_2}\varepsilon(t-h) \tag{2.20}$$

Théorème 2.4. *Soit $L = [l_1, l_2, \ldots, l_{k+2}]^T$ la matrice de gain de l'observateur (2.19). L'erreur d'observation (2.20) est stable $\forall h > 0$ si $l_1 = 0$ et s'il existe des gains l_i, $i = 2, \ldots, k+2$ tels que les valeurs propres $(\lambda_i,\ i = 1, \ldots, k+2)$ solutions de :*

$$\sum_{i=2}^{k+2} l_i s^{k-i+2} + s^{k+1} = 0 \tag{2.21}$$

aient des parties réelles négatives.

Preuve. La stabilité de (2.20) est étudiée directement de l'équation caractéristique. Le gain de l'observateur L est calculé afin que les valeurs propres $(\lambda_i,\ i = 1, \ldots, k+2)$ de $(\tilde{A}_{22} - L\tilde{A}_{12} + \tilde{A}_{d_2} e^{(-hs)})$ aient des parties réelles négatives. Ces derniers sont calculés à partir du déterminant tel que :

$$Det(sI - (\tilde{A}_{22} - L\tilde{A}_{12}) - \tilde{A}_{d_2} e^{(-hs)}) = 0. \tag{2.22}$$

Nous obtenons l'équation caractéristique quasi-polynômiale en fonction de l'ordre k pris pour l'anomalie et la matrice du gain de l'observateur $L = [l_1, l_2, \ldots, l_{k+2}]^T$.

$$D(s, L) = det\left(\begin{bmatrix} s + \frac{N}{R_0^2 C} + l_1 \frac{N}{R_0} + \frac{N}{R_0^2 C} e^{(-hs)} & l_1 & 0 & 0 & \ldots & 0 \\ l_2 \frac{N}{R_0} & s + l_2 & -1 & 0 & \ldots & 0 \\ l_3 \frac{N}{R_0} & l_3 & s & -1 & & 0 \\ \vdots & \vdots & 0 & \ddots & \ddots & \\ \vdots & \vdots & \vdots & & s & -1 \\ l_{k+2} \frac{N}{R_0} & l_{k+2} & 0 & & & s \end{bmatrix}\right) = 0$$

Le calcul du déterminant donne :

$$\begin{aligned} D(s, L) = & \left(s + \frac{N}{R_0^2 C} + \frac{N}{R_0^2 C} e^{(-hs)}\right) \left(\sum_{i=2}^{k+2} l_i s^{k-i+2}\right) \\ & + \left(s + \frac{N}{R_0^2 C} + \frac{N}{R_0^2 C} e^{(-hs)} + l_1 \frac{N}{R_0}\right) s^{k+1} = 0. \end{aligned} \tag{2.23}$$

Comme le terme $e^{(-hs)}$ intervient dans l'équation 2.23, le calcul analytique des pôles aux parties réelles négatives en fonction de h et des gains l_i est compliqué. Nous proposons d'annuler le gain l_1 pour alléger le problème afin d'avoir $D(s, L)$ factorisé sous la forme :

$$D(s, L) = \left(s + \frac{N}{R_0^2 C} + \frac{N}{R_0^2 C} e^{(-hs)}\right) \left(\sum_{i=2}^{k+2} l_i s^{k-i+2} + s^{k+1}\right) = 0$$

Le retard apparaît uniquement dans le premier facteur qui se met sous la forme $a(s, e^{-hs}) = a_0(s) + a_1(s)e^{-hs} = \left(s + \frac{N}{R_0^2 C} + \frac{N}{R_0^2 C} e^{(-hs)}\right)$ et les gains d'observation $(l_i,\ i = 2, \ldots, k+2)$ interviennent uniquement dans le second facteur en fonction des pôles. Il est montré dans [Gu 2003] que pour un système dont l'équation caractéristique quasi-polynômiale est de la

forme de $a(s, e^{-hs})$, la marge de retard $\bar{h}$ ou la plus petite déviation de h de zéro telle que le système devient instable peut être déterminée par :

$$\bar{h} := min\{h \geq 0 \mid a(jw, e^{-jhw}) = 0 \text{ pour } w \in \mathbb{R}\}$$

Le fait que

$$\begin{cases} \left|\frac{a_1(jw)}{a_0(jw)}\right| < 1 & \forall w > 0 \\ \\ \left|\frac{a_1(jw)}{a_0(jw)}\right| = 1 & \text{pour } w = 0 \end{cases}$$

garantit que $\bar{h}$ fini n'existe pas tel que $|a(jw, e^{-jhw})| = 0$ est instable. Le système (2.20) est par conséquent stable indépendamment du retard h. □

Remarque 1. La résolution analytique de l'équation (2.23) reste en perspective. Une méthode envisageable est l'application du théorème du petit gain en considérant des petites valeurs de l_1.

Dans ce chapitre, les performances de l'observateur minimal (2.19) à reconstruire les anomalies suivant le degré de sa forme polynômiale sont testées sous Simulink.

2.3.5 Validation sous Simulink

Comme dans la Figure 2.6, la topologie de TCP comprend 60 sources TCP émettant des paquets via un routeur ayant une capacité de lien $C = 3750$ paquets/s, et considérons le délai de propagation $Tp = 0.2$s.
Pour le contrôle de congestion du routeur, plusieurs mécanismes d'AQM peuvent être simulés. Pour Simulink, nous avons choisi le mécanisme RED qui est l'un des AQM les plus utilisés [Floyd 1993] pour régler la fenêtre de congestion et la longueur de la file d'attente du routeur autour des valeurs à l'équilibre : $W_0 = 15$ paquets et $q_0 = 175$ paquets. D'autres AQM seront présentés dans le chapitre 4 dédié aux simulations sous NS-2.

Modélisation de RED Dans [Hollot 2002], RED est proposé pour la stabilisation de la file d'attente du routeur modélisé par le modèle fluide [Misra 2000]. RED calcule la probabilité de marquage d'un paquet en fonction de longueur moyenne de la file d'attente mesurée. Spécifiquement, il consiste en un filtre passe-bas et un profil de marquage de paquet comme présenté dans la Figure 2.7.
Les paramètres de RED dépendent du pôle du filtre passe-bas K_{RED}, du niveau p_{max} et du gain L_{RED}. En considérant la partie linéaire du profil de la probabilité d'éjection des paquets, RED sera modélisé par une fonction de transfert $\frac{K_{RED}L_{RED}}{s+K_{RED}}$ où $L_{RED} = \frac{p_{max}}{Max_{th}-Min_{th}}$.
Ces paramètres sont déterminés en utilisant les techniques de contrôle classiques. Dans [Hollot 2002], le filtre passe-bas est conçu de façon à ce que la dynamique du système formé de TCP et de RED domine celle de TCP. De ce fait, la fréquence de coupure w_g du système bouclé par RED est calculée de sorte qu'elle soit inférieure aux fréquences de coupure relatives à TCP telle que :

$$w_g \ll min\left\{\frac{2N}{R_0^2 C}, \frac{1}{R_0}\right\}. \tag{2.24}$$

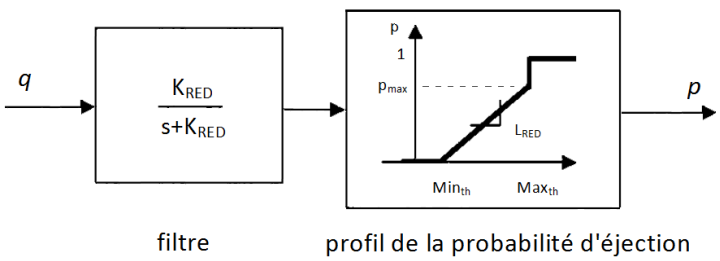


FIG. 2.7: Modélisation du RED.

La stabilité en boucle fermée formée est assurée en fixant un triplet (w_g, L_{RED}, K_{RED}) qui satisfait (2.24). Pour assurer la stabilité du modèle TCP avec RED, les conditions de positivité des marges de gain et de phase en w_g sont traduites par :

$$\left|\frac{K_{RED}L_{RED}(R_0C)^3}{(2N)^2(jw_g+K_{RED})}\right| = 1; \quad -w_gR_0 - \arctan\frac{w_g}{K_{RED}} + \pi > 0.$$

Associé à l'AQM, l'observateur (2.19) noté (OBS) dans la Figure 2.6 doit estimer le vecteur d'état comprenant la fenêtre de congestion des sources, le trafic d'anomalie $d(t)$ supposé polynômial d'ordre prédéfini et ses (k) dérivées par rapport au temps. Les gains l_i sont placés suivant le pôle dominant solution de $\left(s + \frac{N}{R_0^2C} + \frac{N}{R_0^2C}e^{(-hs)} = 0\right)$. Nous nous appuyons sur le logiciel trace-DDE [Breda 2009] pour étudier le placement des pôles des systèmes à retards. Pour la simplification des mesures, un nombre limité de pôles est montré dans la Figure 2.8. Le pôle dominant est $(\lambda = -0.6)$ puisqu'il est le plus proche de l'axe des ordonnées. Ce pôle

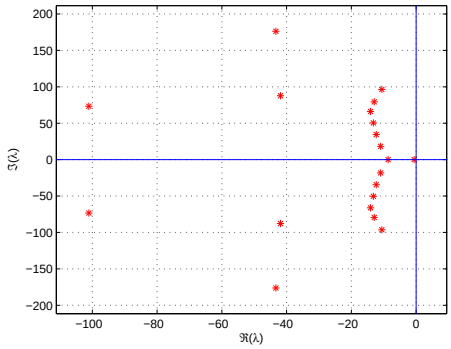
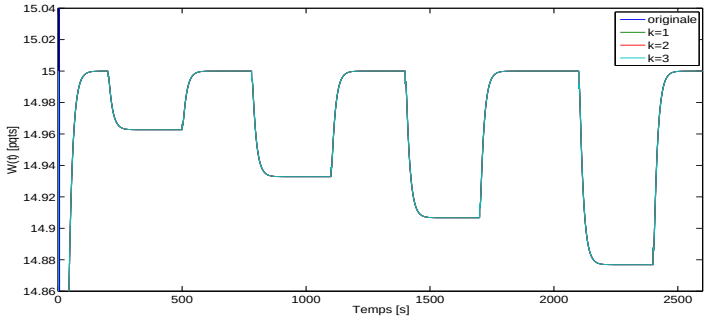
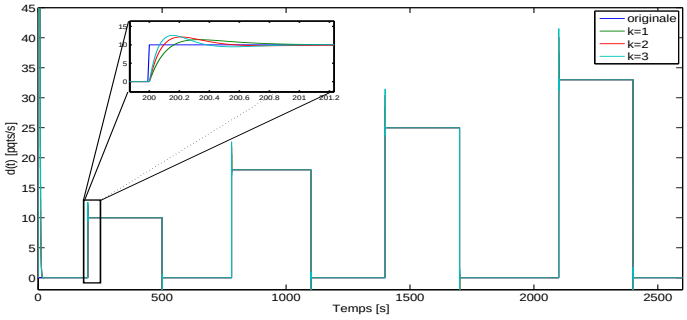


FIG. 2.8: Placement de pôles.

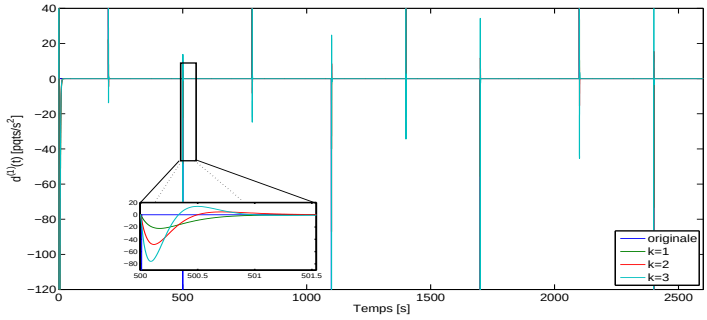
impose le temps de réponse de $3 \times \frac{1}{\lambda} = 5$ secondes. Suivant la valeur de k et en prenant des pôles λ_i plus petits que -0.6 pour garantir la stabilité et un temps de réponse ne dépassant pas les 5 secondes, les gains l_i sont calculés à partir de $\sum_{i=2}^{k+2} l_i s^{k-i+2} + s^{k+1} = 0$.
Nous considérons dans nos simulations des anomalies périodiques ayant différentes amplitudes et de débit constant ou en rampes de durée de 5 minutes. Suivant le degré considéré pour représenter l'anomalie $(k = 1, 2, 3)$, les estimations sont étudiées. Dans les graphes présentés dans les Figures 2.9 et 2.10, la fenêtre de congestion $W(t)$ est estimée instantanément



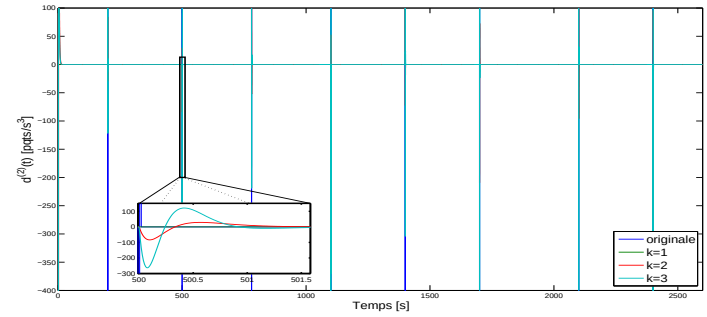
(a) La fenêtre de congestion $W(t)$.



(b) L'anomalie $d(t)$.

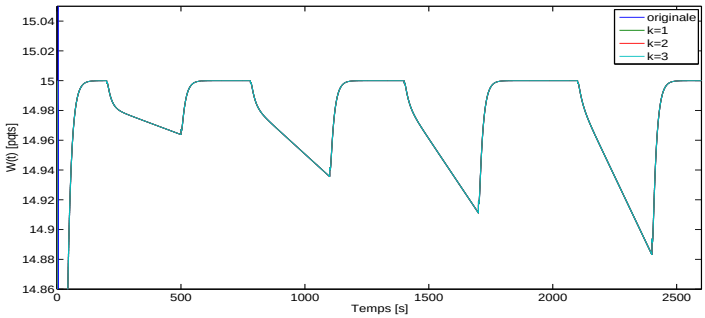


(c) La dérivée première de l'anomalie $\dot{d}(t)$.

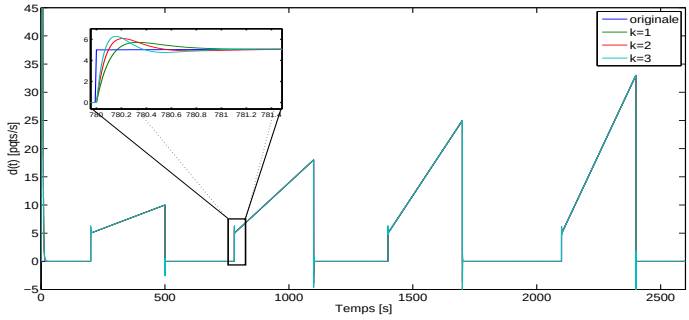


(d) La dérivée seconde de l'anomalie $\ddot{d}(t)$.

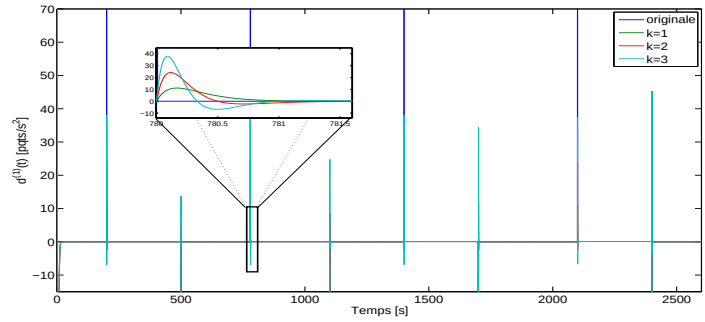
FIG. 2.9: Estimations par des observateurs minimaux avec $d(t)$ constante.



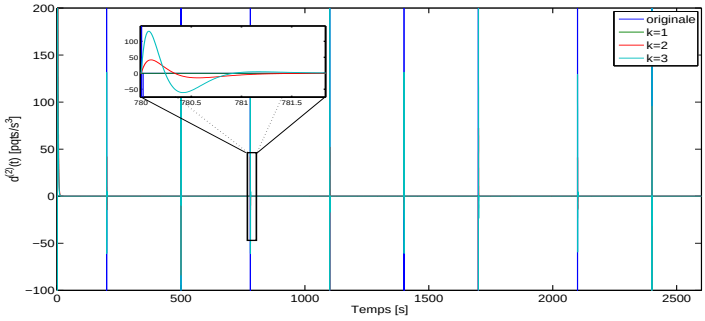
(a) La fenêtre de congestion $W(t)$.



(b) L'anomalie $d(t)$.



(c) La dérivée première de l'anomalie $\dot{d}(t)$.



(d) La dérivée seconde de l'anomalie $\ddot{d}(t)$.

FIG. 2.10: Estimations par des observateurs minimaux avec $d(t)$ triangulaire.

et uniformément entre les différentes valeurs prises pour k. Pour les profils rectangulaires et triangulaires, $d(t)$ et ses dérivées première et seconde sont bien estimées avec un temps de convergence vers la valeur originale n'atteignant pas 1.5s. Lors des apparitions et disparitions des anomalies, l'observateur agit instantanément, les taux des faux positifs et négatifs sont donc négligeables.

Malgré les bonnes performances de l'observateur basé sur l'approche IOD, la stabilisation d'un système à retards quelque soient les valeurs de retards est une condition très restrictive. De ce fait, les méthodes IOD sont généralement insuffisantes. Par conséquent, l'idée de l'approche DD est d'imposer une condition sur la taille du retard afin d'ajouter plus de degrés de liberté aux critères.

2.3.6 Approche DD

La stabilité de l'erreur d'observation (2.15) dépendant du retard est montrée dans le théorème suivant.

Théorème 2.5. *Ayant un scalaire ν et pour un retard constant $\bar{h}$, s'il existe deux matrices définies positives P et $Q \in \mathbb{R}^{(k+3)\times(k+3)}$ et une matrice $S \in \mathbb{R}^{(k+3)\times 1}$ telles que la LMI suivante soit satisfaite*

$$\begin{bmatrix} \Sigma_{11} & \Sigma_{12} & \Sigma_{13} \\ \Sigma_{12}^T & \Sigma_{22} & 0 \\ \Sigma_{13}^T & 0 & \Sigma_{33} \end{bmatrix} < 0, \tag{2.25}$$

où

$$\begin{aligned}
\Sigma_{11} &= \tilde{A}^T P + P\tilde{A} - C^T S^T - S\tilde{C} + Q - \frac{\nu}{\bar{h}} P, \\
\Sigma_{12} &= P\tilde{A}_d + \bar{h}\nu(P\tilde{A} - S\tilde{C})^T \tilde{A}_d + \frac{\nu}{\bar{h}} P, \\
\Sigma_{13} &= (P\tilde{A} - S\tilde{C})^T, \\
\Sigma_{22} &= -Q + \bar{h}\nu \tilde{A}_d^T P \tilde{A}_d - \frac{\nu}{\bar{h}} P, \\
\Sigma_{33} &= -\frac{1}{\nu\bar{h}} P,
\end{aligned}$$

alors le système (2.15) est asymptotiquement stable pour toutes les valeurs de $h < \bar{h}$. Le gain de l'observateur est défini par : $L = P^{-1}S$.

Preuve. La stabilité du système (2.15) est prouvée à partir d'une fonctionnelle de Lyapunov-Krasovskii construite pour l'étude de la stabilité des systèmes dépendamment du retard :

$$V(t) = x^T(t)Px(t) + \int_{t-h}^{t} x^T(s)Qx(s)ds + \int_{t-h}^{t}\int_{s}^{t} \dot{x}^T(v)R\dot{x}(v)dvds.$$

La dérivée de V le long des trajectoires de (2.15) donne :

$$\begin{aligned}
\dot{V}(t) = \; & 2\varepsilon^T(t)P\varepsilon(t) + \varepsilon^T(t)Q\varepsilon(t) - \varepsilon^T(t-h)Q\varepsilon(t-h) \\
& + h\dot{\varepsilon}^T(t)R\dot{\varepsilon}(t) - \int_{t-h}^{t} \dot{\varepsilon}^T(s)R\dot{\varepsilon}(s)ds.
\end{aligned} \tag{2.26}$$

Par application de l'inégalité de Jensen (Annexe B.2) qui se traduit par :

$$-\int_{t-h}^{t} \dot{\varepsilon}^T(s) R \dot{\varepsilon}(s) ds \leq -\frac{1}{h}(\varepsilon(t) - \varepsilon(t-h))^T R(\varepsilon(t) - \varepsilon(t-h)),$$

l'équation (2.26) sera bornée comme suit :

$$\dot{V}(t) \leq \xi^T(t) \begin{bmatrix} \Pi_{11} & \Pi_{12} \\ \Pi_{12}^T & \Pi_{22} \end{bmatrix} \xi(t), \tag{2.27}$$

avec $\xi(t) = \begin{bmatrix} \varepsilon(t) \\ \varepsilon(t-h) \end{bmatrix}$ et

$$\begin{aligned}
\Pi_{11} &= (\tilde{A} - L\tilde{C})^T P + P(\tilde{A} - L\tilde{C}) + Q - \frac{1}{h}R + h(\tilde{A} - L\tilde{C})^T R(\tilde{A} - L\tilde{C}), \\
\Pi_{12} &= P\tilde{A}_d + \frac{1}{h}R + h(\tilde{A} - L\tilde{C})^T R\tilde{A}_d, \\
\Pi_{22} &= -Q - \frac{1}{h}R + h\tilde{A}_d^T R\tilde{A}_d.
\end{aligned}$$

Nous remarquons que l'inégalité matricielle précédente n'est pas linéaire en les paramètres inconnus du fait de la présence des termes non linéaires tels que PL, LR et L^TRL. Pour linéariser cette inégalité, une solution conservative est de considérer $R = \nu P$ pour ν un scalaire constant et, en appliquant le changement de variable $S = PL$, nous aurons :

$$\begin{aligned}
\Pi_{11} &= (\tilde{A} - L\tilde{C})^T P + P(\tilde{A} - L\tilde{C}) + Q - \frac{\nu}{h}P + h\nu(P\tilde{A} - S\tilde{C})^T P^{-1}(P\tilde{A} - S\tilde{C}), \\
\Pi_{12} &= P\tilde{A}_d + \frac{\nu}{h}P + h\nu(P\tilde{A} - S\tilde{C})^T \tilde{A}_d, \\
\Pi_{22} &= -Q - \frac{\nu}{h}P + h\nu\tilde{A}_d^T P\tilde{A}_d.
\end{aligned}$$

Le système (2.15) sera asymptotiquement stable si

$$\begin{bmatrix} \Pi_{11} & \Pi_{12} \\ \Pi_{12}^T & \Pi_{22} \end{bmatrix} < 0.$$

Enfin, le complément de Schur (c.f. Annexe B.1) permet de transformer cette dernière LMI en (2.25) dans le Théorème 2.5. □

Lors de la représentation sous forme d'espace d'état (2.7), R_0 était supposé constant. Cependant, cette hypothèse ne reflète pas la réalité du fait que $R(t)$ prend en compte d'un côté le temps de propagation entre les machines qui varie continuellement avec l'état du trafic dans le réseau, et de l'autre côté la durée d'attente d'un paquet au niveau d'un routeur qui dépend de la taille de la file d'attente. Pour une étude plus rigoureuse, il est impératif de prendre en compte des incertitudes sur le retard R_0. Nous utiliserons dans ce cas une approche polytopique.

2.3.7 Modélisation par un polytope

En réalité, les matrices $\tilde{A}$ et $\tilde{A}_d$ dépendent du retard R_0. Par conséquent, les critères IOD ne sont plus valables $\forall h \in \mathbb{R}^+$, ni les critères DD quelque soit $h < h_{max}$ mais seulement pour un retard constant $h = R_0$. Nous proposons de valider la condition de stabilisation de l'observateur du Théorème 2.5 sur une plage de valeurs du retard.

2.3.7.1 Approche polytopique

L'approche polytopique est une manière de prendre en compte des incertitudes en considérant un ensemble de modèles entre lesquels le comportement du système évolue. Soit Ω un ensemble incertain donné qui englobe les matrices $\tilde{A}(R_0(t))$, $\tilde{A}_d(R_0(t))$ et $\tilde{B}(R_0(t))$ où $R_0(t)$ représente le caractère incertain. Nous supposons que R_0 varie entre deux bornes connues $R_{0_{min}}$ et $R_{0_{max}}$.
Le modèle polytopique consiste à exprimer les matrices incertaines en une somme pondérée de matrices connues et constantes. Ces matrices exprimées en fonction des bornes de l'incertitude, constituent les n sommets du polytope défini comme l'enveloppe convexe (Ω) de ces sommets. Ainsi pour le système (2.15),

$$(\tilde{A}^{[k]}, \tilde{A}_d^{[k]}) \in \Omega, \quad k = 1, \ldots, n.$$

En d'autres termes, l'ensemble incertain est constitué de toutes les combinaisons linéaires des sommets telles que :

$$\Omega = \sum_{k=1}^{n} \alpha_k (\tilde{A}^{[k]}, \tilde{A}_d^{[k]}), \quad \alpha_k \geq 0 \ ; \ \sum_{k=1}^{n} \alpha_k = 1\}.$$

S'il y a p paramètres incertains, l'ensemble des combinaisons possibles est de 2^p sommets.

2.3.7.2 Modèle polytopique

Les équations dynamiques des erreurs d'observation peuvent alors être représentées par le système :

$$\dot{\varepsilon}(t) = (\tilde{A}(R_0(t)) - L\tilde{C})\varepsilon(t) + \tilde{A}_d(R_0(t))\varepsilon(t-h). \tag{2.28}$$

Par l'approche polytopique, la condition de stabilisation doit être vérifiée sur un ensemble de systèmes défini dans (2.28) tel que $R_{0_{min}} \leq R_0(t) \leq R_{0_{max}}$. Cependant, le paramètre R_0 n'apparaît pas linéairement dans les matrices $\tilde{A}$ et $\tilde{A}_d$. De ce fait, l'ensemble défini par l'incertitude de ce paramètre est non convexe, et l'approche polytopique n'est plus valable. Pour résoudre le problème, nous allons considérer un ensemble Φ soit un polytope qui englobe l'ensemble Ω.
Pour définir le polytope Φ, nous allons décorréler les différents termes où apparaissent le paramètre R_0. Posons $\rho_1 = \frac{1}{R_0}$ et $\rho_2 = \frac{1}{R_0^2}$. Puisque qu'il y a $p = 2$ paramètres incertains, le polytope sera composé de $n = 4$ sommets. Pour une valeur de R_0 bornée, les paramètres incertains $\rho_i, \forall i \in \{1, 2\}$ varient sur un intervalle borné $[\rho_{i_{min}},\ \rho_{i_{max}}]$. Ils apparaissent linéai-

rement dans les matrices $\tilde{A}$ et $\tilde{A}_d$:

$$A = \rho_1 \left[\begin{array}{cc|cccc} 0 & 0 & 0 & 0 & \dots & 0 \\ N & -1 & 1 & 0 & \dots & 0 \\ \hline 0 & 0 & 0 & 1 & & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 \end{array}\right] + \rho_2 \left[\begin{array}{cc|cccc} \frac{-N}{C} & -\frac{1}{C} & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ \hline 0 & 0 & 0 & 1 & & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 \end{array}\right],$$

$$A_d = \rho_2 \left[\begin{array}{cc|cccc} \frac{-N}{C} & \frac{1}{C} & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ \hline 0 & 0 & 0 & 0 & & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \end{array}\right].$$

L'ensemble Φ est alors défini comme

$$\Phi = co\{(\tilde{A}^{[k]}, \tilde{A}_d^{[k]}, \quad k = 1, \dots, 4\}. \tag{2.29}$$

Pour assurer la stabilité du modèle polytopique du système (2.28), il suffit de trouver une même fonction de Lyapunov prouvant la stabilité de chacun des 4 sommets de ce polytope. Si le polytope associé est stable alors le système sera stable $\forall R_0 \in [R_{0_{min}}, R_{0_{max}}]$. Le Théorème 2.5 de stabilisation DD de l'observateur est ensuite appliqué sur le polytope défini par les sommets (2.29) de Φ.
La LMI 2.25 dépend du retard h. Dans la plage de variation du retard $[R_{0_{min}}, R_{0_{max}}]$, si h vérifie cette condition de stabilité alors toute valeur inférieure à h pourra la vérifier aussi [Gouaisbaut 2006]. Durant la résolution de la LMI 2.5, nous pouvons donc déduire la valeur maximale du retard assurant la stabilité de l'observateur.

Théorème 2.6. *Ayant un scalaire ν et pour un retard constant incertain dans $[R_{0_{min}}, R_{0_{max}}]$, s'il existe deux matrices définies positives P et $Q \in \mathbb{R}^{(k+3)\times(k+3)}$ et une matrice $S \in \mathbb{R}^{(k+3)\times 1}$ telles que les LMI suivantes soient satisfaites pour $i = \{1, \dots, 4\}$:*

$$\begin{bmatrix} \Sigma_{11}^{[i]} & \Sigma_{12}^{[i]} & \Sigma_{13}^{[i]} \\ \Sigma_{12}^{[i]^T} & \Sigma_{22}^{[i]} & 0 \\ \Sigma_{13}^{[i]^T} & 0 & \Sigma_{33} \end{bmatrix} < 0, \tag{2.30}$$

où

$$\begin{aligned}
\Sigma_{11}^{[i]} &= \tilde{A}^{[i]^T}P + P\tilde{A}^{[i]} - C^T S^T - S\tilde{C} + Q - \frac{\nu}{h}P,\\
\Sigma_{12}^{[i]} &= P\tilde{A}_d^{[i]} + h\nu(P\tilde{A}^{[i]} - S\tilde{C})^T\tilde{A}_d^{[i]} + \frac{\nu}{h}P,\\
\Sigma_{13}^{[i]} &= (P\tilde{A}^{[i]} - S\tilde{C})^T,\\
\Sigma_{22}^{[i]} &= -Q + h\nu\tilde{A}_d^{[i]^T}P\tilde{A}_d^{[i]} - \frac{\nu}{h}P,\\
\Sigma_{33} &= -\frac{1}{\nu h}P,
\end{aligned}$$

alors le système (2.28) *est asymptotiquement stable. Le gain de l'observateur est défini par :* $L = P^{-1}S$.

2.3.8 Validation sous Simulink

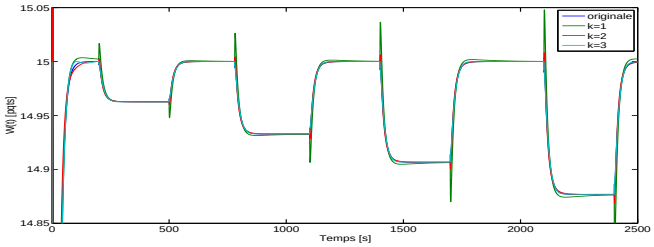
L'observateur construit par l'approche DD est testé sous Simulink en prenant la même topologie représentée par la Figure 2.6. Comme pour les observateurs précédents, le nombre de flux TCP pris est $N = 60$, $C = 3750$ paquets/s la capacité du routeur et $Tp = 0.2$s le délai de propagation. Pour le contrôle de congestion du routeur, le mécanisme RED est utilisé pour régler la fenêtre de congestion et la longueur de la file d'attente du routeur autour des valeurs à l'équilibre : $W_0 = 15$ paquets et $q_0 = 175$ paquets.
Pour chaque valeur de k, nous allons présenter la plage de variation de R_0 et la valeur maximale du retard h satisfaisant la LMI (2.30).

$$\begin{aligned}
k = 1 &: R_{0_{min}} = 0.1s, R_{0_{max}} = 0.5s, h_{max} = 0.5s;\\
k = 2 &: R_{0_{min}} = 0.1s, R_{0_{max}} = 0.5s, h_{max} = 0.36s;\\
k = 3 &: R_{0_{min}} = 0.1s, R_{0_{max}} = 0.45s, h_{max} = 0.36s.
\end{aligned}$$

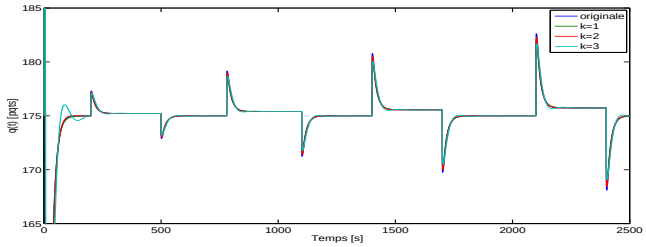
Compte tenu des valeurs précédentes, nous obtenons pour $h = 0.36$s, $R_{0_{min}} = 0.1$s et $R_{0_{max}} = 0.45$s, les matrices de gains d'observation selon k :

$$\begin{aligned}
k = 1 &: L = [0.0519, 14.9081, 4.7976, 0.0366]^T;\\
k = 2 &: L = [= [0.0143, 14.2818, 10.0208, 0.5841, 0.0205]^T;\\
k = 3 &: L = [-0.0016, 13.8721, 3.5693, 0.3138, 0.0140, 0.0004]^T.
\end{aligned}$$

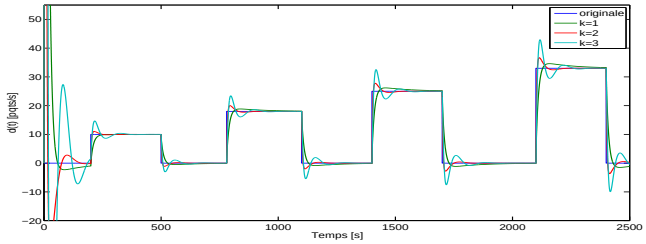
Les graphes présentés dans les Figures 2.11 et 2.12 montrent l'estimation de la fenêtre de congestion $W(t)$, la longueur de la file d'attente du routeur $q(t)$, ainsi que l'anomalie $d(t)$ à débit constant et en rampes. Les profils rectangulaires et triangulaires sont normalement présentés sous forme des polynômes de degrés 0 et 1 respectivement. Pourtant, l'observateur prenant en compte une anomalie polynômiale du premier degré n'est pas suffisant pour la reconstruire. Les graphes des Figures 2.11 et 2.12 montrent qu'en prenant $k = 2$ ou $k = 3$, l'observateur est plus sensible pour détecter la présence ou l'absence de l'anomalie et ses dérivées $\dot{d}(t)$ et $\ddot{d}(t)$. Ainsi, un observateur basé sur une anomalie du second degré converge



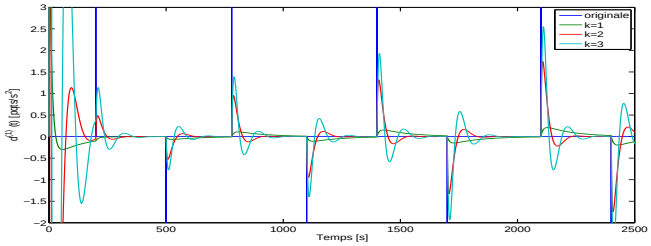
(a) La fenêtre de congestion $W(t)$.



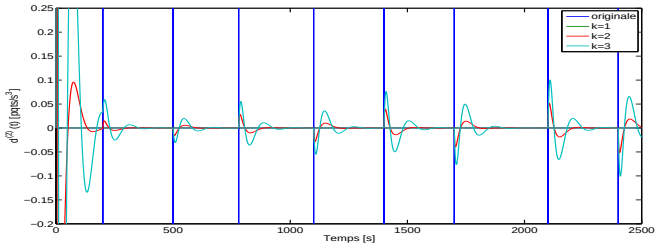
(b) La longueur de la file d'attente $q(t)$.



(c) L'anomalie $d(t)$.

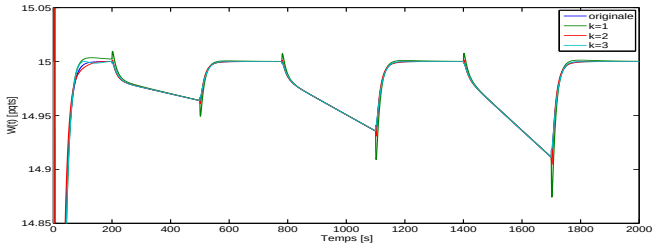


(d) La dérivée première de l'anomalie $\dot{d}(t)$.

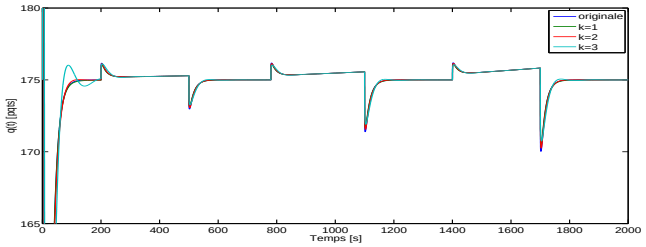


(e) La dérivée seconde de l'anomalie $\ddot{d}(t)$.

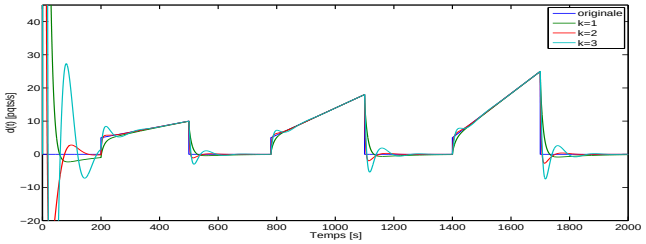
FIG. 2.11: Estimations par l'approche DD avec $d(t)$ constante.



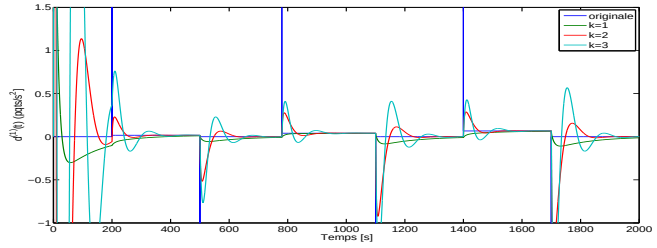
(a) La fenêtre de congestion $W(t)$.



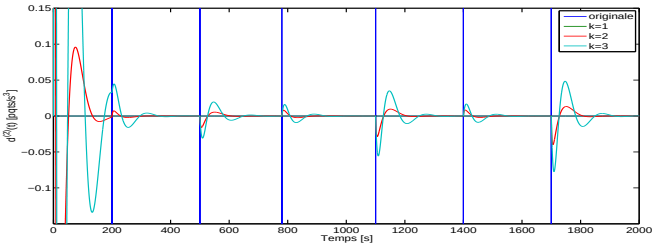
(b) La longueur de la file d'attente $q(t)$.



(c) L'anomalie $d(t)$.

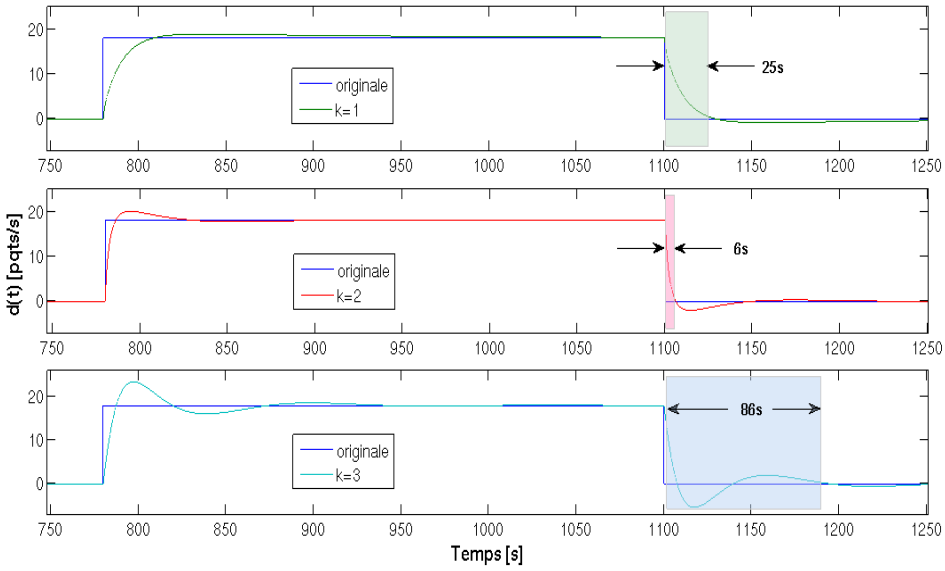


(d) La dérivée première de l'anomalie $\dot{d}(t)$.

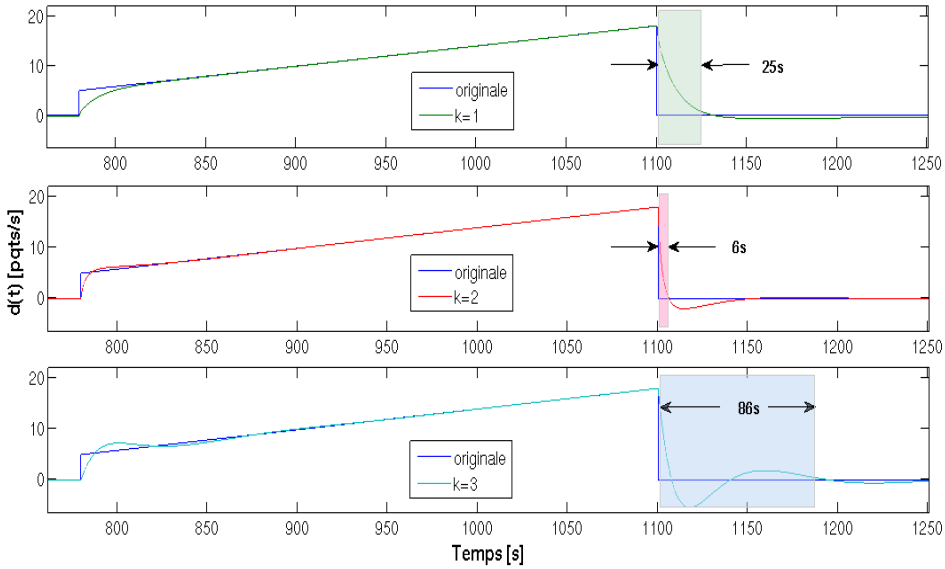


(e) La dérivée seconde de l'anomalie $\ddot{d}(t)$.

FIG. 2.12: Estimations par l'approche DD avec $d(t)$ triangulaire.



(a) $d(t)$ constante.



(b) $d(t)$ triangulaire.

FIG. 2.13: Faux positifs induits suivant k.

plus rapidement que l'observateur avec un polynôme d'ordre 1 ou même 3 vu les oscillations faiblement amorties de ce dernier.

Le principal avantage durant le processus de détection des anomalies est la rapidité surtout pour les anomalies de type *attaques DdS/DDdS (Déni de Service/Déni de Service Distribué)* présentées dans le chapitre 1. Les systèmes de détection sont actifs à chaque fois que l'estimation de l'anomalie donne une valeur positive. Durant cette simulation, les "*faux*

négatifs" ne sont pas induits puisque les observateurs, avec les différentes valeurs de k prises, estiment instantanément des valeurs non nulles suite à l'apparition d'une anomalie. Par contre, la persistance des "*faux positifs*" après l'absence de l'anomalie réelle est considérable comme le montre la Figure 2.13. Nous avons fixé un seuil en-dessous duquel l'estimation reflète l'absence véritable de l'anomalie. Pour un seuil égal à 0.5 paquets/s, les faux positifs durent $25s$ pour $k = 1$, 6s pour $k = 2$ et 86s pour $k = 3$. Les oscillations qu'engendre l'observateur, surtout avec $k = 3$, restent un inconvénient du fait qu'elles augmentent les faux positifs. Nous pouvons déduire de ces résultats expérimentaux que le second degré représente un bon compromis entre la rapidité de détection et le respect du profil de l'anomalie dans le modèle TCP/IP.

2.4 Conclusion

Dans ce chapitre, nous avons traité de la problématique de détection et reconstruction des anomalies particulières dans l'architecture de réseau TCP/IP. Nous avons vu qu'avec un observateur minimal étudié par l'approche IOD, la détection des anomalies est instantanée et le profil est bien suivi indépendamment du retard et du degré de l'anomalie polynômiale. Par ailleurs, l'approche DD est appliquée pour l'observation du système à vecteur d'état complet. Les simulations ont montré qu'en augmentant le degré du polynôme pris pour modéliser les anomalies, la sensibilité de l'observation envers la présence et disparition des anomalies augmente. Cependant, la précision dans la reconstruction du profil original diminue vu les oscillations qui s'amortissent lentement et qui apparaissent principalement avec les anomalies d'ordre 3.

La possibilité de reconstruire les dérivées supérieures de l'anomalie montrant son évolution reste le principal avantage de la modélisation pôlynomiale des anomalies présentées dans ce chapitre. Par contre, cette contrainte reste limitée aux formes polynômiales de l'anomalie et le degré du polynôme associé. Une autre approche traitant l'estimation de telles perturbations d'une façon générique sera présentée dans le chapitre suivant. Ce sont les "modes glissants" dont les multiples avantages apportent des améliorations sur notre problématique de détection des intrusions complètement inconnues.

Chapitre 3

Observation des anomalies par modes glissants

Les techniques des modes glissants ont trouvé une large application dans les domaines de détection de défauts et de supervision durant ces dernières années. Appliqués au modèle TCP/IP, les observateurs glissants permettent d'estimer la fenêtre de congestion moyenne des sources, ensuite de reconstruire les anomalies au niveau d'un routeur. Contrairement aux observateurs de Luenberger manipulés dans le chapitre 2, les observateurs glissants garantissent une convergence en temps fini ainsi que des propriétés de robustesse face aux perturbations, permettant par conséquent de reconstruire tout profil d'anomalies dans une architecture de réseaux TCP/IP. Dans ce chapitre, nous rappelons les principes des modes glissants, ensuite les algorithmes de glissement d'ordres un et supérieur. Dans une seconde partie, nous proposons plusieurs algorithmes d'observation dans le cadre de détection d'anomalies dans le modèle TCP/IP modélisé par un système d'équations différentielles. Sous Simulink, durant la reconstruction de l'anomalie, l'observateur glissant d'ordre 1 engendre de fortes oscillations que le filtrage permet de réduire en induisant des retards considérables. L'observateur d'ordre 2 est proposé pour éliminer ces oscillations montrant ainsi de bonnes performances en termes de détection et reconstruction d'anomalies.

3.1 Définitions

Considérons un système dont la dynamique est décrite par :

$$\begin{cases} \dot{x} &= f(t, x(t), u) \\ s &= s(t, x) \end{cases} \tag{3.1}$$

où $x(t) = [x_1, \ldots, x_n]^T \in \mathcal{X}$ dans $\mathcal{R}^n$ le vecteur d'état. La commande u est une fonction discontinue et bornée dépendant de l'état et du temps. La méthode des modes glissants propose de contraindre les trajectoires de (3.1) à évoluer le long d'une surface de glissement ou de commutation $\mathcal{S}$:

$$\mathcal{S} = \{x \in \mathcal{X} : s(t, x) = 0\}. \tag{3.2}$$

$s : \mathcal{X} \times \mathcal{R}^+ \to \mathcal{R}$ est une fonction suffisamment différentiable. $\mathcal{S}$ est choisie astucieusement afin que le système en boucle fermée soit asymptotiquement stable.
Nous forçons l'état du système à évoluer sur la surface s en introduisant une commande discontinue u. Le problème est que nous obtenons une équation à second membre discontinu. La théorie classique des équations différentielles ordinaires ne permet pas de décrire le comportement des solutions des équations ayant des termes discontinus. Des approches alternatives sont utilisées pour étudier de tels systèmes comme la théorie des inclusions différentielles

[Smirnov 2002] et les théorèmes de Filippov [Filippov 1988] s'y rapportant.
En général, considérons l'équation :

$$\dot{x}(t) = f(t, x), \tag{3.3}$$

où $x \in \mathcal{X}$ et $f(t, x)$ une fonction discontinue. Résoudre (3.3) est difficile au sens classique. Par conséquent, le problème peut être résolu en utilisant la conception d'inclusion. Nous allons rappeler quelques notions importantes des solutions au sens de Filippov.

3.1.1 Dynamique de glissement au sens de Filippov

Soient $S^+ = \{x \in \mathcal{R}^n | s(x) > 0\}$ et $S^- = \{x \in \mathcal{R}^n | s(x) < 0\}$ les deux demi-espaces séparés par S. Nous supposons que les fonctions f^+ et f^- définies par :

$$\lim_{s \to 0^+} f(t, x(t)) = f^+(x, t),$$
$$\lim_{s \to 0^-} f(t, x(t)) = f^-(x, t)$$

existent pour tout x et t considérés. De même, nous définissons les projections normales des champs de vecteurs f^+ et f^- à la surface s telles que :

$$f_n^+(x, t) = Pr_{normal} f^+(x, t),$$
$$f_n^-(x, t) = Pr_{normal} f^-(x, t).$$

L'existence de solution au sens de Filippov [Filippov 1988] est obtenue selon le théorème suivant.

Théorème 3.1. *Solution au sens de Filipppov :*
Soient $\phi(t) \in S$ vérifiant $f_n^+(x, t) < 0, f_n^-(x, t) > 0$ et $f_n^+(x, t) - f_n^-(x, t)$ continument différentiable. Alors il existe une solution $\phi(t)$ unique du système (3.3) si et seulement si

$$\dot{\phi}(t) = \alpha(t) f^+(\phi(t), t) + (1 - \alpha) f^-(\phi(t), t) \tag{3.4}$$

avec $\alpha(t) = \frac{f_n^-(\phi(t),t)}{f_n^-(\phi(t),t) - f_n^+(\phi(t),t)}$.

Une autre théorie est aussi employée lors de l'étude d'un système différentiel à second membre discontinu. Conçue par Utkin [Utkin 1992], cette méthode est appelée la *commande équivalente*.

3.1.2 La commande équivalente

Une fois le système évoluant sur la surface de glissement, il est nécessaire de déterminer son comportement. La *commande équivalente* u_{eq} s'exprime par les conditions d'invariance sur la surface s.

$$\dot{s} = 0 \quad \text{lorsque } s = 0.$$

Le système est alors décrit de la façon suivante :

$$\dot{x}_{eq} = f(t, x_{eq}, u_{eq})$$

Illustration simple sur un mode glissant :

Prenons un système qui agit sous la commande discontinue signe :

$$\dot{x} = K - \lambda \text{signe}(x) \tag{3.5}$$

avec $\lambda > K$. Deux cas sont présents hors de la surface de glissement $x = 0$:
- $x(0) > 0$ ce qui implique que $\dot{x} = K - \lambda$ jusqu'à $t_1 = \frac{x(0)}{\lambda - K}$;
- $x(0) < 0$ ce qui implique que $\dot{x} = K + \lambda$ jusqu'à $t_2 = \frac{x(0)}{-\lambda - K}$.

La dynamique sur la surface de glissement se calcule en résolvant :

$$\begin{cases} x &= 0, \\ \dot{x} &= 0. \end{cases}$$

D'où $u_{eq} = \frac{K}{\lambda}$. Les méthodes de la commande équivalente et les solutions de Filippov pour décrire la dynamique de glissement ne donnent pas des solutions identiques. Néanmoins, l'équivalence est assurée sous certaines conditions développées dans [Bartolini 1986]. Plus spécifiquement, ces théories sont identiques pour les systèmes linéaires en l'entrée qui font le sujet de ce mémoire.

3.1.3 Conditions d'attractivité sur la surface de glissement

En revenant à notre système initial (3.1), les conditions suffisantes permettant de garantir l'existence d'un régime glissant sont traduites mathématiquement par :

$$\lim_{s \to 0^+} \frac{\partial s}{\partial x} f(t, x(t), u) < 0 \text{ et } \lim_{s \to 0^-} \frac{\partial s}{\partial x} f(t, x(t), u) > 0. \tag{3.6}$$

ou d'une manière plus concise :

$$s\dot{s} < 0. \tag{3.7}$$

Cette condition assure une stabilité asymptotique vers la surface s. Pour une stabilité en temps fini, nous introduisons la η-attractivité

$$s\dot{s} \leq \eta |s|. \tag{3.8}$$

Effectivement, en intégrant (3.8), nous obtenons :

$$|s(t)| - |s(0)| \leq -\eta t.$$

La convergence en temps fini t_c est assurée pour une condition initiale $s(0)$, telle que :

$$t_c \geq \frac{|s(0)|}{\eta},$$

où $s(0)$ est la condition initiale.
Une fois la surface atteinte, les trajectoires restent dans un voisinage de la surface $\mathcal{S}$. Un régime est dit idéal si $s(t, x) = 0$ est exactement maintenue.

Pour assurer cette condition d'attractivité et maintenir le glissement autour de la surface, une loi de commande u peut être construite telle que :

$$u = \begin{cases} u^+(x) & \text{si } s(t,x) > 0 \\ u^-(x) & \text{si } s(t,x) < 0 \end{cases} \tag{3.9}$$

où $u^+(x)$ et $u^-(x)$ sont des fonctions continues. En respectant (3.8), la loi de commande discontinue permet d'atteindre en temps fini la surface de glissement en garantissant des propriétés de robustesse vis-à-vis des incertitudes paramétriques et perturbations. Prenons par exemple la loi de commande signe ou,

$$u(s) = -k\text{signe}(s),$$

la η-attractivité (3.8) va être assurée dépendamment de l'amplitude k de la commande. Nous allons voir dans la section d'application (3.5), qu'en considérant des incertitudes et des perturbations bornées, le calcul de k sera fonction de leurs amplitudes maximales. L'utilisation d'une valeur de k assez grande est donc nécessaire pour compléter la η-attractivité en dépit des incertitudes et des perturbations.

3.2 Le problème de réticence

Pour une dynamique de glissement idéal autour de la surface s, la commande discontinue doit commuter en fréquence infinie, ce qui n'est pas possible en réalité. En présence des dynamiques non modélisées des actionneurs, la commutation ne s'effectue pas instantanément et engendre des oscillations importantes au voisinage de la surface. Ce phénomène est appelé réticence (ou *chattering* en anglais) comme illustrée dans la Figure 3.1.

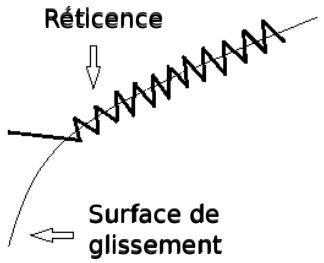


FIG. 3.1: Réticence autour de la surface de glissement.

Implantées dans des procédés industriels, ces fortes oscillations dégradent les performances des actionneurs, et peuvent produire de pertes énergétiques non négligeables dans les circuits électriques de puissance. Pour contourner ce problème, il est possible d'utiliser un filtre haute fréquence ou de remplacer les fonctions discontinues autour de la surface par des fonctions continues à grands gains [Slotine 1986] qui sont susceptibles de filtrer les hautes fréquences. Les fonctions de saturation les plus utilisées sont de la forme suivante :

$$sat(s) = \begin{cases} \frac{s}{\varepsilon} & \text{si} |s| \leq \varepsilon \\ \text{signe}(s) & \text{si} |s| > \varepsilon \end{cases} \tag{3.10}$$

La partie linéaire entre $-\varepsilon$ et ε définit l'écart du régime glissant de la surface de glissement. D'autres fonctions peuvent être appliquées comme $tanh(\frac{s}{\varepsilon})$, $\frac{2}{\pi}arctan(\frac{s}{\varepsilon})$, les sigmoïdes $\frac{\left(\frac{s}{\varepsilon}\right)^{2i+1}}{1+\left(\frac{s}{\varepsilon}\right)^{2i}}$, etc.
Cependant, bien que le phénomène de réticence peut être éliminé, la robustesse est compromise. C'est pourquoi récemment, des travaux se sont penchés sur la réalisation des lois de commande d'ordres supérieurs [Defoort 2009], [Riachy 2008], [Spurgeon 2008].

La théorie des modes glissants d'ordres supérieurs est décrite dans la partie suivante.

3.3 Modes glissants d'ordres supérieurs

Le principe des modes glissants d'ordres supérieurs est d'agir sur les dérivées d'ordre supérieur de la surface de glissement afin de réduire le phénomène de réticence et diminuer le temps de convergence vers la surface [Levant 1993], [Emel'yanov 2005]. Pour cela, nous définissons l'ensemble de glissement d'ordre r par rapport à s :

$$\mathcal{S}_r = \left\{ x \in \mathcal{X} : s = \dot{s} = \ldots = s^{(r-1)} = 0 \right\}. \tag{3.11}$$

Pour l'existence des modes glissants d'ordre r, les $(r-1)$ premières dérivées par rapport au temps doivent être continues et vérifient la condition de régularité :

$$rang \left(\begin{bmatrix} \frac{\partial s}{\partial x} \\ \frac{\partial \dot{s}}{\partial x} \\ \vdots \\ \frac{\partial s^{(r-1)}}{\partial x} \end{bmatrix} \right) = r. \tag{3.12}$$

Proposition 3.1. *Si la condition (3.12) est vérifiée, alors un mode glissant d'ordre r par rapport à s existe si et seulement si l'intersection du champs des vecteurs au sens de Filippov avec l'espace tangent à l'ensemble $\mathcal{S}_r$ n'est vide pour aucun point de glissement.*

L'élaboration d'une loi de commande u qui génère un mode glissant d'ordre r permet de définir un algorithme glissant d'ordre r par rapport à s. La commande équivalente u_{eq} est définie, pour tout degré $p \leq r$, à partir de

$$s^{(p)}(t, x, u_{eq}) = 0.$$

Un algorithme de commande idéal n'existe pas dans la réalité vu l'impossibilité de commuter à fréquence infinie. Les dynamiques glissantes prennent place au voisinage de la surface de glissement. Nous parlons ainsi d'algorithme de glissement réel.
Considérons τ le plus grand pas d'intégration. Il peut être aussi le temps de commutation maximal des actionneurs ou la période d'échantillonnage de la commande. L'algorithme idéal

d'ordre r aura la précision de convergence :

$$\begin{aligned} |s| &= o(\tau) \\ |\dot{s}| &= o(\tau^{(r-1)}) \\ \vdots &= \vdots \\ |s^{(r-1)}| &= o(\tau) \end{aligned}$$

En ce qui concerne les algorithmes réels, l'un des problèmes majeurs pour l'implantation est que la quantité d'informations nécessaires augmente régulièrement avec l'ordre de ce régime glissant. D'une manière générale, si un algorithme de glissement d'ordre r par rapport à $s = 0$ est utilisé, la connaissance de $s, \dot{s}, ..., s^{(r-1)}$ sera nécessaire. La seule exception est le super-twisting [Levant 1993] qui est un algorithme d'ordre deux qui ne requiert que l'information sur s. Nous allons maintenant décrire plus en détails ces algorithmes, essentiellement d'ordre deux qui sont utilisés en majorité dans la littérature [Riachy 2008], [Defoort 2009], [Martinez 2008].

3.4 Algorithmes de glissement d'ordre 2

Le degré relatif d'un système par rapport à sa sortie $y = s(t, x)$ dans (3.1) joue un rôle important dans le cadre des systèmes non linéaires et plus précisément dans l'existence des modes glissants.

3.4.1 Notion de degré relatif

Le degré relatif d'un système est le nombre minimum de fois que la sortie doit être dérivée par rapport au temps pour que l'entrée apparaisse d'une façon explicite [Isidori 1995]. Dans l'étude de l'ordre de la dynamique des modes glissants appliqués au système (3.1), deux cas peuvent être envisagés selon le degré relatif p du système :

i. si $p = 1$ ou $\frac{\partial \dot{s}}{\partial u} \neq 0$;

ii. si $p \geq 2$ ou $\frac{\partial s^{(i)}}{\partial u} = 0$ pour $i = 1, \ldots, p-1$ et $\frac{\partial s^{(r)}}{\partial u} \neq 0$.

Dans le cas i, les modes glissants d'ordre 1 peuvent être appliqués, néanmoins le second ordre peut être utilisé pour éviter le phénomène de réticence. u représente l'entrée du système dans le cas d'une synthèse d'une loi de commande ou la fonction discontinue ajoutée dans le cas d'élaboration d'un observateur. Par conséquent, sa dérivée $\dot{u}$ devient la commande discontinue qui force le système à atteindre et maintenir les trajectoires autour de $s = 0$ à l'aide du mode glissant d'ordre 2. De cette façon, la commande u est continue et la réticence est éliminée [Bartolini 1998], [Levant 1993]. Pour le cas ii, un mode glissant d'ordre r peut être atteint sachant que $r \geq p$.

3.4.2 Contraintes et hypothèses

Considérons le système :

$$\begin{aligned} \dot{x} &= f(t, x(t), u) \\ s &= s(t, x) \end{aligned} \tag{3.13}$$

Nous supposons que f est une fonction $\mathcal{C}^1$ par rapport à chacune de ses variables et que s est une fonction $\mathcal{C}^2$. Considérons le système de degré relatif un. Les dérivées de s s'écrivent de la manière suivante :

$$\begin{aligned} \dot{s}(t) &= \tfrac{\partial}{\partial t}s(t,x) + \tfrac{\partial}{\partial x}s(t,x)f(t,x,u) \\ \ddot{s}(t) &= \tfrac{\partial}{\partial t}\dot{s}(t,x,u) + \tfrac{\partial}{\partial x}\dot{s}(t,x,u)f(t,x,u) + \tfrac{\partial}{\partial u}\dot{s}(t,x,u)\dot{u}(t) \end{aligned} \tag{3.14}$$

Les hypothèses nécessaires pour les algorithmes de glissement de second ordre se résument ainsi :

i. Soit l'ensemble $\mathcal{U} = \{u : |u| < U_M\}$, avec U_M constante réelle et u est une fonction bornée discontinue dans le temps.

ii. Le système (3.13) admet des solutions au sens de Filippov sur l'ensemble de glissement d'ordre 2 : $s = \dot{s} = 0$ pour tout t.

iii. Il existe $u_1 \in]0, U_M[$ telle que pour toute fonction continue $u \in \mathcal{U}$ avec $|u| > u_1$, il existe un temps t_1 tel que $s(t,x)u(t) < 0$ pour tout $t > t_1$. Ainsi, la commande $u = -U_M \text{signe}(s(t_0))$, où t_0 est l'instant initial, assure d'atteindre $s = 0$ dans un temps fini.

iv. Il existe des constantes positives s_0, K_m et K_M telles que , si $|s(t,x)| < s_0$,

$$0 < K_m \leq \left|\frac{\partial \dot{s}}{\partial u}\right| \leq K_M.$$

Ces inégalités traduisent le fait que le terme $\frac{\partial \dot{s}}{\partial u}$ ne doit pas s'annuler pour l'existence d'une commande équivalente en régime glissant.

v. Soit la région de linéarité définie par $\mathcal{R}_l = \{(t,x) : |s(t,x)| < s_0\}$ autour de la surface de glissement. A l'intérieur de cette région et $\forall (t,x,u)$, il existe une constante C_0 telle que :

$$\left|\frac{\partial}{\partial t}\dot{s}(t,x,u) + \frac{\partial}{\partial x}\dot{s}(t,x,u)f(t,x,u)\right| < C_0.$$

Autour de la surface de glissement, un système de degré relatif un est décrit par :

$$\ddot{s}(t) = \gamma(t,x) + \sigma(t,x)\dot{u}.$$

Pour un système de degré relatif deux par rapport à s, si le système (3.13) peut se mettre sous la forme $f(t,x,u) = a(t,x) + b(t,x)u(t)$, par dérivations successives, le système s'écrit autour de la surface de glissement comme :

$$\ddot{s}(t) = \gamma(t,x) + \sigma(t,x)u.$$

Les hypothèses précédentes peuvent donc être considérées avec des fonctions γ et σ qui sont bornées à l'intérieur de la région de linéarité $\mathcal{R}_l$:

$$0 < K_m \leq \sigma(t,x) \leq K_M,$$

$$|\gamma(t,x)| < C_0.$$

Nous présentons dans les sous-sections suivantes les principaux algorithmes des modes glissants d'ordre 2.

3.4.3 Algorithme du twisting

Considérons un système de degré relatif un. Posons $y_1 = s$ et $y_2 = \dot{s}$. Le problème de concevoir un algorithme de second ordre est équivalent au problème de stabilisation du système du second ordre suivant :

$$\begin{cases} \dot{y}_1 &= y_2, \\ \dot{y}_2 &= \gamma(t,x) + \sigma(t,x)\dot{u}. \end{cases} \tag{3.15}$$

Les hypothèses présentées dans la section 3.4.2 sont vérifiées si $|\gamma(t,x)| < C_0$ et $0 < K_m \leq \sigma(t,x) \leq K_M$. L'algorithme de twisting [Levant 1993], [Emel'yanov 2005] est défini par la loi de commande :

$$\dot{u}(t) = \begin{cases} -u & \text{si} \quad |u| > u_{eq} \\ -U_M & \text{si } s\dot{s} > 0 \quad |u| \leq |u_{eq}| \\ -U_m & \text{si } s\dot{s} \leq 0 \quad |u| \leq |u_{eq}| \end{cases}, \tag{3.16}$$

avec u_{eq} la commande équivalente. U_M et U_m vérifient les inégalités suivantes :

$$\begin{aligned} U_m &> 4\tfrac{K_M}{s_M}, \\ U_m &> \tfrac{C_0}{K_m}, \\ U_M &< \tfrac{2C_0}{K_m} + \tfrac{K_M U_m}{K_m}. \end{aligned}$$

La commande pour un système de degré relatif deux est définie plus simplement par :

$$u(t) = \begin{cases} -U_M & \text{si } s\dot{s} > 0, \\ -U_m & \text{si } s\dot{s} \leq 0 \end{cases} \tag{3.17}$$

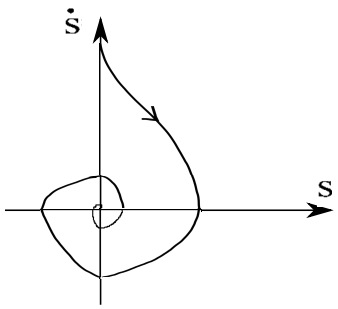


FIG. 3.2: Algorithme du twisting.

Historiquement, cet algorithme est le premier à présenter des trajectoires du système qui encerclent l'origine du plan $(s, \dot{s})$. Ces trajectoires convergent en un temps fini vers l'origine en faisant un nombre infini de rotations (c.f. Figure 3.2). Ce phénomène est dû aux commutations entre les paramètres U_M et U_m dans (3.16) et (3.17). En passant par chacun des axes s et $(\dot{s})$, les amplitudes des oscillations ainsi que les temps de rotation diminuent en suivant une progression géométrique [Fridman 2002], [Floquet 2000]. L'inconvénient de cet algorithme est qu'il nécessite la connaissance des valeurs de y_2, donc de la dérivée de s.

3.4.4 Algorithme sous-optimal

Cet algorithme est développé pour le contrôle d'un système à double intégrateur. L'intérêt de cet algorithme est qu'il ne requiert aucune connaissance de la valeur de y_2, mais une connaissance de la valeur singulière de y_1. Considérons un système de degré relatif deux. Le système auxiliaire en s est de la forme :

$$\begin{cases} \dot{y}_1 &= y_2, \\ \dot{y}_2 &= \gamma(t,x) + \sigma(t,x)u(t). \end{cases} \tag{3.18}$$

La loi de commande [Bartolini 1998] :

$$\begin{aligned} u(t) &= -\alpha(t)U_M \text{signe}(y_1(t) - \tfrac{1}{2}y_{1M}), \\ \alpha(t) &= \begin{cases} \alpha^* & \text{si} \quad [y_1(t) - \frac{1}{2}y_{1M}][y_{1M} - y_1(t)] > 0 \\ 1 & \text{si} \quad [y_1(t) - \frac{1}{2}y_{1M}][y_{1M} - y_1(t)] \leq 0 \end{cases}, \end{aligned} \tag{3.19}$$

où y_{1M} représente la valeur singulière de y_1 qui correspond au dernier instant d'annulation de $y_2 = \dot{y}_1$ [Bartolini 1998]. Les conditions suivantes assurent la convergence en temps fini :

$$\begin{aligned} \alpha^* &\in (0,1] \bigcap (0, \tfrac{3K_m}{K_M}); \\ U_M &> max\left(\tfrac{C_0}{\alpha^* K_m}, \tfrac{4C_0}{3K_m - \alpha^* K_M}\right) \end{aligned}$$

Les trajectoires du système dans le plan (y_1, y_2) sont confinées à des arcs paraboliques incluant l'origine. Nous pouvons remarquer dans la Figure 3.3, les trajectoires en paraboles successives lorsque y_1 et y_2 ne changent pas de signes. C'est le phénomène de *leaping* qui caractérise cet algorithme. Les amplitudes des oscillations des trajectoires diminuent selon une progression géométrique en passant par l'axe s.

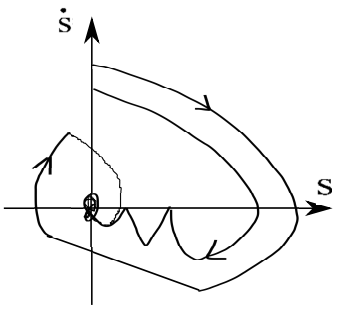


FIG. 3.3: Algorithme sous-optimal.

3.4.5 Algorithme du super-twisting

Cet algorithme est développé pour le contrôle des systèmes de degré relatif un [Levant 1993], [Floquet 2007a]. Pour cela, la loi de commande discontinue est composée de deux termes :

le premier représente la première dérivée de la commande, le second terme est fonction de la variable de glissement.

$$\begin{aligned} u(t) &= u_1(t) + u_2(t), \\ \dot{u}_1(t) &= \begin{cases} -u & \text{si} \quad |u| > |u_{eq}| \\ -V & \text{si} \quad |u| \leq |u_{eq}| \end{cases}, \\ u_2(t) &= \begin{cases} -\lambda s_0^{\rho} \text{signe}(y_1) & \text{si} \quad |s| > |s_0| \\ -\lambda |y_1|^{\rho} \text{signe}(y_1) & \text{si} \quad |s| \leq |s_0| \end{cases}, \end{aligned} \tag{3.20}$$

avec u_{eq} la commande équivalente ; V et λ vérifiant les inégalités suivantes :

$$\begin{aligned} 0 &< \rho \leq \tfrac{1}{2}, \\ W &> \tfrac{C_0}{K_m}, \\ \lambda^2 &\geq \tfrac{4C_0}{K_m^2} \tfrac{K_M(W+C_0)}{K_m(W-C_0)}. \end{aligned}$$

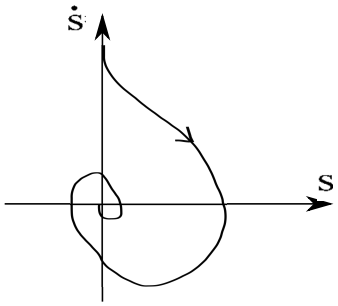


FIG. 3.4: Algorithme du super-twisting.

Cette loi de commande ne demande aucune information sur la dérivée de s [Levant 1998]. Il est montré dans [Levant 1993] qu'en prenant la loi de commande (3.20), la convergence est de type exponentiel. Le choix de $\rho = \frac{1}{2}$ assure un glissement optimal en temps de convergence (c.f. Figure 3.4).

L'étape suivante consiste à appliquer les modes glissants dans le domaine de supervision du modèle TCP/IP. Implanté au niveau du routeur, l'observateur a un double objectif : le premier est d'estimer la fenêtre de congestion moyenne des sources TCP, et le second est de reconstruire les profils d'anomalies passant par la file d'attente du buffer du routeur.

3.5 Observation des anomalies pour un modèle TCP/IP

Le modèle fluide de TCP [Misra 2000], linéarisé autour du point d'équilibre (c.f. Annexe A) et soumis à une anomalie $d(t)$, est représenté par les équations suivantes :

$$\begin{cases} \delta\dot{W}(t) &= -\frac{N}{R_0^2 C}(\delta W(t) + \delta W(t-h)) - \frac{1}{R_0^2 C}(\delta q(t) - \delta q(t-h)) \\ & \quad -\frac{R_0^2 C^2}{2N^2}\delta p(t-h), \\ \delta\dot{q}(t) &= \frac{N}{R_0}\delta W(t) - \frac{1}{R_0}\delta q(t) + d(t), \end{cases} \tag{3.21}$$

Comme dans le chapitre 2, le retard $h(t)$ est considéré constant égal à sa valeur à l'équilibre : $h = R_0 = T_p + \frac{q_0}{C}$.
Tout d'abord, à l'instar de [Edwards 1998], nous introduisons une forme canonique permettant de décomposer le système en un sous-espace inconnu et un autre défini par les sorties du système. La robustesse aux perturbations sera assurée si celle-ci satisfait les conditions de recouvrement classiques, en d'autres termes, l'appartenance à l'espace mesuré. En considérant que $\delta q(t)$ est mesuré, le système (3.21) peut se mettre sous la forme :

$$\begin{cases} \dot{x}(t) &= Mx(t) + M_d x(t-h) + Dy(t) + D_d y(t-h) + E_d u(t-h), \\ \dot{y}(t) &= Gx(t) + Hy(t) + d(t), \end{cases} \tag{3.22}$$

avec

$$M = M_d = -\frac{N}{R_0^2 C},\ D = -\frac{1}{R_0^2 C},\ D_d = \frac{1}{R_0^2 C}, E_d = -\frac{R_0 C^2}{2N^2},\ G = \frac{N}{R_0}\ \text{, et } H = -\frac{1}{R_0}.$$

$u(t)$ représente la probabilité d'éjection de paquets obtenue à partir d'un mécanisme d'AQM choisi et connu par l'observateur.
Le modèle (3.21) étant d'ordre relatif égal à 1, des observateurs glissants du premier et second ordre peuvent donc être développés.

3.5.1 Observateur glissant d'ordre 1

Pour le système (3.22), nous proposons l'observateur d'ordre 1 suivant :

$$\begin{cases} \dot{\hat{x}}(t) &= M\hat{x}(t) + M_d\hat{x}(t-h) + Dy(t) + D_d y(t-h) + E_d u(t-h), \\ \dot{\hat{y}}(t) &= G\hat{x}(t) + Hy(t) + L(\hat{y}(t) - y(t)) + \nu(t), \end{cases} \tag{3.23}$$

où L est le gain linéaire de l'observateur et $\nu(t)$ la fonction discontinue introduite pour assurer une convergence vers la surface de glissement $s = \hat{y}(t) - y(t) = 0$.
Soient les erreurs d'observation :

$$\begin{cases} e_x(t) &= \hat{x}(t) - x(t), \\ e_y(t) &= \hat{y}(t) - y(t). \end{cases} \tag{3.24}$$

Les dynamiques des erreurs le long de (3.22) sont représentées par les équations :

$$\begin{cases} \dot{e}_x(t) &= Me_x(t) + M_d e_x(t-h) \\ \dot{e}_y(t) &= Ge_x(t) + Le_y(t) + \nu(t) - d(t) \end{cases} \tag{3.25}$$

Afin de démontrer la stabilité de l'observateur glissant d'ordre 1 et la convergence de $\hat{x}$ vers x, nous proposons le théorème suivant [Rahmé 2009a], [Rahmé 2009b].

Théorème 3.2. *Soient les scalaires $L < 0$, $p_y > 0$ et une fonction discontinue appropriée*

$$\nu = \begin{cases} -ksigne(e_y), & si\ e_y \neq 0, \\ 0 & sinon. \end{cases} \tag{3.26}$$

sachant que $k > G|e_x|_{max} + |d|_{max}$, le système (3.25) est asymptotiquement stable pour tout retard $h > 0$.

Preuve. La stabilité asymptotique des erreurs d'observation (3.25) est étudiée pour chacune des erreurs séparément. La stabilité de l'erreur d'observation de l'état e_x est vérifiée par l'étude des pôles, puis la stabilité de l'erreur d'observation de la sortie e_y est étudiée en utilisant une fonction candidate de Lyapunov.

Stabilité de l'erreur d'observation de l'état e_x :

Pour assurer la convergence asymptotique de e_x vers zéro, une méthode directe est appliquée en se basant sur l'équation caractéristique quasi-polynômiale [Gu 2003]. Pour un système à retard unique avec M et M_d des constantes réelles, l'équation caractéristique quasi-polynômiale est :

$$a(s, e^{-hs}) = s - M - M_d e^{-hs}$$

qui s'écrit sous la forme :

$$a(s, e^{-hs}) = a_0(s) + a_1(s)e^{-hs}.$$

Comme $M + M_d < 0$, e_x est stable pour $h = 0$ ou, en d'autres termes, $a(s, 1)$ est stable [Gu 2003]. La marge de retard $\bar{h}$ ou la plus petite déviation de h de zéro telle que e_x devient instable peut être déterminée par :

$$\bar{h} := min\{h \geq 0 \mid a(jw, e^{-jhw}) = 0 \text{ pour } w \in \mathbb{R}\}$$

Le fait que $M = M_d < 0$ garantit que

$$\bar{h} = \frac{\cos^{-1}\left(\frac{|M_d|}{-M}\right)}{\sqrt{M_d^2 - M^2}} \to \infty.$$

Le fait que $\bar{h}$ fini n'existe pas tel que $|a(jw, e^{-jhw})| = 0$ prouve que e_x est asymptotiquement stable indépendamment du retard.

Stabilité de l'erreur d'observation de la sortie e_y :

Pour l'erreur e_y, les conditions de stabilité en temps fini sont établies en utilisant la fonction de Lyapunov suivante :

$$V(t) = p_y e_y^2(t).$$

En tenant en compte les trajectoires de la dynamique de e_y et pour ν dans (3.26), la dérivée de V devient :

$$\dot{V}(t) = (L^T p_y + p_y L)e_x^2 + 2[e_y p_y (G e_x - d(t)) - k p_y ||e_y||.]$$

Comme L est négatif, il existe $p_y > 0$ vérifiant $p_y L < 0$, et en considérant la fonction discontinue (3.26), la borne supérieure de $\dot{V}(t)$ sera :

$$\dot{V}(t) < 2p_y|e_y|(G|e_x| + d(t) - k). \tag{3.27}$$

e_x est asymptotiquement stable, donc borné par $|e_x|_{max}$. Supposons que l'anomalie admet une borne supérieure d_{max}, la condition (3.27) exige un réel $k > 0$ choisi pour compenser les termes positifs :

$$k > G|e_x|_{max} + d_{max}. \tag{3.28}$$

Cette inégalité implique l'existence de $\beta > 0$ vérifiant

$$\dot{V}(t) < -\beta\sqrt{V(t)}, \quad \beta = 2\sqrt{p_y}(k - G|e_x|_{max} + d_{max}). \tag{3.29}$$

Par conséquent, e_y converge en temps fini vers la surface de glissement $\sqrt{p_y}e_y = 0$, ou $e_y = 0$. □

Reconstruction de l'anomalie

Nous assistons à un mode de glissement sur la surface $p_y e_y = 0$ malgré la présence des perturbations $d(t)$. La nouveauté de ces modes glissants est que l'observateur tente de reconstituer les défauts. La dynamique de glissement étant maintenue, les défauts seront détectés par l'analyse de la dynamique de sortie équivalente qui représente la dynamique sur la surface de glissement. La commande discontinue étudiée pour la dynamique glissante n'est donc plus la commande $\nu(t)$ appliquée à l'observateur, mais la commande équivalente qui nécessite, en moyenne, de maintenir la dynamique glissante [Utkin 1992]. Dans les équations des erreurs d'observation (3.25), le mouvement de glissement sur $e_y = 0$ et $\dot{e}_y = 0$ mène à :

$$\nu_{eq} = -G e_x + d(t), \tag{3.30}$$

avec e_x stable. $\nu_{eq}(t)$ tend donc vers $d(t)$. Cependant, la fonction discontinue signe commute avec une fréquence infinie, introduisant ainsi une estimation de $d(t)$ avec des oscillations de très hautes fréquences [Rahmé 2009a]. La commande équivalente, et par conséquent $d(t)$, peuvent être facilement obtenues par un filtrage approprié de la commande discontinue introduite par l'observateur [Utkin 1992], [Edwards 2000], [Spurgeon 2008].

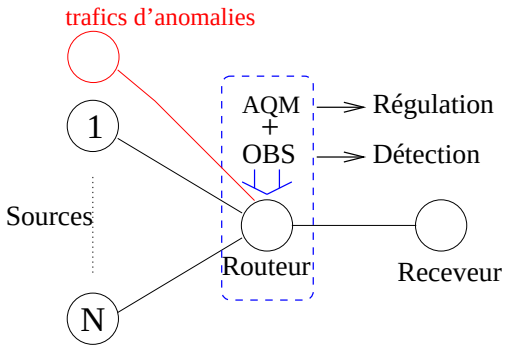


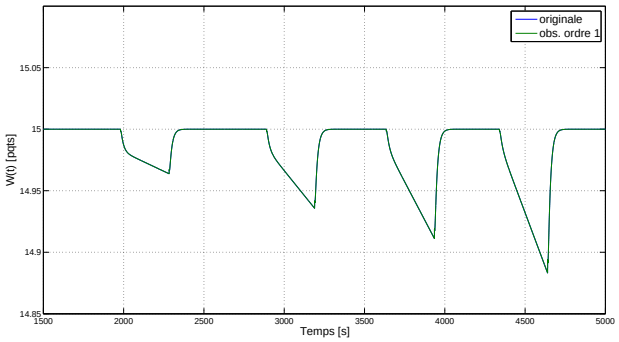
FIG. 3.5: La topologie de validation.

3.5.2 Validation sous Simulink

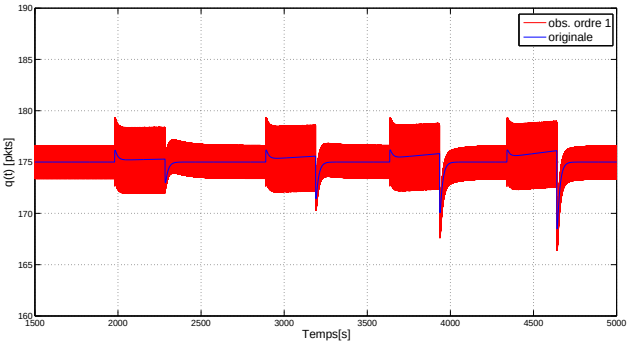
L'observateur à modes glissants d'ordre 1 proposé dans l'équation (3.23) est testé pour le modèle linéaire de TCP (3.21). Dans la topologie du modèle TCP/IP (Figure 3.5), 60 sources envoient des paquets aux destinations par l'intermédiaire d'un routeur ayant une capacité $C = 3750$ paquets/s équivalente à 15Mbps pour des paquets de taille moyenne de 500 octets. Le délai de propagation dans les liens vaut $T_p = 0.2$s.
Pour éviter la congestion au niveau du routeur, plusieurs AQM associés à l'observation proposée dans (3.23) peuvent être simulés. Dans cette simulation, nous avons choisi d'utiliser le retour d'état statique Gain-K développé dans [Ariba 2008] pour réguler la longueur de la file d'attente $q(t)$ à 175 paquets. Comparé à d'autres AQM, le Gain-K améliore les performances du routeur en réduisant les oscillations autour du point d'équilibre, limitant ainsi le taux d'éjection de paquets et optimisant l'utilisation du buffer.
L'observateur est conçu pour estimer la fenêtre de congestion moyenne en utilisant comme entrées les valeurs actuelles et retardées de la longueur de la file d'attente du routeur et la probabilité d'éjection des paquets. Une fois la convergence de l'erreur d'observation est assurée, l'anomalie $d(t)$ est reconstruite. Comme dans le chapitre 2, des anomalies périodiques de profils rectangulaires et triangulaires avec des amplitudes différentes viennent perturber le comportement normal de TCP.
Pour le calcul du gain d'observation k, nous considérons $d_{max} = 50$ paquets/s conformément aux profils pris en compte dans la Figure 3.7. Pour satisfaire la condition (3.28), nous prenons $k = 4000$. Comme amplitude de la fonction signe, cette valeur est suffisamment grande pour diminuer le temps de convergence de l'observateur. Cependant, elle induit des oscillations autour de $q(t)$ comme nous pouvons voir dans la Figure 3.6, et autour de $d(t)$ originale de telle sorte que la fonction saturation de la forme (3.10) n'est capable de les réduire. Pour cette raison, nous avons recours à la méthode de filtrage de la commande équivalente [Utkin 1992] associée à la fonction signe.

Analyse des résultats

Dans la Figure 3.6, la fenêtre de congestion est bien estimée par l'observateur d'ordre 1. Le phénomène de réticence est présent dans l'estimation de $q(t)$ résultant de la fonction signe ajoutée à sa dynamique. Un filtre passe-bas d'ordre 1 de la forme $\frac{1}{1+\tau p}$ n'est pas suffisant pour réduire correctement la réticence durant la reconstruction de $d(t)$, comme nous pouvons déduire des graphes de la Figure 3.7. En augmentant le facteur du temps τ, les oscillations



(a) La fenêtre de congestion $W(t)$.



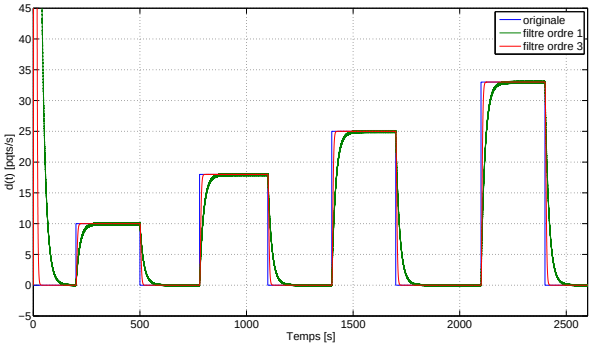
(b) La longueur de la file d'attente $q(t)$.

FIG. 3.6: Estimations par un observateur glissant d'ordre 1 de : a) la fenêtre de congestion $W(t)$ et b) la longueur de la file d'attente $q(t)$.

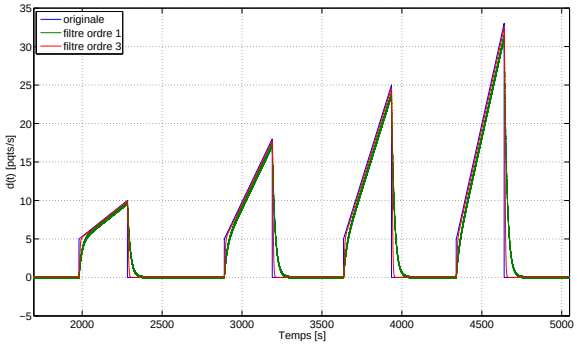
diminuent mais la reconstruction des anomalies est perturbée. Par simulations, nous avons remarqué que le filtre passe-bas d'ordre 3 est capable de réduire la réticence avec un bon suivi du profil réel d'anomalies [Rahmé 2009b].
Dans les graphes comparatifs présentés dans la Figure 3.7, la détection des anomalies par le filtre d'ordre 3 révèle une vitesse plus élevée suite à l'apparition ou la disparition de l'anomalie avec moins d'oscillations.

Notre objectif principal est de prévenir le réseau d'une anomalie passant à travers le routeur et de reconstruire ensuite son signal. Suite à l'apparition et disparition de l'anomalie, nous nous sommes intéressés à définir les durées de persistance des faux négatifs et positifs respectivement. Avant d'aller plus loin dans l'analyse des graphes, nous rappelons que la détection de l'anomalie à partir de l'équation (3.30) suppose une convergence idéale de l'estimation de la fenêtre de congestion et de la longueur de la file d'attente du routeur. Le temps de convergence de l'anomalie inclut donc les temps de convergence de e_x et de e_y vers 0 et le retard induit des filtres passe-bas.
De manière conservative, nous pouvons estimer les différents temps de convergence de e_x, e_y et des filtres passe-bas [Rahmé 2009b].

- Convergence de l'erreur d'observation de l'état e_x :



(a) $d(t)$ à débit constant.



(b) $d(t)$ à débit triangulaire.

FIG. 3.7: Reconstruction de l'anomalie par un observateur glissant d'ordre 1 avec les filtres d'ordre 1 et 3.

Le temps de convergence T_{e_x} peut être déduit de l'analyse des placements de pôles (λ_i) dans le plan s. Notons que les systèmes à retards ont un nombre infini de pôles, solutions de l'équation caractéristique $s - M - M_d e^{-hs} = 0$. Le logiciel trace-DDE [Breda 2009] permet de tracer un nombre limité de pôles comme montré dans la Figure 3.8. La dynamique est

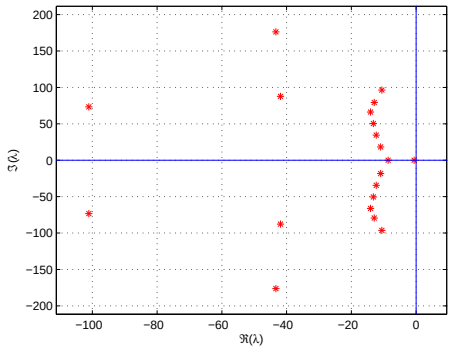


FIG. 3.8: Placement de pôles.

reliée au pôle dominant ($\lambda = -0.6$), le plus proche de l'axe des ordonnées alors que les autres sont suffisamment éloignés pour être négligés. Ce pôle impose le temps de convergence T_{e_x} à

la dynamique de e_x de la forme :

$$T_{e_x} = 3 \times \frac{1}{\lambda} = 5sec.$$

- Convergence de l'erreur d'observation de l'état e_y :
Le temps de convergence T_{e_y} est obtenu en intégrant l'inéquation différentielle de $\dot{V}(t)$ (3.29) donnant :

$$T_{e_y} = t_0 - \frac{2}{\beta}[\sqrt{V(T_{e_y})} - \sqrt{V(t_0)}]$$

avec t_0 le temps d'apparition ou disparition de l'anomalie. $V(T_{e_y}) = 0$ puisque $e_y = 0$ lorsque le système est en mode de glissement. Nous obtenons alors l'équation du temps de convergence :

$$T_{e_y} = t_0 + \frac{2}{\beta}\sqrt{p_y}|e_y(t_0)|,$$

Dans cette topologie, nous obtenons un temps maximal de stabilisation $T_{e_y} - t_0 = 0.6$s.

Dans nos expériences, les courbes des anomalies dans la Figure 3.7 révèlent des oscillations à haute fréquence autour des valeurs moyennes. Pour l'approximation du temps de convergence, une solution adoptée consiste à considérer la première fois où les oscillations rejoignent la forme originale de l'anomalie. Afin d'étudier l'efficacité d'observation de l'anomalie, nous pouvons fixer dans la période de présence (respectivement absence) d'anomalie un seuil au-dessus (respectivement en dessous) duquel la reconstruction est considérée correcte. Nous définissons ainsi, durant la période de présence d'anomalie, un seuil $thresh_{on}$ à partir duquel l'anomalie est considérée positive, ainsi la durée d'atteinte de ce seuil correspond aux faux négatifs. La durée $\mathcal{T}_{on}$ est requise pour converger vers une valeur proche de l'anomalie réelle. De plus, suite à l'absence d'anomalie, nous considérons un seuil $thresh_{off}$ à partir duquel nous garantissons l'absence d'une anomalie. La durée $\mathcal{T}_{off}$ correspondante pour atteindre $thresh_{off}$ définit les faux positifs.
Nous pouvons ensuite analyser les mécanismes de détection et de reconstruction d'anomalies avec les filtres passe-bas d'ordre 1 et 3.

Filtrage d'ordre 1

Considérons un filtre passe-bas d'ordre 1

$$\tau \dot{u}_s + u_s = u_e$$

avec la constante de temps τ, u_e l'entrée et u_s la sortie. Comme la sortie de ce filtre est de forme exponentielle, $\mathcal{T}_{on}$ et $\mathcal{T}_{off}$ sont définis par les équations suivantes :

$$\begin{aligned} \mathcal{T}_{on} &= T_{obs} + 3\tau \\ \mathcal{T}_{off} &= T_{obs} - \tau \log\left(\frac{thresh_{off}}{\hat{d}(\mathcal{T}_{0_{off}})}\right) \end{aligned} \tag{3.31}$$

T_{obs} est relatif au temps de convergence de l'observateur qui est pris en considération à chaque fois que l'anomalie varie dans le temps. Pour $\mathcal{T}_{on}$, 3τ est le temps pris par le filtre pour atteindre 95% de l'anomalie originale. Dans $\mathcal{T}_{off}$, $thresh_{off}$ est le seuil fixé durant l'absence

de l'anomalie et $\hat{d}(\mathcal{T}_{0_{off}})$ l'estimation de l'anomalie à l'instant de sa disparition $\mathcal{T}_{0_{off}}$.

Dans notre topologie, en prenant $\tau = 15$s, nous trouvons d'une part que dans (3.31), $\mathcal{T}_{on}$ est supérieure à 45s pour une reconstruction d'une anomalie et d'autre part que pour $thresh_{off} = 0.1$ paquets/s (puisque les oscillations sont de 0.2 autour de 0) et $\hat{d}(\mathcal{T}_{0_{off}}) = 5$ paquets/s, $(\mathcal{T}_{off} = T_{obs} + 58.68)$[s]. De même pour $\hat{d}(\mathcal{T}_{0_{off}}) = 10$ paquets/s, des fausses alarmes vont persister $(\mathcal{T}_{off} = T_{obs}+69)$[s] avec le filtre d'ordre 1, les mécanismes de détection et reconstruction deviennent très lents en comparant avec les temps de convergence des erreurs de l'observateur glissant.

Filtrage d'ordre 3

Considérons un filtre passe-bas d'ordre 3 ayant la fonction de transfert $\frac{1}{(1+\tau s)^3}$ et la constante de temps $\tau = 2$s. $\mathcal{T}_{on}$ et $\mathcal{T}_{off}$ sont déterminés respectivement dans les équations suivantes :

$$\begin{aligned} 0.95 \quad &= T_{obs} - \tfrac{1}{8}(-8 + 8e^{-\frac{1}{2}(\mathcal{T}_{on}-\mathcal{T}_{0_{on}})} \\ &\quad +4te^{-\frac{1}{2}(\mathcal{T}_{on}-\mathcal{T}_{0_{on}})} + t^2e^{-\frac{1}{2}(\mathcal{T}_{on}-\mathcal{T}_{0_{on}})}) \\ \tfrac{thresh\_off}{\hat{d}(\mathcal{T}_{0_{off}})} \quad &= T_{obs} + \tfrac{1}{8}(8e^{-\frac{1}{2}(\mathcal{T}_{off}-\mathcal{T}_{0_{off}})} \\ &\quad +4te^{-\frac{1}{2}(\mathcal{T}_{off}-\mathcal{T}_{0_{off}})} + t^2e^{-\frac{1}{2}(\mathcal{T}_{off}-\mathcal{T}_{0_{off}})}) \end{aligned} \tag{3.32}$$

où $\mathcal{T}_{0_{on}}$ et $\mathcal{T}_{0_{off}}$ sont respectivement les instants d'apparition et disparition d'une anomalie. Comme vu précédemment, $\mathcal{T}_{on}$ est le temps pris pour atteindre 95% du profil original de l'anomalie.
De l'équation (3.32), nous trouvons que la reconstruction, pendant les périodes de présence d'une anomalie, prend vers $(T_{obs} + 12.5)$[s] après la convergence de l'observateur. Dans les Figures 3.7 nous pouvons voir que la réduction significative de la réticence mène à diminuer le seuil $thresh_{off}$, donc à une meilleure précision de détection. Pour cela en prenant $thresh_{off} = 0.01$ paquets/s dans (3.32), une anomalie atteignant 5 paquets/s induit des faux positifs de $(T_{obs} + 20.8)$[s] et pour 10 paquets/s, les faux positifs $(T_{obs} + 22.45)$[s].

Les faux négatifs et positifs relatifs aux différentes amplitudes d'anomalies sont réduits significativement avec l'observateur glissant d'ordre 1 complété par le filtre d'ordre 3 par rapport au même observateur complété du filtre d'ordre 1.

Remarque 1. La fonction saturation introduite dans l'observateur à la place de signe est une approche pour réduire le phénomène de réticence. Pour le modèle TCP que nous avons adopté, elle ne montre pas d'efficacité durant la reconstruction de l'anomalie. Comme nous allons voir dans la partie de simulation, l'utilisation des filtres passe-bas est toujours nécessaire avec cette fonction en raison de fortes amplitudes que la fonction saturation n'arrive à réduire.

Bien que les techniques de filtrage soient pratiques, leurs principaux inconvénients demeurent dans le réglage manuel de leurs paramètres et le retard dans le temps de réponse à la variation de l'anomalie. Ce qui nous amène à adopter les techniques de glissement d'ordre supérieur

[Emel'yanov 2005]. Nous proposons dans la partie suivante un observateur glissant d'ordre 2 comme observateur d'ordre supérieur.

3.5.3 Observateur glissant d'ordre 2

Pour l'observation d'ordre 2, l'algorithme du super-twisting est considéré. Pour un système dépendant linéairemant de la commande, u peut être simplifiée n'ayant pas besoin d'être bornée comme dans (3.20) :

$$\begin{cases} \vartheta(s) &= \vartheta_1 - \lambda|s|^{\rho}\text{signe}(s), \\ \dot{\vartheta}_1 &= -\alpha\text{signe}(s), \end{cases} \tag{3.33}$$

où $\alpha > 0, \lambda > 0$ et $\rho \in (0,1)$. Cet algorithme ne dépend pas des dérivations de la surface de glissement, ce qui constitue un avantage par rapport à d'autres types de commande par modes glissants [Floquet 2007a], [Levant 1993].

L'anomalie $d(t)$ et sa dérivée sont supposées bornées par des limites supérieures d_{max} et $\dot{d}_{max}$ respectivement. Considérons la structure d'observateur décrite dans (3.34) en se basant sur l'algorithme du super twisting présenté précédemment [Rahmé 2010] :

$$\begin{cases} \dot{\hat{x}}(t) &= M\hat{x}(t) + M_d\hat{x}(t-h) + Dy(t) + D_dy(t-h) + E_du(t-h), \\ \dot{\hat{y}}(t) &= G\hat{x}(t) + Hy(t) + z(t) - \lambda|\hat{y}(t) - y(t)|^{\rho}\text{signe}(\hat{y}(t) - y(t)), \\ \dot{z}(t) &= -\alpha\text{signe}(\hat{y}(t) - y(t)). \end{cases} \tag{3.34}$$

Les erreurs d'observation sont définies par :

$$\begin{aligned} e_x(t) &= \hat{x} - x(t), \\ e_y(t) &= \hat{y} - y(t), \\ e_z(t) &= z + Ge_x - d(t). \end{aligned} \tag{3.35}$$

D'après les équations (3.22) et (3.34), les dynamiques des erreurs d'observation sont décrites par :

$$\begin{cases} \dot{e}_x(t) &= Me_x(t) + M_de_x(t-h), \\ \dot{e}_y(t) &= Ge_x(t) + z(t) - \lambda|e_y(t)|^{\rho}\text{signe}(e_y(t)) - d(t) \\ &= e_z(t) - \lambda|e_y(t)|^{\rho}\text{signe}(e_y(t)), \\ \dot{e}_z(t) &= G(Me_x(t) + M_de_x(t-h)) - \dot{d}(t) - \alpha\text{signe}(e_y(t)). \end{cases} \tag{3.36}$$

Le théorème suivant montre la stabilité asymptotique de l'observateur par modes glissants (3.34) [Rahmé 2010], [Rahmé 2011].

Théorème 3.3. *En prenant $\rho = \frac{1}{2}$, l'origine du système (3.36) est asymptotiquement stable s'il existe une matrice définie positive $P = \begin{pmatrix} p_1 & p_3 \\ p_3 & p_2 \end{pmatrix}$ et $W \in \mathbb{R}^{2\times 1}$ telles que les LMI*

suivantes soient satisfaites :

$$\frac{1}{2}E_{12}^T P + \frac{1}{2}PE_{12} - C^T W^T - WC \pm 2\Pi \begin{pmatrix} p_3 & \frac{1}{2}p_2 \\ \frac{1}{2}p_2 & 0 \end{pmatrix} < 0 \tag{3.37}$$

où $\Pi = sup_t|G(Me_x(t) + M_d e_x(t-h))| + \dot{d}_{max}$, $C = [1 \quad 0]$, *et* $E_{12} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$.

Le gain de l'observateur est obtenu à partir de :

$$L = \begin{bmatrix} \frac{\lambda}{2} \\ \alpha \end{bmatrix} = P^{-1}W.$$

Preuve. Comme pour l'algorithme de glissement d'ordre 1, les convergences des erreurs e_x et e_y sont prouvées pour (3.36) séparément. L'équation de e_x est la même que pour le premier ordre, sa stabilité est garantie indépendamment du retard h.
Ensuite, les conditions de stabilité de e_y sont étudiées en utilisant la théorie de Lyapunov. Pour cela, effectuons le changement de variables tel que ϕ soit le nouveau vecteur d'état :

$$\phi = \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} = \begin{bmatrix} |e_y|^\rho \text{signe}(e_y) \\ e_z \end{bmatrix}.$$

ϕ définit un difféomorphisme sur $\mathbb{R} - \{0\}$. Le changement de variable est bijectif puisqu'à partir de ϕ_1, nous pouvons déduire $e_y \in \mathbb{R} - \{0\}$ tel que $|e_y| = |\phi_1|^{\frac{1}{\rho}}$ et $\text{signe}(e_y) = \text{signe}(\phi_1)$. En outre, ϕ est différentiable sur $\mathbb{R} - \{0\}$. Finalement, nous excluons $e_y = 0$ mais une fois la convergence de ϕ assurée, la convergence en mode glissant sur la surface $e_y = 0$ sera atteinte.

Par dérivation de ϕ_1, nous obtenons :

$$\dot{\phi_1} = \rho \dot{e_y}|e_y|^{\rho-1} = \rho e_z |e_y|^{\rho-1} - \lambda\rho|e_y|^{2\rho-1}\text{signe}(e_y).$$

En fonction de ϕ_1 et ϕ_2,

$$\dot{\phi_1} = |e_y|^{\rho-1}[\rho\phi_2 - \lambda\rho\phi_1].$$

En choisissant $\rho = \frac{1}{2}$, l'expression de $\dot{\phi_1}$ est plus simple à élaborer en fonction de ϕ_1 et ϕ_2. Nous aurons $|e_y|^{\rho-1} = |\phi_1|^{-1}$. En complément, il est montré dans [Levant 1993] que $\rho = \frac{1}{2}$ mène à une convergence optimale vers $s = \dot{s} = 0$.
En fonction des coordonnées de ϕ, le système en e_y et e_z peut être écrit sous la forme :

$$\begin{aligned} \dot{\phi} &= |\phi_1|^{-1} \begin{pmatrix} -\frac{\lambda}{2} & \frac{1}{2} \\ -\alpha & 0 \end{pmatrix} \phi + \begin{bmatrix} 0 \\ \pi(t) \end{bmatrix} \\ &= |\phi_1|^{-1} \left(K\phi + \begin{bmatrix} 0 \\ |\phi_1|\pi(t) \end{bmatrix} \right), \end{aligned} \tag{3.38}$$

où $K = \begin{pmatrix} -\frac{\lambda}{2} & \frac{1}{2} \\ -\alpha & 0 \end{pmatrix}$ une matrice Hurwitz.
Afin de prouver la convergence de ϕ vers zéro au bout d'un temps fini, introduisons la fonction de Lyapunov candidate :

$$V = \phi^T P \phi \qquad \text{avec } P = \begin{pmatrix} p_1 & p_3 \\ p_3 & p_2 \end{pmatrix} \text{ une matrice définie positive.}$$

La dérivée de V le long de (3.38) doit être négative pour garantir la convergence des erreurs vers zéro. Elle est donnée par :

$$\dot{V} = |\phi_1|^{-1} \left(\phi^T (K^T P + PK) \phi + 2\phi^T P \begin{bmatrix} 0 \\ |\phi_1| \pi(t) \end{bmatrix} \right). \tag{3.39}$$

En d'autres termes, pour étudier la convergence en temps fini, l'existence d'une matrice Q définie positive doit être démontrée telle que :

$$\dot{V} = -|\phi_1|^{-1} \phi^T Q \phi.$$

Soit $\pi(t) = G(Me_x(t) + M_d e_x(t-h)) - \dot{d}(t)$ une fonction bornée telle que pour tout t :

$$|\pi(t)| < \Pi.$$

La condition de la convergence (3.39) montre des bilinéarités $K^T P + PK$ et des non linéarités dans le second terme. D'où la nécessité de la reformuler sous forme d'une LMI pour obtenir les gains α et λ de l'observateur.
En effet

$$\begin{aligned} K &= \begin{pmatrix} -\frac{\lambda}{2} & \frac{1}{2} \\ -\alpha & 0 \end{pmatrix} = - \begin{pmatrix} -\frac{\lambda}{2} \\ \alpha \end{pmatrix} (1 \quad 0) + \tfrac{1}{2} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \\ &= -LC + \tfrac{1}{2} E_{12} \end{aligned}$$

où L est la matrice des gains de l'observateur à calculer et en posant $W = PL$ nous aurons la forme linéaire :

$$\frac{1}{2} E_{12}^T P + \frac{1}{2} P E_{12} - C^T W^T - WC.$$

Comme $|p_3| = \pm p_3$, les non linéarités sont reformulées puisque

$$\begin{aligned} 2\phi^T P \begin{bmatrix} 0 \\ |\phi_1| \pi(t) \end{bmatrix} &= 2\pi(t) |\phi_1| (p_3 \phi_1 + p_2 \phi_2) \\ &= \pm 2\pi(t) \phi^T \begin{pmatrix} p_3 & \frac{1}{2} p_2 \\ \frac{1}{2} p_2 & 0 \end{pmatrix} \phi \\ &< \pm 2\Pi(t) \phi^T \begin{pmatrix} p_3 & \frac{1}{2} p_2 \\ \frac{1}{2} p_2 & 0 \end{pmatrix} \phi \end{aligned}$$

Nous obtenons enfin la forme quadratique telle que :

$$\dot{V} < |\phi_1|^{-1}\phi^T \left[\frac{1}{2}E_{12}^T P + \frac{1}{2}PE_{12} - C^T W^T - WC \pm 2\Pi \begin{pmatrix} p_3 & \frac{1}{2}p_2 \\ \frac{1}{2}p_2 & 0 \end{pmatrix} \right] \phi.$$

Pour garantir que $\dot{V}$ soit strictement négative, nous obtenons la LMI de stabilité définie dans (3.37). □

Reconstruction de l'anomalie

Après avoir garanti la convergence de ϕ vers zéro en temps fini où nous aurons $e_y = 0$ et $e_z = 0$, l'anomalie $d(t)$ peut être reconstruite. Puisque

$$\lim_{t\to\infty} e_x = 0,$$

l'estimation de $d(t)$ est obtenue en intégrant

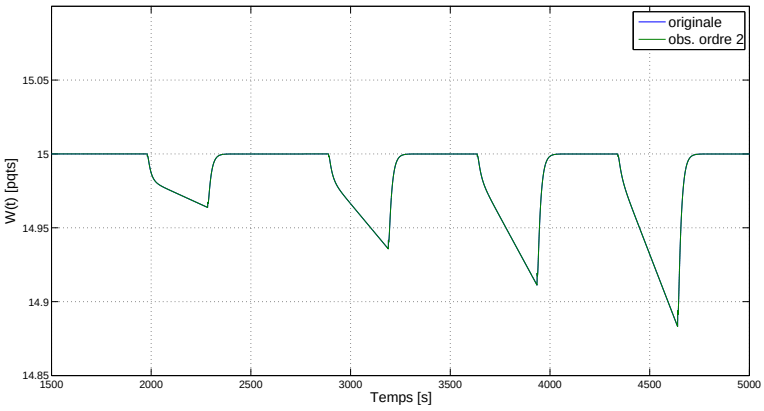
$$\dot{z} = -\alpha \text{signe}(e_y).$$

3.5.4 Validation sous Simulink

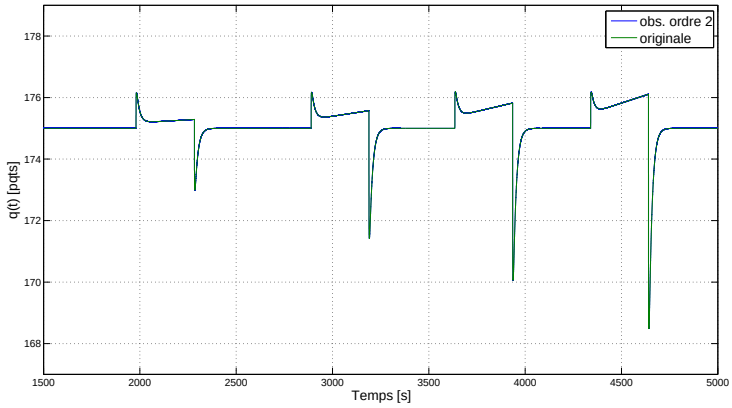
Pour tester les performances de l'observateur glissant d'ordre 2 (3.34), nous considérons la même topologie introduite pour analyser l'observateur glissant d'ordre 1 (c.f. Figure 3.5). La matrice de gain de l'observateur satisfaisant (3.37) est trouvée égale à $L = [77.227,\ 2.233]^T$. Nous comparons les performances de l'observateur glissant d'ordre 1 avec le filtre passe-bas d'ordre 3 (3.23) à celles de l'observateur glissant d'ordre 2.

Dans la Figure 3.9, l'observateur de second ordre montre son efficacité en répondant instantanément aux variations de la fenêtre de congestion $W(t)$ et la longueur de la file d'attente $q(t)$. Pour la reconstruction des anomalies dans la Figure 3.10, en comparant avec l'observateur de premier ordre, l'amélioration est également prouvée par la réduction significative du phénomène de réticence sans introduire de filtres passe-bas.

Les durées des faux positifs et faux négatifs lors de la détection des anomalies extraites des Figures 3.7 et 3.10 sont résumées dans le Tableau 3.1. Comme nous avons déjà montré dans l'analyse des filtres (c.f. section 3.5.2) pour l'observateur glissant d'ordre 1, les faux positifs dépendent des valeurs des anomalies avant leurs disparitions et de la valeur du seuil $thresh_{off}$ à partir duquel l'anomalie est estimée correctement. Pour chacun des filtres, les faux sont déduits pour les valeurs des anomalies prises dans les simulations. Les seuils $thresh_{off}$ considérés dépendent du comportement de chaque observateur durant la reconstruction de l'anomalie. Comme l'observateur d'ordre 1 associé au filtre d'ordre 1 présente des oscillations d'amplitude 0.2 de part et d'autre de zéro, le seuil est pris à 0.1 pour prendre en compte l'instant où l'anomalie estimée présente des oscillations autour de la valeur nulle. Pour le filtre d'ordre 3, les oscillations sont trop amorties, le seuil considéré est 0.01. De même pour l'observateur glissant d'ordre 2 qui présente des oscillations d'amplitude de 0.02 autour de zéro. Quant aux faux négatifs, ils sont déduits des instants où l'estimation de l'anomalie dépasse le seuil $thresh_{on}$ égal à $thresh_{off}$.



(a) La fenêtre de congestion $W(t)$.



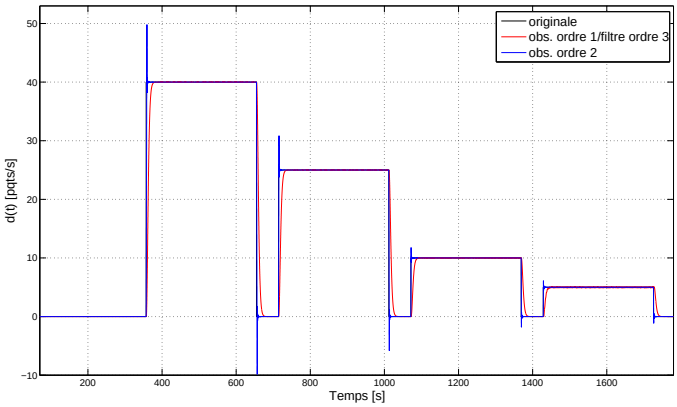
(b) La longueur de la file d'attente $q(t)$.

FIG. 3.9: Estimations par un observateur glissant d'ordre 2 de : a) la fenêtre de congestion $W(t)$ et b) la longueur de la file d'attente $q(t)$.

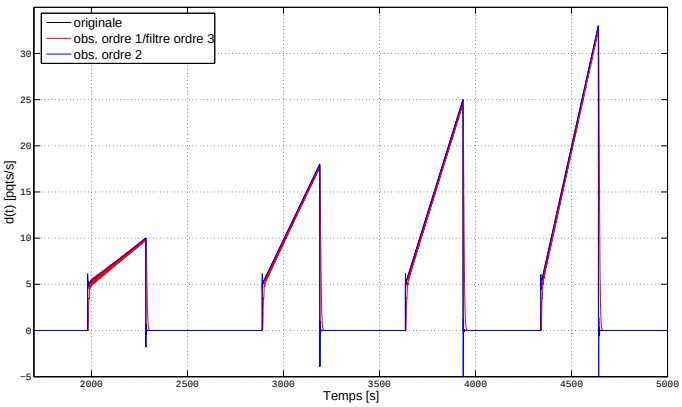
Du Tableau 3.1, nous pouvons conclure quantitativement sur les améliorations apportées par l'observateur d'ordre 2 à la Qualité de Service du modèle TCP/IP. La rapidité de détection des anomalies est garantie avec une bonne reconstruction du profil de l'anomalie.

3.6 Conclusion

Dans ce chapitre, nous avons étudié les performances des observateurs glissants pour la détection et reconstruction des anomalies dans le modèle TCP/IP. Basé sur les études analytiques et des validations sous Matlab/Simulink, l'observateur de second ordre révèle de meilleures performances par rapport au premier ordre et aux observateurs de Luenberger conçus dans le chapitre 2 en termes de détection de la présence/disparition de l'anomalie. En outre, les avantages de l'observateur du second ordre sont montrés en estimant la taille de la fenêtre de congestion du trafic TCP, puis le bon suivi en reconstruisant la forme du



(a) $d(t)$ à débit constant.



(b) $d(t)$ à débit triangulaire.

FIG. 3.10: Reconstruction de l'anomalie par un observateur glissant d'ordre 1 avec filtre d'ordre 3 et un observateur glissant d'ordre 2.

flux anormal.
Le chapitre suivant est consacré à simuler les observateurs de Luenberger et glissants sur le simulateur de réseau NS-2. Notons que Simulink est un environnement maîtrisé où TCP est représenté par son modèle mathématique. Ce qui n'est pas le cas dans NS-2 où un environnement simule les comportements des agents sources TCP, récepteurs TCP et le routeurs dans le réseau. Suite aux simulations sous NS-2, une étape intéressante est l'application sur un trafic réel. Une solution envisagée dans le chapitre suivant consiste à rejouer une trace de trafic capturé ayant des caractéristiques réelles sur le simulateur NS-2.

	ordre 1		ordre 2
	filtre 1	filtre 3	
d(t)=10	122	23	10.2
d(t)=18	131	24.1	10.5
d(t)=33	140	25.5	11.35

(a) Faux positifs

	ordre 1		ordre 2
	filtre 1	filtre 3	
d(t)=10	0.15	0.16	0.0125
d(t)=18	0.12	0.122	0.01
d(t)=33	0.06	0.065	0.008

(b) Faux négatifs

TAB. 3.1: Persistance des Faux (exprimée en secondes).

Chapitre 4

Simulations sous NS-2 et Expérimentations

Ce chapitre est dédié à la présentation des résultats expérimentaux obtenus à l'aide du simulateur de réseaux NS-2. Le logiciel Matlab/Simulink n'étant qu'une étape intermédiaire pour valider les méthodologies d'observation proposées aux chapitres précédents, nous nous orientons vers un simulateur du domaine des réseaux. Dans un premier temps, pour chacune des approches vues précédemment, nous simulons des communications TCP au travers d'un routeur en congestion géré par différentes politiques de gestion des files d'attente (AQM). Nous étudions les performances des observateurs en termes de reconstruction d'anomalies à débits constants et en rampes. Les faux négatifs et positifs analysés traduisent la rapidité de détection des observateurs, en particulier les faux négatifs qui sont les plus intéressants à réduire. Cependant, les simulations présentées dans la première partie de ce chapitre ne prennent pas en compte les caractéristiques réalistes du trafic Internet. Ainsi, dans un second temps, nous proposons une approche qui consiste à rejouer des traces capturées par des équipements de métrologie dans le simulateur NS-2, de façon à reproduire les sources de trafic réalistes et les comportements des utilisateurs. Étant implantées dans le code de NS-2, les dynamiques des observateurs sont évaluées sous des conditions réalistes de trafic TCP.

4.1 Le simulateur de réseaux NS-2 : Motivations

Les simulateurs de réseaux sont des logiciels qui offrent des supports pratiques pour observer et tester les comportements des réseaux sous différents scénarios. Avant de choisir le simulateur, nous rappelons que nous avons besoin d'une topologie de modèle TCP/IP dans laquelle nous pouvons créer des trafics d'anomalies ayant des profils constants ou triangulaires. Au niveau du routeur, les observateurs conçus dans ce mémoire doivent être implantés afin de tester leurs capacités à observer ces anomalies. Des expérimentations de rejeu de trafics TCP réels sont aussi envisagées à l'aide du simulateur. Après la phase de conception des observateurs dans ce mémoire, la validation sous des conditions réalistes est l'objectif principal de nos travaux.
Les outils de simulations les plus largement utilisés sont : OPNET basé sur la librairie OMNET++ [Varga 2009] et NS-2 [Fall 2010]. Ces simulateurs orientés objet permettent de concevoir, de simuler des protocoles et des applications pour les réseaux de communication. Ainsi, leurs noyaux codés en C++ permettent d'introduire des observateurs et de créer des générateurs de trafic.
Nous choisissons donc de simuler l'architecture ainsi que l'observateur sous NS-2 qui est l'un des outils de référence pour les chercheurs dans la communauté des réseaux informatiques. En particulier, nous profitons des modules intégrés dans ce simulateur permettant de rejouer des traces de trafic réel. Le simulateur NS-2 est particulièrement bien adapté aux réseaux à

commutation de paquets et à la réalisation de simulations de petites tailles. Il contient les fonctionnalités nécessaires à l'étude des algorithmes de routage, des protocoles de transport (TCP, UDP), de sessions, des services intégrés, des protocoles d'application comme HTTP, etc. De plus, le simulateur possède une palette de systèmes de transmission (couche 1 de l'architecture TCP/IP), d'ordonnanceurs et de politiques de gestion de files d'attente pour effectuer des études de contrôle de congestion (DropTail, filtre Token bucket ou TBF, RED, PI).
NS-2 est bâti autour d'un langage de programmation appelé Tcl (*Tool Command Language*) où la topologie du réseau est décrite ainsi que le comportement de ses composants. Le noyau de NS-2 est construit en langage C++ où des modifications peuvent être intégrées puis compilées afin de doter le logiciel de nouvelles fonctionnalités. Comme notre étude sur le modèle TCP/IP est centralisée au niveau du routeur, nous avons intégré les modèles discrétisés des observateurs dans les codes relatifs aux mécanismes des objets de gestion de la file d'attente (TBF, RED, PI). Ces scripts des observateurs sont présentés dans l'Annexe D.

4.2 Topologies des simulations sous NS-2

Comme dans les simulations sous Simulink présentées dans les chapitres précédents, la topologie du modèle TCP/IP est rappelée dans la Figure 4.1 suivante. Le trafic TCP est généré

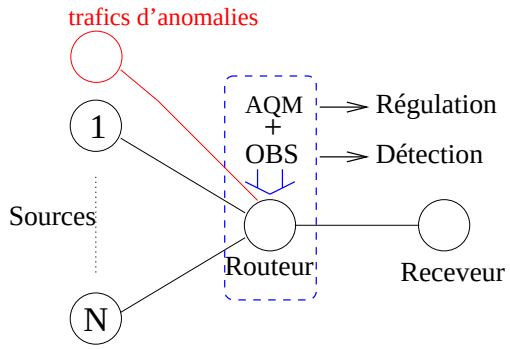


FIG. 4.1: La topologie choisie pour la simulation NS-2.

par 60 sources sous forme de connexions FTP (*File Transfer Protocol*) vers les récepteurs. L'ensemble des communications TCP empruntent le même routeur dont la capacité est égale à 10Mbps. Les flux TCP sont fractionnés en paquets de 500 octets chacun. Le trafic anormal est généré par 3 sources attaquant le routeur. Deux formes d'anomalies sont introduites à l'aide du protocole UDP dans NS-2 dans l'intervalle de $50 - 100$s. Un trafic d'anomalies à débit constant (Constant Bit Rate - CBR) est mis en œuvre dans NS-2, ainsi qu'un trafic à débit triangulaire (Triangular Bit Rate - TBR). Nous avons ajouté un générateur de trafic pour simuler la forme triangulaire comme vu précédemment dans la validation sous Simulink (chapitres 2 et 3). Le débit pour les anomalies de type CBR est de 100Kbps et celui de type TBR croît de 0 à 100Kbps.
Au niveau du routeur, la taille maximale du buffer est fixée à 800 paquets et différents AQM sont implantés pour réguler la taille de la file d'attente à $q_{ref} = 175$ paquets. Nous testons les AQM de type informatique et ceux basés sur la théorie de contrôle pour une validation plus

complète des observateurs. Comme AQM informatique, nous avons choisi le mécanisme RED [Floyd 1993] qui est conçu pour éjecter aléatoirement des paquets en calculant la probabilité d'éjection/de marquage en fonction de la taille moyenne de la file d'attente du routeur. Des AQM basés sur la théorie de contrôle comme la commande *Proportionnelle Intégrale* (PI) [Hollot 2002] et *le retour d'état statique* (Gain-K) [Ariba 2008] sont testés pour stabiliser la fenêtre de congestion moyenne des sources TCP ainsi que la file d'attente moyenne au niveau du routeur autour de leurs valeurs à l'équilibre (W_0 et q_0).

Les propositions d'observation des anomalies à profils polynômiaux et des anomalies inconnues ont été respectivement développées et validées de manière théorique, puis par simulation sous Matlab/Simulink dans les chapitres 2 et 3. Ces observateurs sont testés à l'aide du simulateur NS-2 en tenant en compte des anomalies constantes ou en rampes attaquant un routeur soumis à différents mécanismes d'AQM. Les scripts de simulation sous NS-2 et les codes d'implémentation des observateurs dans le noyau de NS-2 sont présentés dans l'Annexe D.

Les performances des observateurs sont reliées aux AQM associés et au degré du polynôme associé à l'anomalie. Dans les parties suivantes, une étude comparative est menée pour chacun des observateurs sur les deux phases d'observation : la phase de reconstruction des anomalies et de ses dérivées, ensuite la phase de détection de présence/absence des anomalies.
Durant la phase de reconstruction, des caractéristiques relatives à l'observation de $d(t)$ de type CBR et TBR sont déterminées : les temps de convergence, les erreurs moyennes entre l'estimation et l'originale, et les écart-types moyens autour des erreurs. La reconstruction des dérivées $\dot{d}(t)$ et $\ddot{d}(t)$ pour chaque cas étudié est également analysée dans le but d'étudier l'évolution de l'anomalie.
Lors de la détection des anomalies, les durées de persistance des faux positifs et négatifs sont exploitées selon les différents observateurs associés aux AQM. Nous rappelons que suite à l'apparition de l'anomalie réelle, les faux négatifs sont émis durant l'intervalle de temps où l'anomalie estimée n'atteint pas de valeurs positives. Par contre, après la disparition de l'anomalie, les faux positifs persistent tant que l'anomalie estimée est positive.

4.3 Observation IOD d'une anomalie polynômiale

Comme nous l'avons décrit dans la section 2.3.4 du chapitre 2, un observateur minimal est conçu pour reconstruire la partie non mesurée du vecteur d'état du modèle TCP (2.7) qui comprend la fenêtre de congestion moyenne des sources TCP $W(t)$, l'anomalie et ses k dérivées supérieures (l'anomalie est considérée dans cette partie comme un polynôme d'ordre k). Sous NS-2, l'observateur (2.19) est implanté au niveau du routeur gérant la longueur de la file d'attente suivant le mécanisme d'AQM considéré. La topologie étant la même que dans les simulations sous Simulink, les gains l_i $(i = 1, \ldots, k+2)$ de l'observateur sont calculés à l'aide du Théorème 2.4 suivant l'ordre k tels que les pôles sont réels et placés inférieurs à -0.6. Nous rappelons qu'une analyse de placement de pôles dans le chapitre 2, a montré que pour tout k, l'erreur d'observation admet un pôle dominant égal à -0.6 montrant une convergence exponentielle autour de 5s.
Les observations de $W(t)$, $d(t)$ constante ou en rampes, $\dot{d}(t)$ et $\ddot{d}(t)$ pour $k = 1, 2, 3$ sont

montrées dans les Figures 4.2, 4.3 et 4.4 en associant les observateurs aux AQM : RED, PI et Gain-K. Dans les cas étudiés, nous pouvons remarquer que $W(t)$ est bien estimée puisqu'elle suit toujours l'évolution moyenne de la fenêtre de congestion réelle.

Nous analysons les performances des observateurs durant la reconstruction des profils des anomalies (de type CBR et TBR) et de leurs dérivées, ainsi que leur rapidité de détection de présence/absence des anomalies à partir des Figures 4.2, 4.3 et 4.4.

Reconstruction des anomalies et leurs dérivées

Nous commençons par étudier, pour chaque AQM, l'observation de $d(t)$ de type CBR et ses dérivées $\dot{d}(t)$ et $\ddot{d}(t)$. Selon les valeurs de k, les temps de convergence (exprimés en secondes), les erreurs moyennes obtenues de la moyenne de la différence entre les signaux observés et les originaux, ainsi que les écart-types moyens (exprimés en paquets/s) résultants sont présentés dans le Tableau 4.1. Nous pouvons remarquer que les erreurs moyennes d'observation de $d(t)$ donnent des valeurs proches en comparant selon les valeurs de k. Les écart-types qui augmentent avec k proviennent évidemment des oscillations induites
Durant l'observation de $\dot{d}(t)$ et $\ddot{d}(t)$, les oscillations deviennent significatives en augmentant k pour tous les AQM. $\dot{d}(t)$ n'est pas correctement observée à cause des fortes oscillations arrivant à 500 paquets/s^2 pour $k = 2, 3$, ainsi que $\ddot{d}(t)$ avec des oscillations de 200paquets/s^3. Nous pouvons voir que pour $k = 1$, la convergence de $\dot{d}(t)$ n'est pas atteinte vers 0 mais vers des valeurs très éloignées :

- -26 paquets/s^2 après 20s avec le RED,
- 19 paquets/s^2 après 17s avec le PI,
- -11 paquets/s^2 après 14s avec le Gain-K.

	Temps de convergence [s]	Erreur moyenne [paquets/s]			Écart-type [paquets/s]		
		k=1	k=2	k=3	k=1	k=2	k=3
RED	25	-168.16	-159.1	-155.33	183.41	208.08	225.6
PI	13.7	357.57	356.73	362.02	198.31	221.01	221.06
Gain-K	15.8	64.66	66.75	71.78	195.45	226.46	241.12

TAB. 4.1: Les caractéristiques de l'observation des anomalies de type CBR par l'observateur IOD.

En ce qui concerne les signaux observés pour une anomalie de type TBR, l'observateur minimal reconstruit $d(t)$ avec des oscillations autour de pentes parallèles à la pente originale. Dans le Tableau 4.2, nous présentons pour chaque AQM, le temps de convergence vers une pente parallèle à l'originale avec les erreurs moyennes et les écarts-types autour des erreurs moyennes. Cependant les oscillations fortement générées rendent l'observation trop difficile avec les grands écarts-types comme nous pouvons déduire du Tableau 4.2. Les mêmes difficultés d'oscillations apparaissent durant l'analyse de l'observation de $\dot{d}(t)$ avec $k = 1, 2, 3$.

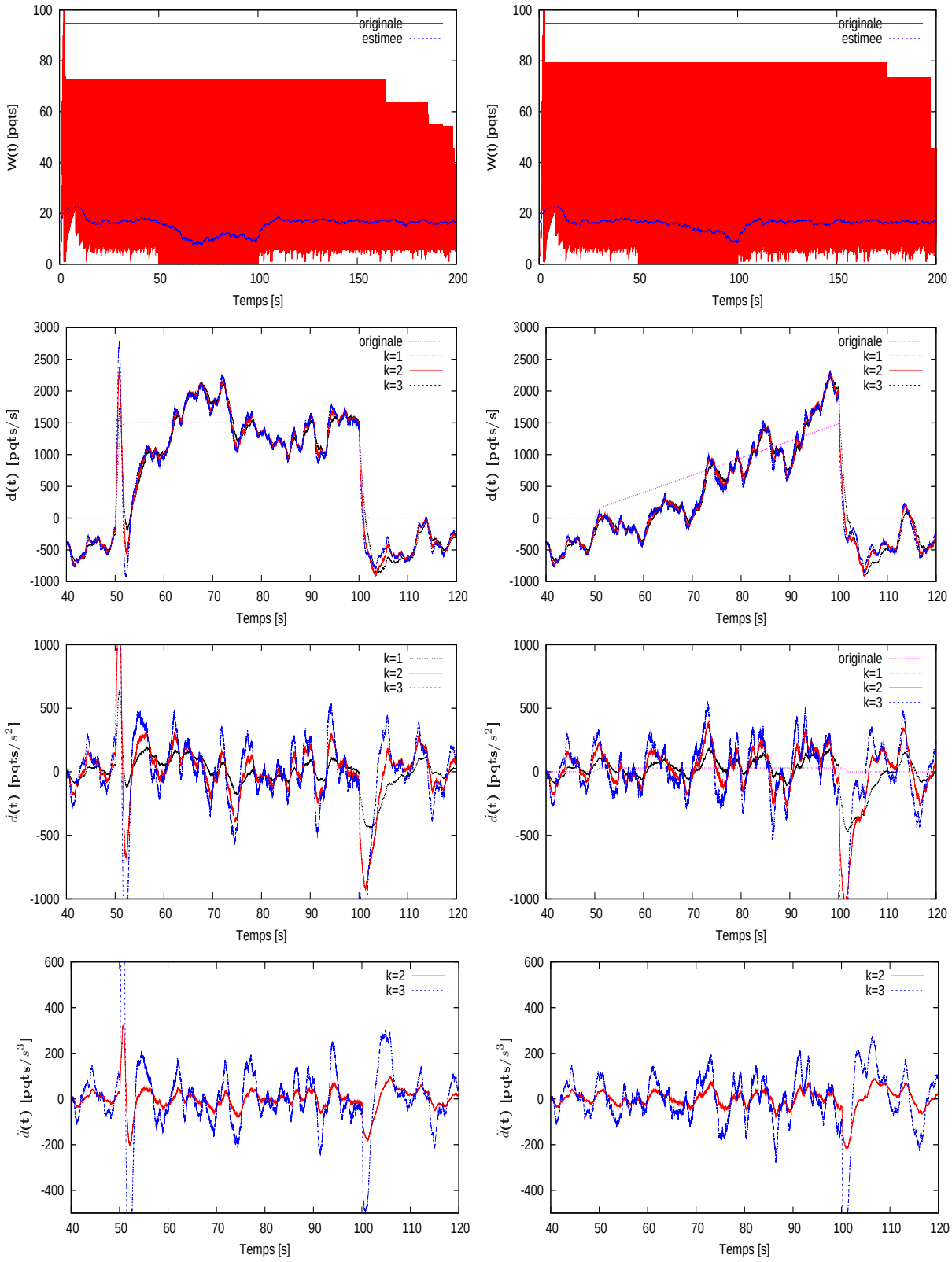


FIG. 4.2: Estimations par l'observateur minimal IOD avec l'AQM RED.

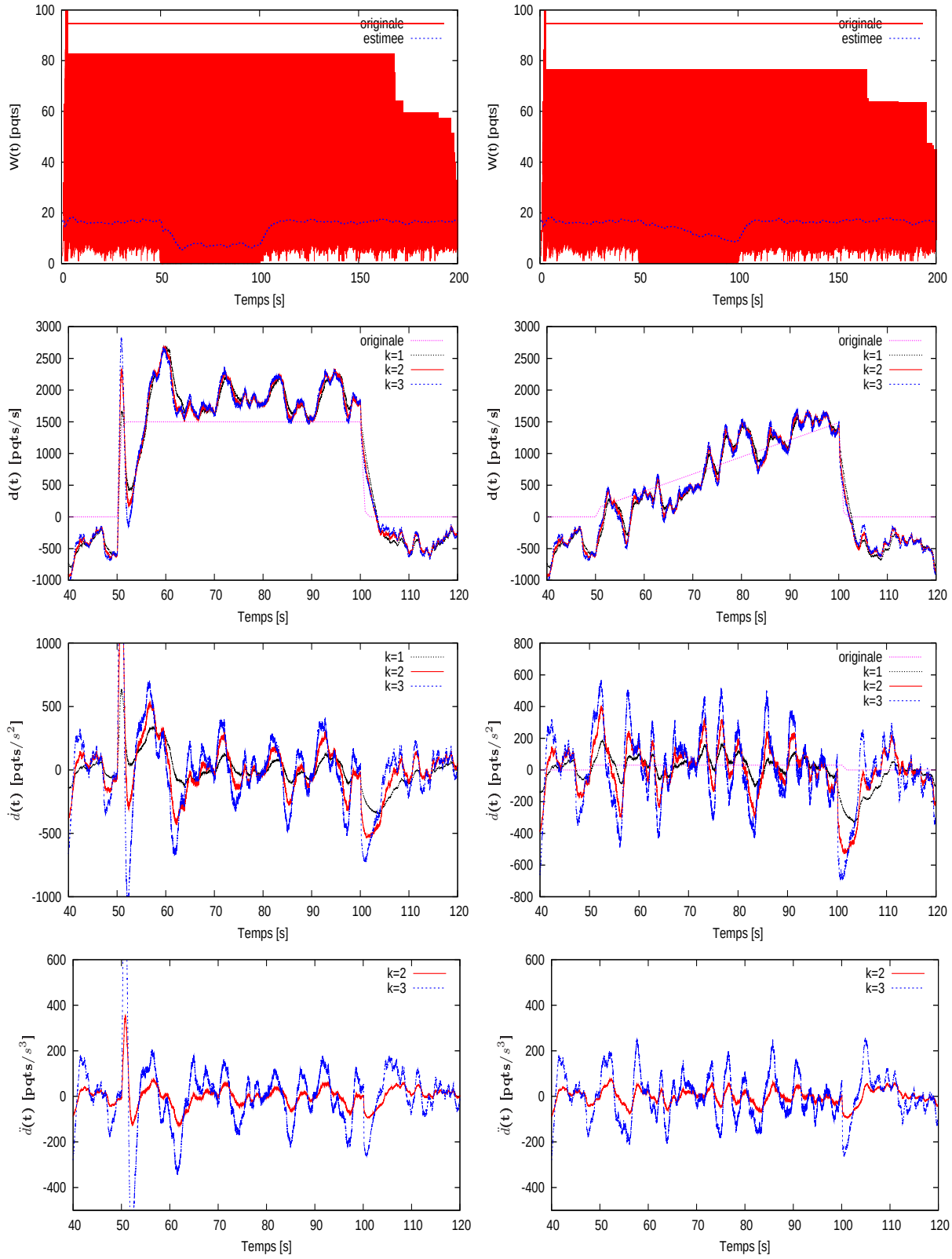


FIG. 4.3: Estimations par l'observateur minimal IOD avec l'AQM PI.

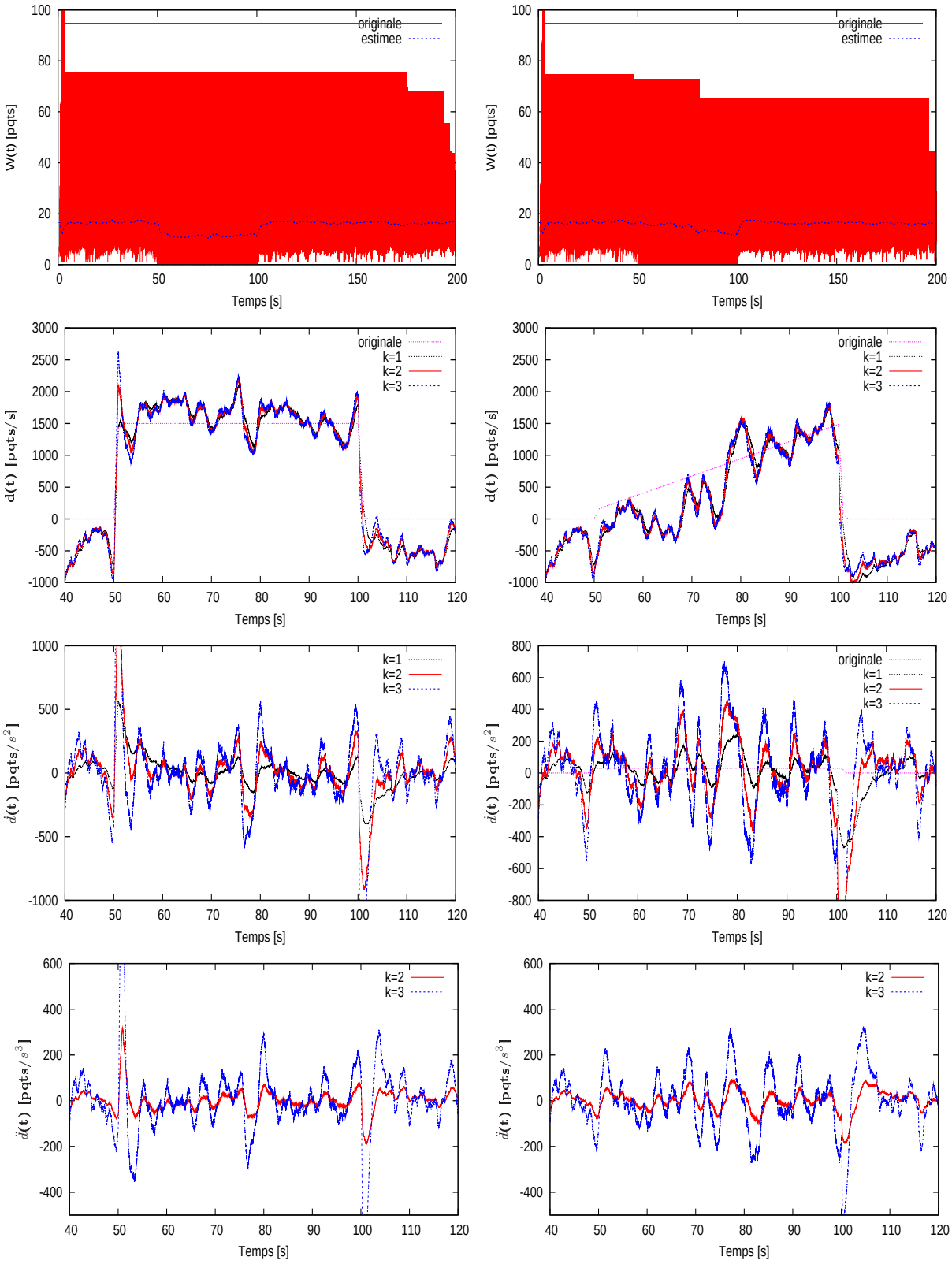


FIG. 4.4: Estimations par l'observateur minimal IOD avec l'AQM Gain-K.

	Temps de convergence [s]	Erreur moyenne [paquets/s]			Écart-type [paquets/s]		
		k=1	k=2	k=3	k=1	k=2	k=3
RED	21.3	34.23	42.4	41.37	301.28	317.29	317.96
PI	14.1	53.1	50.91	51.12	226.39	243.2	252.8
Gain-K	17.2	92.1	92.31	90.84	199.95	216.18	229.89

TAB. 4.2: Les caractéristiques de l'observation des anomalies de type TBR par l'observateur IOD.

Pour $\ddot{d}(t)$ l'observation reste possible pour $k = 2$ après 4s avec le RED, 15s avec le PI ; par contre elle est difficile à voir avec le Gain-K.

Pour la reconstruction des anomalies, nous pouvons conclure que, d'une part, les profils rectangulaires sont bien reconstruits en valeurs moyennes par les observateurs minimaux IOD conçus pour les différentes valeurs de k. Le PI est l'AQM le plus adapté à l'observateur IOD en terme de temps de convergence minimal. Par contre, en termes d'erreur minimale, le Gain-K est le plus adapté pour les CBR, et le RED pour les TBR. En analysant les écart-types, les rôles sont inversés : RED est le plus adapté pour les CBR et le Gain-K pour les TBR. D'autre part, durant l'estimation de $\dot{d}(t)$ et $\ddot{d}(t)$, les oscillations deviennent plus significatives avec k plus grand. C'est évident que le fait d'augmenter k, nous pouvons observer des dérivées supérieures mais la contre-partie réside dans les oscillations engendrées. Dans notre simulation, $k = 2$ montre un bon compromis d'estimation de $\ddot{d}(t)$ pour une meilleure analyse du comportement de l'anomalie sans trop d'oscillations avec le PI et RED.

Détection des anomalies

Nous étudions les faux négatifs et positifs qui jouent un rôle important dans la preuve d'efficacité des observateurs à détecter les anomalies, en particulier les faux négatifs qui traduisent l'échec de l'observateur à signaler leur présence effective. Ces faux sont comparés entre les différents observateurs testés avec les différents AQM. Pour définir les durées de persistance des faux négatifs et positifs, nous choisissons, dans un premier temps, un seuil à partir duquel nous considérons que l'anomalie est bien détectée. L'intervalle de temps écoulé entre l'apparition réelle de l'anomalie et le début de détection définit la durée des faux négatifs. Respectivement, les faux positifs sont relatifs à l'intervalle de temps entre la disparition réelle de l'anomalie et la fin de détection. En l'absence d'anomalie, les observateurs montrent des oscillations autour d'une valeur moyenne de -450 paquets/s, c'est ce que nous prenons comme seuil de référence pour le calcul des faux négatifs et positifs. Nous notons que nous sommes dans le cadre de la simulation, à contrario de la réalité où des estimations négatives n'existent pas.

Dans les Tableaux 4.3 et 4.4, les faux négatifs et positifs sont étudiés sur les signaux $d(t)$ observés. Pour les anomalies considérées dans cette simulation, les observateurs montrent

une rapidité de détection des anomalies. Notons qu'avec le RED, pour la détection des anomalies de type CBR, les faux négatifs sont beaucoup plus longs qu'avec les autres AQM vu que l'observateur montre dans le graphe de la Figure 4.2 de grandes oscillations passant par dessous du seuil avant de converger vers des valeurs plus grandes. Pour les anomalies de type TBR avec les AQM PI et Gain-K, les faux négatifs persistent plus longtemps qu'avec les anomalies de type CBR. Les oscillations ont la plus grande influence sur l'augmentation des durées des faux. En comparant selon k, nous notons qu'avec le PI et le RED, les faux négatifs et positifs sont dans certains cas plus petits pour $k = 2$ que pour $k = 3$.

	k=1	k=2	k=3
RED	2.05	2.65	2.85
PI	0.12	0.1	0.08
Gain-K	0.18	0.17	0.14

(a) Faux négatifs.

	k=1	k=2	k=3
RED	2.1	1.41	1.11
PI	8.6	8.54	8.56
Gain-K	4.78	1.5	1.05

(b) Faux positifs.

TAB. 4.3: Les durées de persistance des faux négatifs et positifs pour des anomalies de type CBR (exprimées en secondes).

	k=1	k=2	k=3
RED	0.16	0.14	0.11
PI	0.65	5.1	6.6
Gain-K	1.22	1.1	0.93

(a) Faux négatifs.

	k=1	k=2	k=3
RED	3.5	3.47	3.91
PI	3.8	3.4	3.6
Gain-K	1.41	0.94	0.67

(b) Faux positifs.

TAB. 4.4: Les durées de persistance des faux négatifs et positifs pour des anomalies de type TBR (exprimées en secondes).

La même procédure d'analyse que pour les observateurs IOD est prise en compte pour étudier les observateurs DD sous NS-2 dans la partie suivante.

4.4 Observation DD d'une anomalie polynômiale

La synthèse d'un observateur selon l'approche DD pour le modèle TCP (2.7) est traitée dans la partie 2.3.7.2 du chapitre 2 afin d'estimer le vecteur d'état constitué de la fenêtre de congestion moyenne des sources $W(t)$, la taille de la file d'attente du routeur $q(t)$, l'anomalie $d(t)$ et ses k dérivées successives. Pour la validation de l'observateur (2.14) sous NS-2, les gains l_i $(i = 1, \ldots, k+3)$ de l'observateur vérifiant le Théorème 2.6 sont introduits dans le code de NS-2 associé à l'AQM.

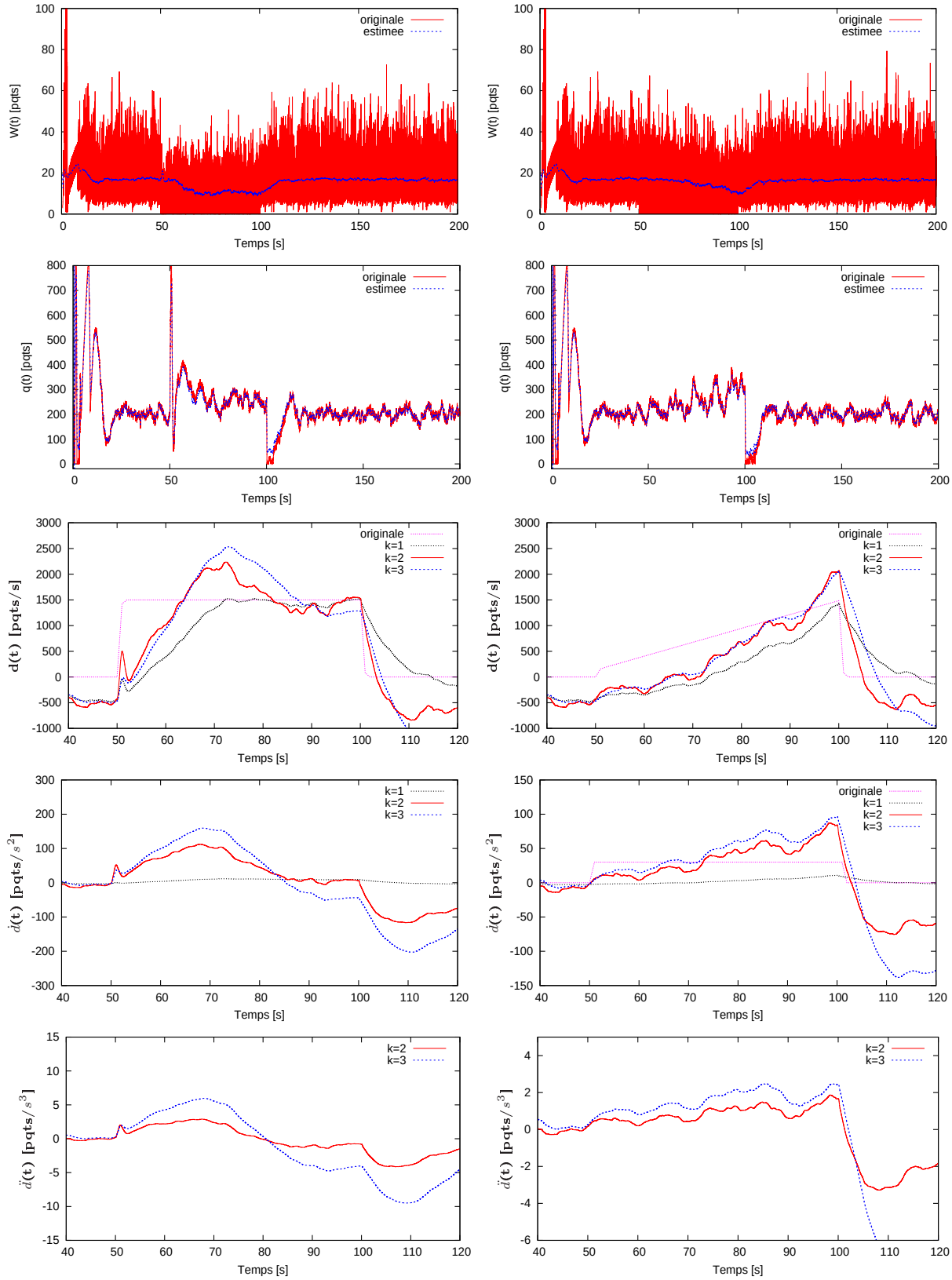


FIG. 4.5: Estimations DD avec RED.

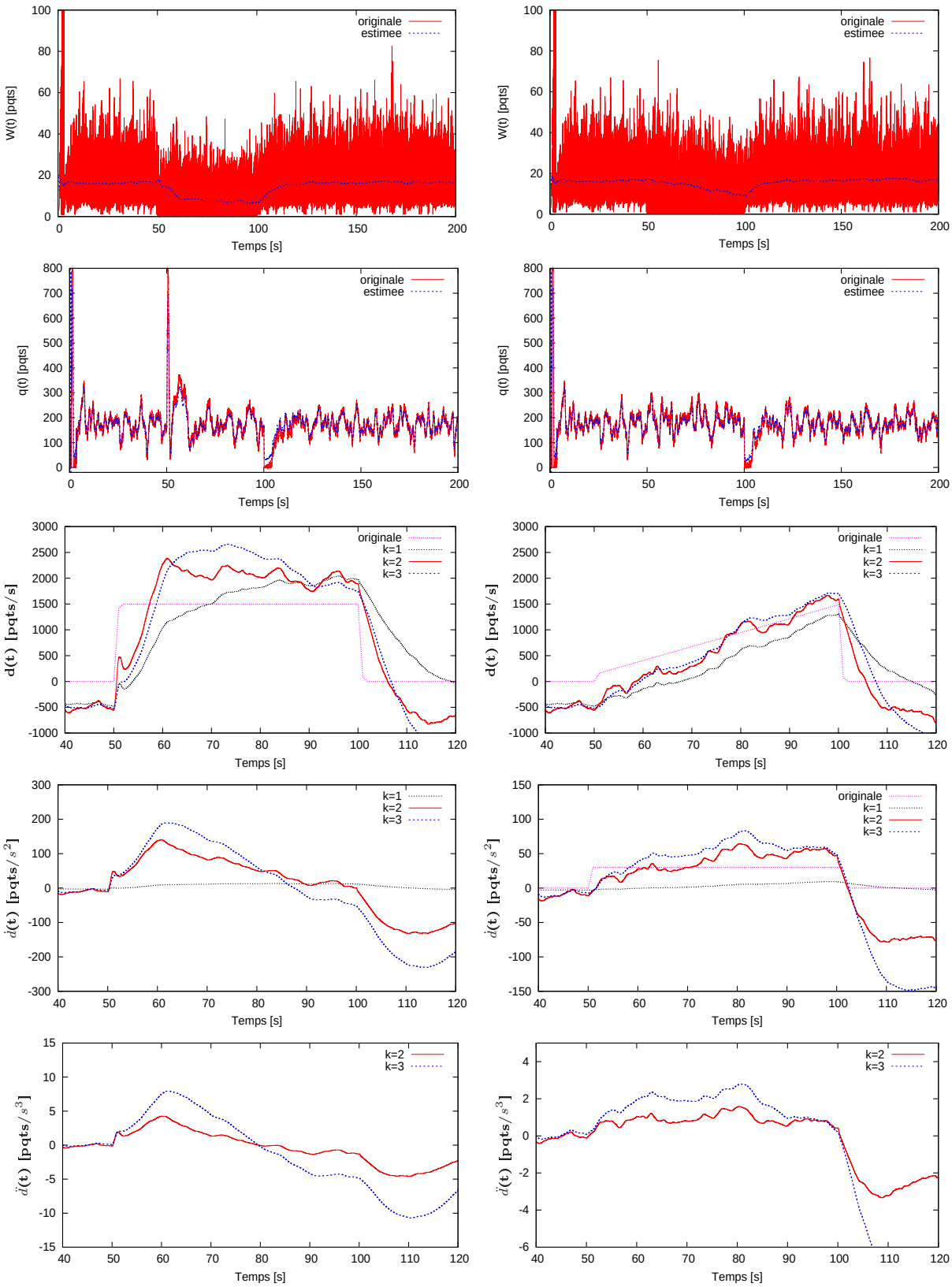


FIG. 4.6: Estimations DD avec PI.

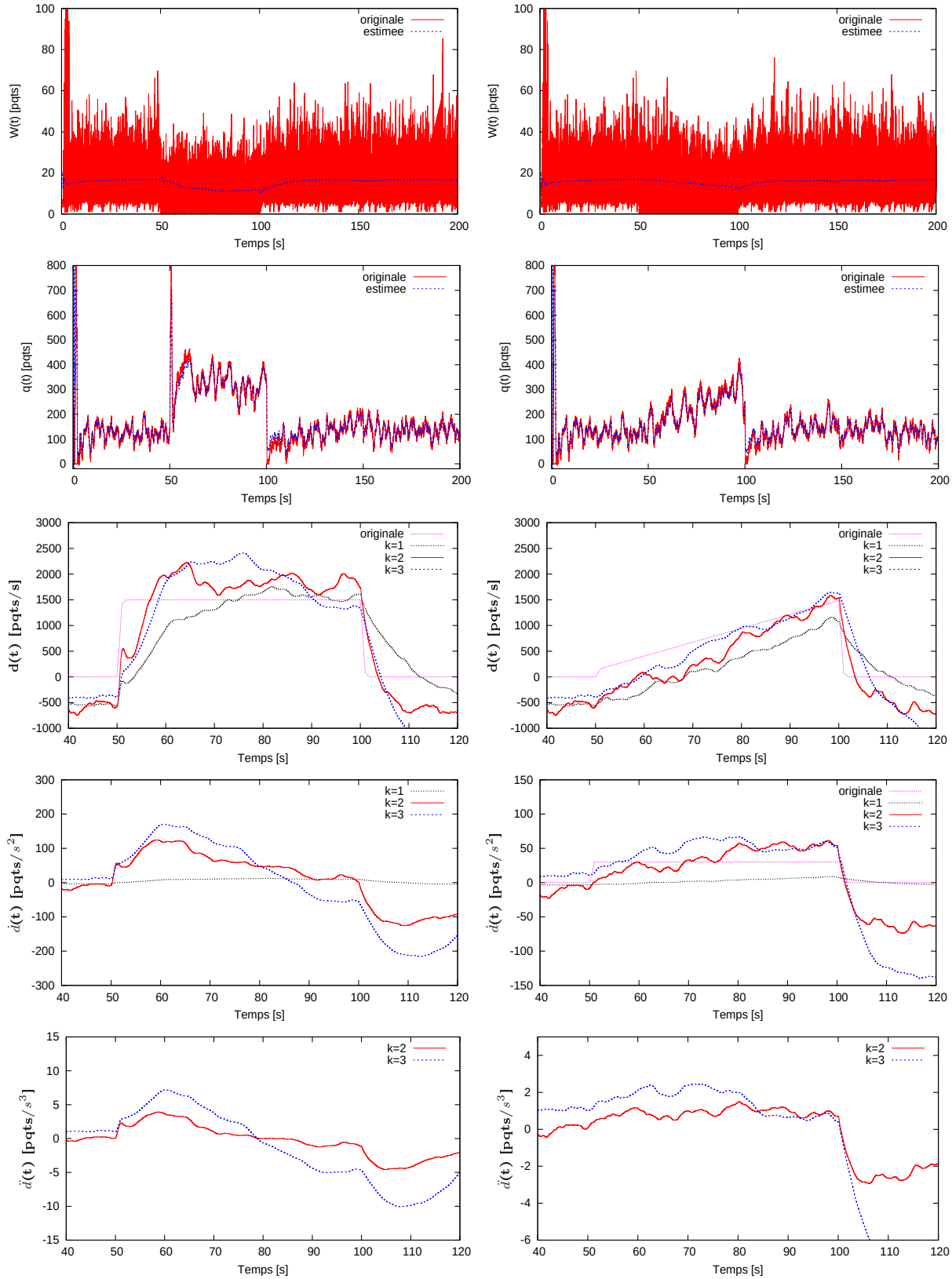


FIG. 4.7: Estimations DD avec Gain-K.

A partir des Figures 4.5, 4.6 et 4.7, nous pouvons remarquer que les observations de $d(t)$, $\dot{d}(t)$ et $\ddot{d}(t)$ par les observateurs DD ne contiennent pas de fortes oscillations comme c'est le cas avec les observateurs minimaux IOD. Comme précédemment, nous allons traiter séparément la reconstruction des anomalies et ses dérivées, ainsi que la détection des anomalies de type CBR et TBR pour chaque observateur associé à un degré du polynôme d'anomalie k et à un mécanisme d'AQM.

Reconstruction des anomalies et ses dérivées

Durant la reconstruction de l'anomalie de type CBR, les caractéristiques des performances des observateurs DD suivant $k = 1, 2, 3$ sont reportées dans le Tableau 4.5. Avec les AQM PI et Gain-K qui sont basés sur l'Automatique, nous remarquons que l'observateur $k = 2$ reconstruit plus rapidement le profil de l'anomalie. Ainsi nous pouvons voir des Figures 4.6 et 4.7 leurs convergences vers des valeurs supérieures à l'anomalie originale. Par contre, avec le RED, la reconstruction (pour $k = 1, 2$) est plus précise en terme d'erreurs moyennes et des écarts-types plus petits. De plus, $k = 1$ permet toujours d'observer $d(t)$ avec une erreur et un écart-type plus petit que ceux avec $k = 2, 3$.

	Temps de convergence [s]			Erreur moyenne [paquets/s]			Écart-type [paquets/s]		
	k=1	k=2	k=3	k=1	k=2	k=3	k=1	k=2	k=3
RED	21.8	35.4	43.2	-49.26	-114.88	-214.73	51.54	108.4	71.24
PI	30.3	10	39.1	432.4	559.1	355.7	57.8	134.84	51.7
Gain-K	24.1	7.8	41.3	89.2	351	308.4	76.81	145.2	126.5

TAB. 4.5: Les caractéristiques de l'observation des anomalies de type CBR par l'observateur DD.

Durant les reconstructions des dérivées $\dot{d}$ et $\ddot{d}$, les dépassements qui apparaissent pour $k > 1$ retardent la convergence avec tous les AQM. $\dot{d}$ converge après 0.1s vers 0 pour $k = 1$. En revanche, $k = 3$ n'assure pas de convergence.
En terme de performances d'observation de $\dot{d}$, l'observateur conçu pour $k = 1$ est meilleur que pour $k = 2$.
-Pour $k = 2$, nous avons la convergence de $\dot{d}$ vers 0 après 35s avec le RED et PI, et après 40s avec le Gain-K.
-Pour $\ddot{d}$, l'observateur avec $k = 2$ permet la convergence après 30s avec tous les AQM tandis que $k = 3$ converge vers des valeurs négatives.
Les paramètres de reconstruction des anomalies de type TBR avec les AQM sont présentés dans le Tableau 4.6. La convergence est considérée lorsque l'estimation suit le profil de l'anomalie originale. Les observateurs DD pour $k = 3$ sont plus rapides à reconstruire des anomalies de type TBR avec des erreurs moyennes plus petites que les autres valeurs de k. Notons que les observateurs DD produisent des erreurs très grandes par rapport aux autres observateurs testés avec les anomalies de type CBR et TBR. Nous pouvons remarquer dans le Tableau 4.6 que les erreurs diminuent mais les écart-types augmentent en augmentant k.

La convergence la plus rapide est assurée par l'observateur $k = 3$ associé à l'AQM PI.
Pour les dérivées des anomalies de type TBR, $\dot{d}$ pour $k = 1$ n'arrive pas à converger avec tous AQM, elle prend des valeurs très petites.
-Avec le RED, $\dot{d}$ diverge pour $k = 2, 3$.
-Avec le PI, elle converge autour de 60 paquets/s^2 au bout de 26s pour $k = 2$ et 23s pour $k = 3$.
-Avec le Gain-K, $\dot{d}$ converge autour de 50 paquets/s^2 après 30s pour $k = 2$ et 12s pour $k = 3$.
Pour $\ddot{d}$, la convergence est assurée pour $k = 2$. Elle atteint 0.8 paquets/s^3 après 3s avec le PI, et 6s avec le Gain-K.

	Temps de convergence [s]			Erreur moyenne [paquets/s]			Écart-type [paquets/s]		
	k=1	k=2	k=3	k=1	k=2	k=3	k=1	k=2	k=3
RED	10.2	10.1	9.3	-576.43	-259.67	-239.3	172.74	313.77	302
PI	7.75	4.9	5.2	-395.5	-101.9	-39.59	127.7	203.69	224.69
Gain-K	15.9	12.6	6.8	-508.55	-242.78	-97.1	83.53	208.86	143.16

TAB. 4.6: Les caractéristiques de l'observation des anomalies de type TBR par l'observateur DD.

Pour en conclure sur la reconstruction des signaux d'anomalies à l'aide des observateurs DD, $k = 1$ montre son efficacité à réduire l'erreur moyenne et l'écart-type relatifs aux types CBR, ainsi qu'à observer la convergence de $\dot{d}$. L'AQM le plus adapté est le RED. Par contre, PI et RED avec $k = 2$ montrent la meilleure rapidité de convergence avec une estimation de $\ddot{d}$.
Pour les anomalies de type TBR, l'observateur DD pour $k = 3$, associé aux PI et Gain-K, montre une meilleure convergence et reconstruction de l'anomalie avec une erreur minimale et une estimation rapide de $\dot{d}$. Pour $k = 2$, l'observateur est meilleur pour estimer $\ddot{d}$.

Détection des anomalies

Les faux négatifs et positifs sont ensuite déterminés par rapport aux seuils obtenus en moyenne en cas d'absence d'anomalies : -370 paquets/s pour $k = 1$, -400 paquets/s pour $k = 2$ et $k = 3$. Nous pouvons déduire des Tableaux 4.7 et 4.8 qu'en prenant en compte les différents AQM, la rapidité de détection de la présence ou l'absence de l'anomalie est signalée pour $k = 2$. L'observateur pour $k = 3$ agit en premier mais les oscillations induites augmentent la durée d'émission de faux négatifs après laquelle l'anomalie est estimée positive.

Après les observateurs de Luenberger simulés sous NS-2, les observateurs par modes glissants conçus dans le chapitre 3 sont analysés durant la reconstruction des anomalies complètement inconnues dans le modèle TCP/IP.

	k=1	k=2	k=3
RED	0.18	0.12	0.24
PI	0.34	0.28	0.41
Gain-K	0.4	0.25	0.24

(a) Faux négatifs.

	k=1	k=2	k=3
RED	33.72	5.16	6.25
PI	36.9	55.1	85.6
Gain-K	20.9	61.4	78.6

(b) Faux positifs.

TAB. 4.7: Les durées de persistance des faux négatifs et positifs pour des anomalies de type CBR (exprimées en secondes).

	k=1	k=2	k=3
RED	1.73	0.22	0.92
PI	2.5	1.6	2.11
Gain-K	2.8	1.4	0.3

(a) Faux négatifs.

	k=1	k=2	k=3
RED	36.4	7.33	10.27
PI	44.2	51.4	73.1
Gain-K	52.2	42.1	61.9

(b) Faux positifs.

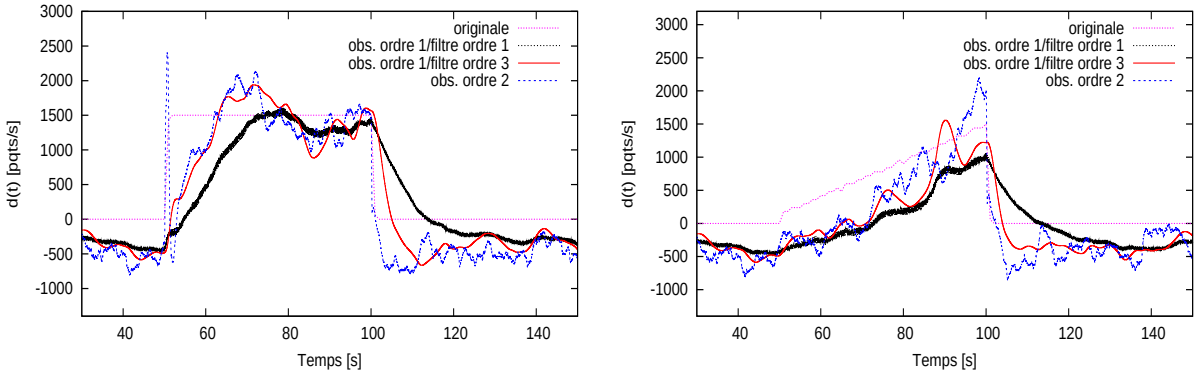
TAB. 4.8: Les durées de persistance des faux négatifs et positifs pour des anomalies de type TBR (exprimées en secondes).

4.5 Observateurs glissants

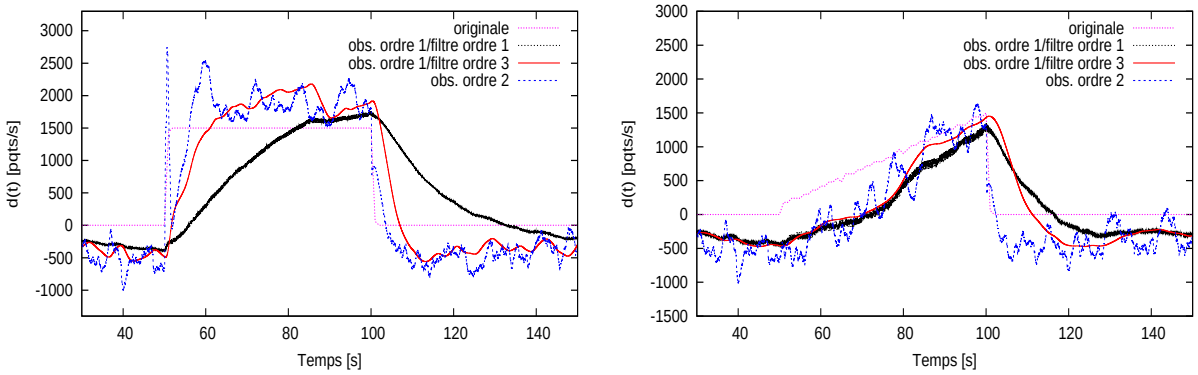
Comme dans les simulations préliminaires sous Matlab/Simulink dans le chapitre 3, les simulations sous NS-2 montrent qu'il est nécessaire d'introduire un filtre passe-bas à l'observateur glissant d'ordre 1 afin de réduire le phénomène de réticence apparu dans la reconstruction de l'anomalie [Rahmé 2010], [Rahmé 2011]. La constante de temps prise pour le filtre passe-bas d'ordre 1 est égale à 18s, et 1s pour le filtre d'ordre 3. Le réglage du filtre reste une tâche difficile dépendant du type d'anomalies et de la période d'échantillonnage. C'est pourquoi, nous avons proposé un observateur d'ordre 2 (équation (3.34) du chapitre 3) qui a prouvé sa capacité à réduire considérablement la réticence comme il est illustré dans la Figure 3.10.

Reconstruction des anomalies

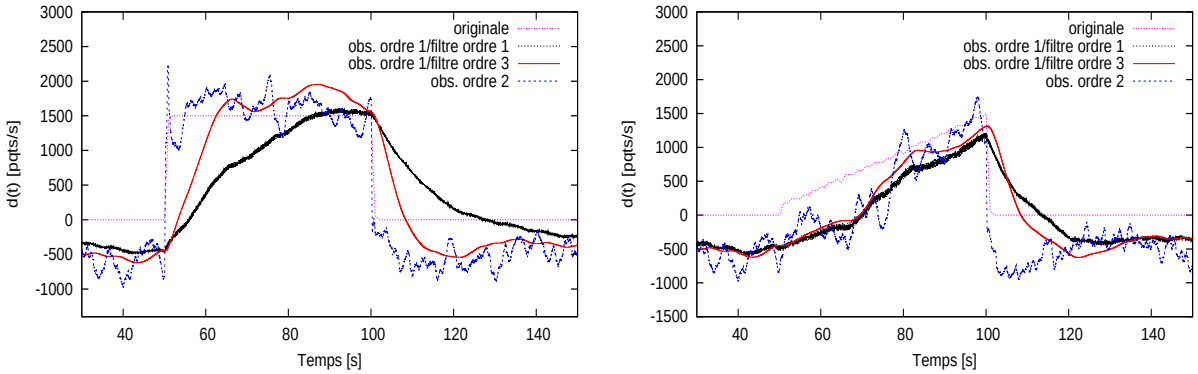
Les observateurs glissants sont simulés avec les AQM RED, PI et Gain-K. Les performances des observateurs glissants d'ordre 1 avec les filtres d'ordre 1 et 3 et d'ordre 2 sont montrées dans la Figure 4.8. Le Tableau 4.9 présente les caractéristiques de l'observation de $d(t)$ de type CBR. L'observateur d'ordre 2 montre une convergence plus rapide que l'ordre 1. Cependant l'observateur d'ordre 1/filtre d'ordre 1 montre un meilleur suivi de l'anomalie surtout avec les AQM PI et Gain-K. En effet, en présence des oscillations de hautes fréquences, le filtrage d'ordre 1 est lent mais précis. Par contre, l'augmentation de l'ordre de filtrage à 3 afin de réduire la réticence, diminue le temps de convergence en s'écartant de l'anomalie réelle et en augmentant l'écart-type. Par rapport à l'erreur d'observation et



(a) RED



(b) PI



(c) Gain-K

FIG. 4.8: Reconstruction des anomalies avec les AQM : RED, PI et Gain-K.

l'écart-type de $d(t)$ de type CBR, l'observateur d'ordre 1/filtre d'ordre 1 est meilleur que l'observateur glissant d'ordre 2 et l'observateur d'ordre 1/filtre d'ordre 3. Du Tableau 4.9, nous pouvons conclure que le Gain-K est le plus adapté à la théorie des modes glissants pour la rapidité de reconstruction et la convergence vers des formes plus proches de l'anomalie de type CBR par rapport aux autres AQM.

Pour les anomalies de type TBR, l'observateur glissant d'ordre 2 permet d'avoir une

convergence plus rapide, ainsi qu'une meilleure observation de l'amplitude en rampe par rapport aux observateurs glissants d'ordre 1 avec les AQM étudiés. Par contre, les écart-types les plus faibles que nous pouvons remarquer sont obtenus avec les observateurs d'ordre 1/filtre d'ordre 1. Pour les anomalies qui varient lentement avec le temps, les observateurs d'ordre 2 permettent d'identifier plus finement leurs amplitudes réelles.

	Temps de convergence [s]			Erreur moyenne [paquets/s]			Écart-type [paquets/s]		
	ordre 1 filtre 1	ordre 1 filtre 3	ordre 2	ordre 1 filtre 1	ordre 1 filtre 3	ordre 2	ordre 1 filtre 1	ordre 1 filtre 3	ordre 2
RED	33.9	32.1	23.3	-193	-250.4	-146.7	62.5	213.7	179.8
PI	35.5	22.9	11.1	104.3	438.7	353.4	39.7	156.1	177.2
Gain-K	36.2	18.1	6.02	46.82	254.5	118.9	24.1	129.2	189.1

TAB. 4.9: Les caractéristiques de l'observation des anomalies de type CBR par les observateurs glissants.

	Temps de convergence [s]			Erreur moyenne [paquets/s]			Écart-type [paquets/s]		
	ordre 1 filtre 1	ordre 1 filtre 3	ordre 2	ordre 1 filtre 1	ordre 1 filtre 3	ordre 2	ordre 1 filtre 1	ordre 1 filtre 3	ordre 2
RED	12.2	14.1	11.3	-634.4	-414.4	-236.3	108.4	263.1	356.7
PI	8.6	11.5	7.9	-503.2	-307.6	-341.8	160.5	227.3	312
Gain-K	6.1	3.8	4.1	-482.9	-371.6	-346.3	131.2	179.8	324.8

TAB. 4.10: Les caractéristiques de l'observation des anomalies de type TBR par les observateurs glissants.

Détection des anomalies

Les seuils à partir desquels les faux négatifs et positifs sont déterminés sont -450 paquets/s pour l'observateur glissant d'ordre 2, -350 paquets/s pour l'observateur d'ordre 1 avec les filtres passe-bas d'ordres 1 et 3. Dans les Tableaux 4.11 et 4.12, l'observateur glissant d'ordre 2 permet de mieux réduire les faux négatifs et positifs que l'ordre 1 en présence des filtres. Il faut noter qu'en présence du RED, les faux négatifs de l'ordre 2 montent à 2.9s vu la présence d'oscillations qui retardent l'observation d'une anomalie au-dessus du seuil.

	ordre 1 filtre 1	ordre 1 filtre 3	ordre 2
RED	0.39	0.83	2.4
PI	1.2	0.98	0.06
Gain-K	1.5	1.3	0.05

(a) Faux négatifs.

	ordre 1 filtre 1	ordre 1 filtre 3	ordre 2
RED	50.01	8.07	2.57
PI	62.5	9.6	8.1
Gain-K	61.6	12.02	5.9

(b) Faux positifs.

TAB. 4.11: Les durées de persistance des faux négatifs et positifs pour des anomalies de type CBR (exprimées en secondes).

	ordre 1 filtre 1	ordre 1 filtre 3	ordre 2
RED	3.21	1.6	0.36
PI	5.4	3.8	2.2
Gain-K	5.84	4.88	1.12

(a) Faux négatifs.

	ordre 1 filtre 1	ordre 1 filtre 3	ordre 2
RED	32.08	6.5	3.8
PI	56.6	50.9	3.5
Gain-K	22.3	15.4	1.08

(b) Faux positifs.

TAB. 4.12: Les durées de persistance des faux négatifs et positifs pour des anomalies de type TBR (exprimées en secondes).

4.6 Discussions des résultats des simulations

Pour conclure sur la partie de simulations sous NS-2, nous proposons pour chaque mécanisme d'AQM, les observateurs qui montrent de meilleures performances durant : la reconstruction des anomalies de type CBR et TBR ainsi que leurs dérivées où les temps de convergence et les erreurs moyennes de leurs observations sont pris en compte, et la détection de présence et absence de ses anomalies. Les observateurs IOD servent idéalement à minimiser les faux négatifs et positifs. Par contre, les observateurs DD et les observateurs glissants sont meilleurs pour la reconstruction et dans certains cas pour la diminution des faux négatifs et positifs. Nous allons essayer de combiner des observateurs à chaque AQM.

i. avec le RED :

- Pour les anomalies de type CBR, les observateurs les plus efficaces sont :
  - DD avec $k = 1$ pour la reconstruction, sinon $k = 2$ est le second choix pour l'observation de la dérivée seconde ;
  - DD avec $k = 2$ pour les faux négatifs ;
  - IOD $\forall k$, puis glissants d'ordre 2 pour les faux positifs.
- Pour les anomalies de type TBR, les observateurs sont :
  - DD avec $k = 2$ pour la reconstruction ;
  - IOD $\forall k$, puis DD avec $k = 2$ pour les faux négatifs ;

  - glissants d'ordre 2, puis IOD $\forall k$ pour les faux positifs.

ii. avec le PI :

- Pour les anomalies de type CBR, les observateurs sont :
  - glissants d'ordre 2 pour la reconstruction ;
  - glissants d'ordre 2 pour les faux négatifs ;
  - glissants d'ordre 2 pour les faux positifs.
- Pour les anomalies de type TBR, les observateurs sont :
  - DD avec $k = 3$ pour la reconstruction ;
  - DD avec $k = 3$ pour les faux négatifs ;
  - glissants d'ordre 2, puis IOD $\forall k$ pour les faux positifs.

iii. avec le Gain-K :

- Pour les anomalies de type CBR, les observateurs sont :
  - glissants d'ordre 2 pour la reconstruction ;
  - glissants d'ordre 2 pour les faux négatifs ;
  - IOD $\forall k$, puis glissants d'ordre 2 pour les faux positifs.
- Pour les anomalies de type TBR, les observateurs sont :
  - glissants d'ordre 2 pour la reconstruction ;
  - glissants d'ordre 2 pour les faux négatifs ;
  - glissants d'ordre 2 pour les faux positifs.

Par rapport aux observateurs de Luenberger DD et les observateurs glissants d'ordre 2 conçus dans ce mémoire prouvent qu'ils sont meilleurs à observer l'évolution pour les profils d'anomalies considérés sous NS-2. Les avantages de l'observateur glissant d'ordre 2 sont la détection et la reconstruction de n'importe quel profil d'anomalie sans avoir besoin des paramètres de filtrage à régler ou de l'ordre du polynôme de l'anomalie à préciser. Le point fort des observateurs de Luenberger est leur capacité à apporter plus d'informations sur l'anomalie et l'évolution de ses dynamiques en augmentant le degré pour le polynôme représentant l'anomalie. Le compromis reste entre le nombre de dérivées que nous désirons élaborer et les durées tolérables aux émissions des faux négatifs et positifs.

Les observateurs de Luenberger IOD, DD et les observateurs glissants d'ordre 2 sont appliqués aux trafics TCP ayant des caractéristiques réalistes [Rahmé 2010], [Rahmé 2011]. La méthode de rejeu des traces de trafic que nous avons adoptée est détaillée dans la section suivante.

4.7 Rejeux de traces de trafic

Dans ce mémoire, notre travail s'est concentré sur un modèle mathématique simplifiant le comportement du modèle TCP/IP à partir duquel des observateurs ont été construits pour la détection/reconstruction des anomalies. La vérification des performances est validée à l'aide des logiciels de modélisation Matlab/Simulink et NS-2 où les trafics sont synthétiques. Un autre niveau de test important est d'introduire la notion de trafic réel capturé.

L'approche que nous proposons consiste à rejouer, dans un simulateur, les traces capturées par les équipements de métrologie de façon à générer des sources de trafic réalistes et à reproduire les comportements des utilisateurs et de leurs applications. Dans [Owezarski 2004],

un module est proposé pour extraire toutes les caractéristiques pertinentes des trafics capturés pour les rejouer sous NS-2, ainsi que les outils nécessaires pour la simulation. Dans la suite, nous allons présenter la méthode de rejeu de traces. Comme il est montré dans [Owezarski 2004], les trafics rejoués à l'aide de cette approche reproduisent la complexité et la dynamique des trafics réels.

4.7.1 Présentation de la méthode de rejeu

Comme point de départ pour l'utilisation de l'approche du rejeu, une trace de trafic qui peut donner des informations au niveau du débit est requise. Cette trace qui contient les informations sur les paquets peut être générée par le logiciel de capture *tcpdump* ou de type trace DAG [Cleary 2000]. Un exemple du format d'une trace capturée est montré dans l'Annexe C.
Le simulateur NS-2 contient les fonctionnalités nécessaires pour rejouer un trafic réseau à partir de traces réelles. Il faut cependant appliquer un certain nombre de traitements aux traces métrologiques brutes pour exploiter les informations des flux émis. L'environnement de simulation doit être construit de façon à ce que la mise en forme des paquets soit faite de façon cohérente avec ce qui s'est passé dans la réalité. Les agents d'émission et de réception NS doivent injecter dans le réseau les flux aux moments précis dans la trace et selon les tailles des paquets lues dans la trace réelle [Floyd 2001]. En particulier, pour rejouer intégralement une trace, les éléments caractéristiques du trafic réel qui doivent intervenir sont, d'une part, les dates relatives des débuts de flux qui représentent le comportement réel et aléatoire des utilisateurs, et d'autre part, les tailles des paquets à l'intérieur de chaque flux.
NS-2 ne fournit qu'une seule classe pour rejouer une trace réelle : la classe "trafficTrace" qui permet de générer un flux à partir d'un fichier contenant la liste des tailles des paquets de ce flux.
Par conséquent, il faut parcourir la trace, classer les paquets TCP par flux en enregistrant pour chaque flux la date de début et la taille de chaque paquet inclu dans ce flux. Un exemple de flux extrait dans un fichier sous le format suivant :

[N] //Nième flux
début_du_flux
nombre_de_paquets
taille_paquet_1 taille_paquet_2 . . .

Ensuite, il faut créer un fichier de trace par flux. Le format de ce fichier, imposé par "trafficTrace" du simulateur, doit correspondre à une succession d'enregistrements de la forme :

```
typedef struct_trec {
unsigned int trec_time; //durée inter-paquet
unsigned int trec_len; //taille du paquet
} trec;
```

Comme TCP gère lui-même les instants d'envois des paquets par flux, le paramètre de durée inter-paquet est mis à 0 lorsqu'on rencontre un paquet TCP dans la trace. En outre, les agents UDP ou TCP dans NS-2 ont un fonctionnement basé sur une taille de paquet fixe.

Pour respecter exactement les tailles des paquets, il nous a fallu apporter des modifications au code C++ de NS-2 pour que les paquets soient émis aux tailles voulues.

Pour introduire toutes les données précédentes dans le script TCL de la simulation, un autre fichier *info* est créé pour contenir les dates de début de chaque flux et le nom des fichiers traces associés. Le script de base permettant le rejeu des flux TCP dans une trace réelle doit comporter les étapes suivantes :

Récupération du fichier info le nombre de flux à rejouer,
le début, les RTT et les pertes pour chacun des flux TCP.
Paramètres de la simulation (délais, files d'attente et capacités des liens)
Création des générateurs de trafic

Création des nœuds
Création des liens

Création des agents et sources émetteurs (les paquets de tailles définies dans le fichier de trace par flux)
Création des agents récepteurs
Connexion émetteurs/récepteurs

Description du scénario

Procédure de fin
Lancement de la simulation

Pour la définition de la topologie de simulation nécessaire, les pertes représentent les principales caractéristiques à reproduire dans le rejeu de la trace puisque les mécanismes de TCP sont basés sur une réponse prédéfinie aux pertes. De plus, le temps d'aller-retour ou RTT est un paramètre important pour les mécanismes de contrôle de congestion et pour définir le profil du trafic. Ainsi, en respectant les pertes et les RTT des flux simulés, nous améliorons le réalisme des simulations [Park 1996].
A noter que cette approche ne se focalise pas sur des topologies réseaux très complexes. En fait, cette topologie est la plus simple qui soit capable de reproduire aussi précisément que possible les taux de pertes et les RTT moyens observés dans les traces de trafic réelles [Owezarski 2004]. Pour cela, l'analyse se fait pour chaque flux de la trace originale afin de finalement dimensionner les files d'attente et capacités moyennes des liens de l'environnement de simulation. De même, les délais de chacun des liens sont fixés de façon à respecter en moyenne les RTT mesurés pour les flux [Owezarski 2004].
Pour résumer, les éléments caractéristiques extraits de la trace sont les suivants :

- les dates relatives des débuts de flux,
- les tailles de paquets à l'intérieur de chaque flux,
- la durée de chaque flux,
- le temps d'aller retour (RTT) moyen pour chaque flux,
- le taux de perte expérimenté par chacun des flux,

– le débit moyen obtenu par chacun des flux.

Pour limiter la complexité de la topologie de simulation, et en se basant sur les analyses des taux de pertes, seulement six classes de taux de pertes différentes sont définies [Owezarski 2004] dans le Tableau 4.13. Avec les informations extraites des traces originales, la capacité et la taille de la file d'attente sont déduites pour chacun des liens de la topologie de simulation où le flux devra être transmis, en fonction de la classe à laquelle le flux appartient.

Classe	Taux de perte de la classe (%)
Cl_0	0
Cl_{10}	0-10
Cl_{20}	10-20
Cl_{30}	20-30
Cl_{50}	30-50
Cl_{100}	50-100

TAB. 4.13: Classes des flux dans les rejeux.

La capacité du lien pour la classe Cl_i (C_i) est calculée selon l'équation :

$$C_i = \frac{\sum_{n=1}^{N_f} d_n.Deb_n}{d_{trace}} \tag{4.1}$$

où :

– N_f est le nombre de flux de la classe Cl_i,
– d_n est la durée du flux n,
– Deb_n est la moyenne du débit du flux n,
– d_{trace} est la durée de la trace.

Ainsi, la taille de la file d'attente de la classe Cl_i (Q_i) est déduite selon l'équation :

$$Q_i = C_i(100 - \tau_{perte}) \tag{4.2}$$

où τ_{perte} est la moyenne du taux de perte (en %) obtenue par chaque flux de la classe.

Enfin, la topologie expérimentale qui sera utilisée pour rejouer la trace considérée est décrite dans la Figure 4.9 où RTT_{Cl_i} est la moyenne des RTT de l'ensemble des flux appartenant à la classe Cl_i.

4.7.2 Description des expérimentations

Dans cette section, nous présentons des résultats expérimentaux basés sur des traces réelles de trafic collectées au niveau du réseau RENATER[1]. Pour obtenir ces traces de trafic,

[1]RENATER est le Réseau National de télécommunications pour la Technologie, l'Enseignement et la Recherche

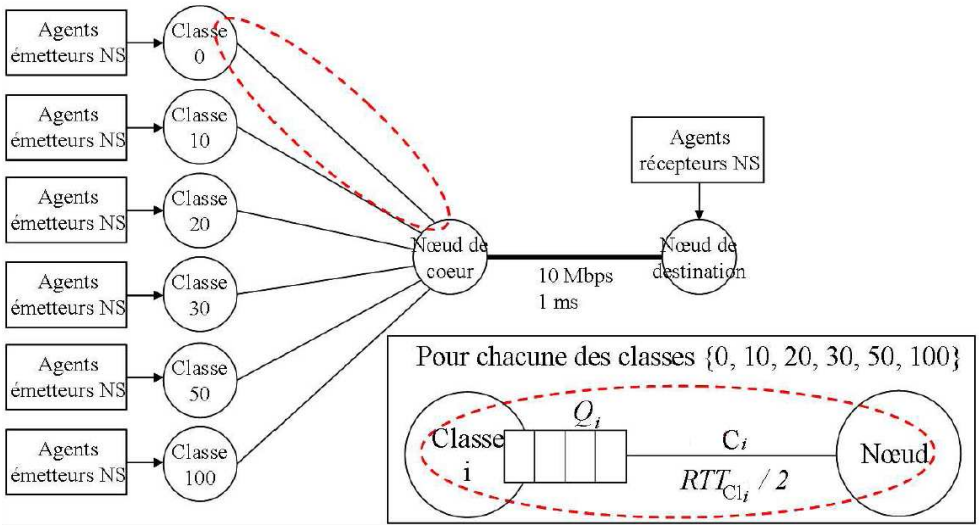


FIG. 4.9: Topologie des rejeux de traces.

les outils de capture sont installés dans la plateforme expérimentale "LaasNetExp" installée au LAAS-CNRS [Owezarski 2008]. Elle a été conçue pour permettre à la fois de réaliser des expérimentations en émulations réseau, mais aussi en environnement réel. La Figure 4.10 présente la plateforme LaasNetExp dans sa globalité. Pour remplir les besoins expérimentaux, LaasNetExp est complètement séparée du réseau opérationnel du LAAS afin d'éviter les perturbations mutuelles. LaasNetExp est reliée à l'Internet directement par RENATER. Actuellement, LaasNetExp est composée d'un serveur et de machines d'expérimentation fonctionnant sous différents systèmes d'exploitation et équipés de quatre interfaces Ethernet. Le réseau d'expérimentation réel est connecté à Ethernet avec des ports de capacités en Gigabits.

Pour nos expérimentations, une machine spécifique est choisie comme routeur dans la topologie de réseau adoptée. Les données entrantes et les données sortantes de cette machine sont captées par l'analyseur de réseau *Wireshark* [Orebaugh 2007]. Durant la capture, des attaques de type *UDP Flooding* sont envoyées d'une machine de l'Institut de Mont-de-Marsan de technologie (c.f. Figure 4.10) à travers le réseau RENATER. Les attaques sont générées par l'intermédiaire du logiciel TFN2K [Barlow 2000]. Les outils TFN modifiés permettent de générer des UDP/TCP/ICMP floodings en formes plates, rampes, variables et même asynchrones comme dans la Figure 4.11.

Une fois que la trace est obtenue, une analyse hors-ligne permet d'extraire des flux (TCP, UDP et d'autres) de la trace ainsi que leurs propriétés non triviales déjà mentionnées dans la section 4.7.1.
Notre étude théorique est basée sur un modèle mathématique contraignant de l'homogénéité des flux TCP émis par les sources. Par conséquent, dans nos expériences détaillées par la suite, nous avons envoyé du trafic TCP provenant de sources homogènes. Dans la topologie de rejeux définie dans la Figure 4.9, les flux analysés appartiennent à une seule classe de pertes.

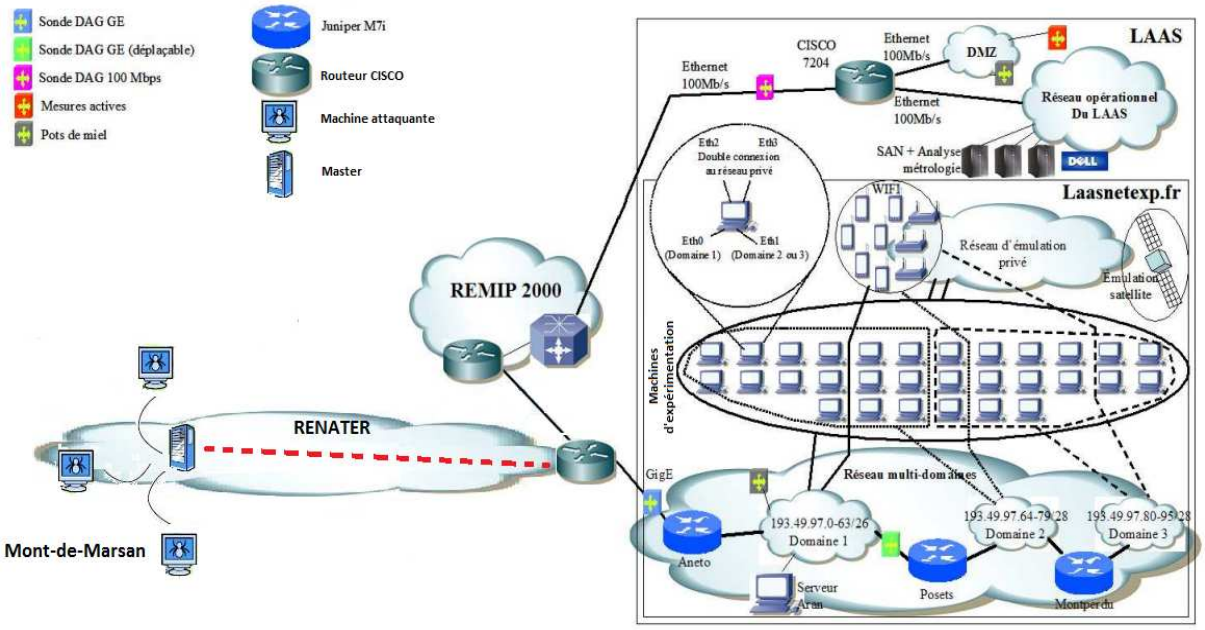


FIG. 4.10: la plateforme LaasNetExp au LAAS-CNRS.

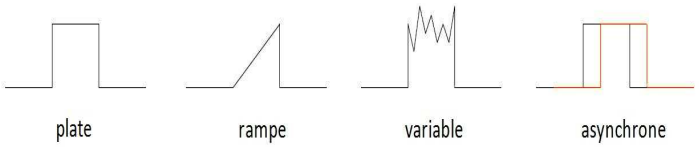


FIG. 4.11: Les formes des attaques générées par TFN modifiés.

4.7.3 1^{er} rejeu

Durant la capture de trafic, 6 flux TCP sont émis vers une destination dans LaasNetExp via le routeur choisi. Le trafic d'attaques est envoyé de Mont-de-Marsan à débit constant égal à 100Kbps en séries de courtes et longues durées. Les courtes attaques sont de 7 secondes chacune avec 5 secondes entre deux attaques consécutives, les longues attaques persistent 4 minutes chacune avec une durée variable de 1 à 3 minutes entre elles.
Dans la Figure 4.12, nous montrons le trafic capturé entrant au buffer du routeur. Ce trafic comprend le trafic TCP et l'attaque. Une simple observation de l'évolution du trafic ne permet pas de confirmer l'existence d'une attaque. Après des analyses des traces permettant l'extraction des caractéristiques moyennes des flux TCP [Owezarski 2004], les observateurs sont mis en œuvre au niveau du routeur pour estimer l'attaque passant dans son buffer.
La méthodologie proposée dans [Owezarski 2004] montre que la taille moyenne des paquets TCP envoyées est égale à 1500 octets. Les valeurs moyennes des capacités de liens et les RTT sont ainsi déterminées pour chaque flux. Le taux d'envoi moyen de paquets TCP calculé est égal à 0.122Mbps avec une latence de 94.3ms dans les liens menant au routeur.

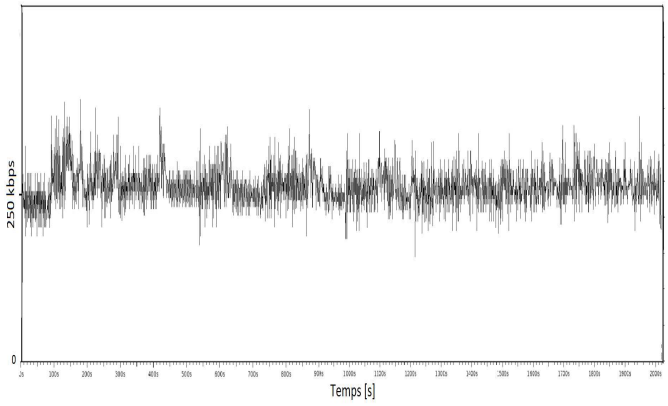


FIG. 4.12: Trafic capturé au niveau du routeur du rejeu n^o1.

Les caractéristiques des sources et des liens en amont du routeur sont définies dans la topologie de simulation. Durant la capture comme pour le rejeu sous NS-2, le routeur est configuré pour régler la longueur de file d'attente en utilisant le mécanisme de Token Bucket Filter (TBF). TBF est un algorithme souvent utilisé pour limiter le débit et réguler la longueur de file d'attente au niveau du routeur. Cependant, pour concevoir l'observateur proposé, nous devons calculer le point d'équilibre. Ce dernier est défini pendant le comportement normal du réseau TCP avant le début des attaques. Les valeurs de la taille de fenêtre de congestion et la longueur de file d'attente à l'équilibre (W_0 et q_0) sont déterminées à partir des valeurs moyennes autour desquelles $W(t)$ et $q(t)$ respectivement oscillent dans des intervalles consécutifs de 5 secondes. Les valeurs de p_0 et R_0 correspondant à l'équilibre de la probabilité d'éjection des paquets et au temps d'aller retour sont calculées à partir du système (A.1) dans l'annexe A.
Pour résumer, les caractéristiques du routeur ainsi que le point d'équilibre sont présentés dans le Tableau 4.14.

Point d'équilibre	
W_0	5.8 paquets
q_0	17.08 paquets
p_0	0.05945
R_0	4.037 s

Configuration du routeur	
C	0.15Mbps
q_{max}	20 paquets

TAB. 4.14: Configurations du rejeu n^o1.

Le script de rejeu définissant la topologie est présenté dans l'Annexe **??**. Les observateurs de Luenberger IOD et DD et les observateurs glissants d'ordre 2 sont construits puis analysés dans les parties suivantes.

4.7.3.1 Avec l'observateur minimal IOD

En tenant en compte la topologie considérée dans la Figure 2.6, les simulations sur Matlab/Simulink et NS-2 montrent la rapidité de l'observateur pour toutes valeurs de k puisque les pôles ont été placés conformément au pôle dominant. Selon le même principe, les pôles de l'observateur minimal IOD sont calculés pour la topologie du rejeu n^o1. Ayant un pôle dominant égal à -0.114, le temps de convergence est de 26.5s. Les gains d'observation pour $k = 1$ sont obtenus relativement aux autres pôles placés à des valeurs inférieures au pôle dominant, soient -0.3. L'observateur dans la Figure 4.13 arrive à détecter la présence des anomalies mais engendre de fortes oscillations en s'écartant des bonnes valeurs de l'anomalie. La moyenne observée est de 40 paquets/s au lieu du débit d'attaque de 15 paquets/s.
Ces oscillations augmentent avec l'ordre k pour arriver à $k = 3$ où les graphes deviennent

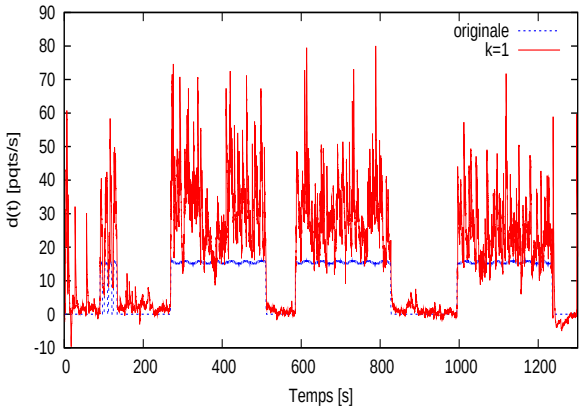


FIG. 4.13: Estimation de d(t) avec l'observateur IOD minimal.

trop bruités. Même en fixant des pôles dominants plus grands, la valeur moyenne observée est toujours très grande. L'observateur minimal, ne prenant pas en compte l'estimation de la taille de la file d'attente $q(t)$, semble insuffisant pour estimer l'anomalie $d(t)$ avec les flux TCP à caractéristiques réelles. Ce type d'observateur appliqué aux autres rejeux montre aussi les mêmes comportements. Pour chacune des expérimentations sur le rejeu, l'observateur basé sur l'approche DD est simulé, ainsi que l'observateur glissant d'ordre 2 comme il a montré des meilleures reconstructions des anomalies par rapport à l'observateur glissant d'ordre 1.

4.7.3.2 Avec l'observateur DD

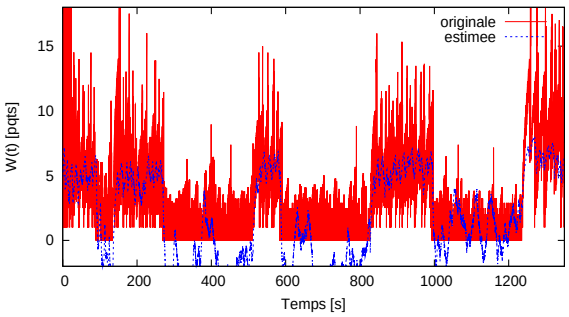
L'application du Théorème 2.5 sur les configurations de la topologie de ce rejeu présentées dans le Tableau 4.14 aboutit aux gains d'observation obtenus selon les valeurs de k :

$$\text{pour } k = 1, \quad L = [0.1869, 2.8376, 2.8951, 0.855]^T; \tag{4.3}$$

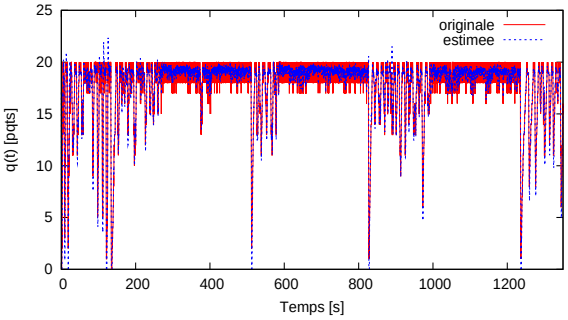
$$\text{pour } k = 2, \quad L = [0.0449, 3.4684, 5.1122, 3.1059, 0.7739]^T; \tag{4.4}$$

$$\text{pour } k = 3, \quad L = [-0.0424, 4.6982, 9.3513, 8.6321, 4.3112, 0.9262]^T. \tag{4.5}$$

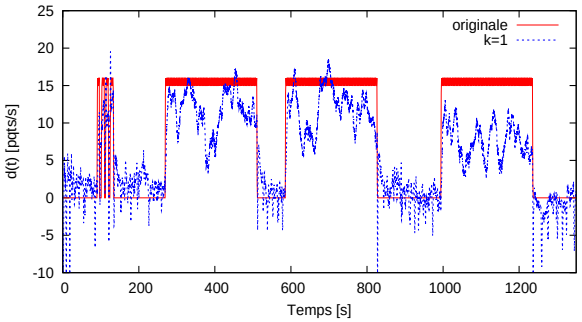
Dans toutes les expérimentations, nous allons présenter les résultats des observateurs avec



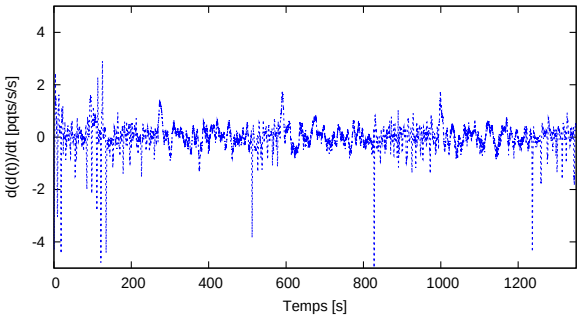
(a) La fenêtre de congestion $W(t)$.



(b) La longueur de la file d'attente $q(t)$.

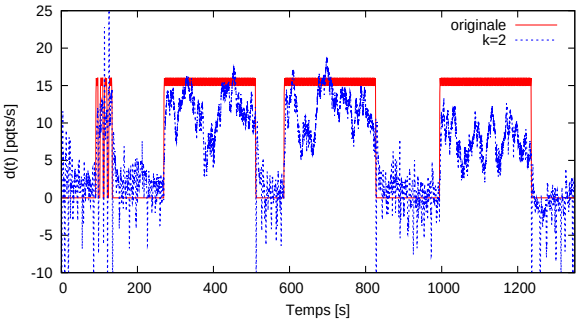


(c.1) $d(t)$

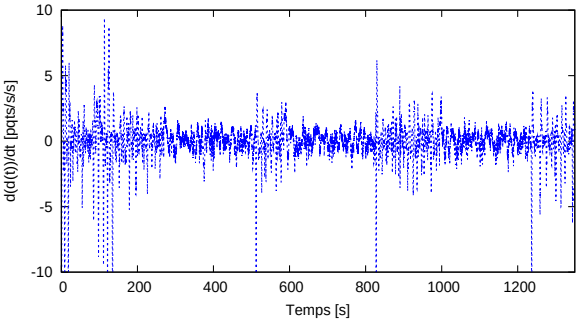


(c.2) $\dot{d}(t)$

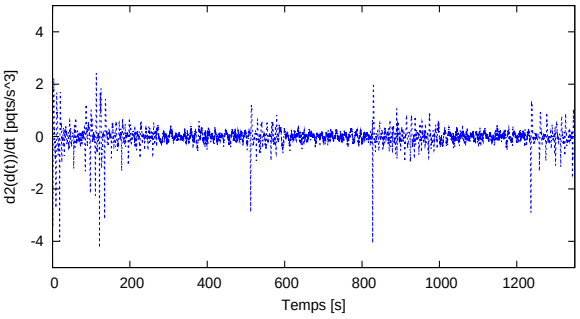
(c) L'attaque et sa dérivée pour $k = 1$.



(d.1) $d(t)$



(d.2) $\dot{d}(t)$



(d.3) $\ddot{d}(t)$

(d) L'attaque et ses dérivées pour $k = 2$.

FIG. 4.14: Estimations DD du rejeu $n^o 1$.

$k = 1$ et $k = 2$ appliqués aux flux à caractéristiques réelles. Ces derniers produisent des oscillations qui deviennent sévères pour $k = 3$, perturbant par conséquent l'observation des anomalies et ses dérivées. Les performances des observateurs sont montrées dans la Figure 4.14. L'analyse de l'observation pour chacune des séries d'attaques est présentée quantitativement dans les Tableau 4.15. 4 séries de durée entre 6 et 7.5s et 3 séries de durée de 4 minutes ont été envoyées de Mont-de-Marsan.

Les valeurs dans le Tableau 4.15 montrent que l'observateur construit avec $k = 2$ réduit, par rapport à $k = 1$, le temps de convergence pour la reconstruction de la plupart des séries d'attaques, ainsi que l'erreur moyenne d'observation en augmentant légèrement l'écart de la valeur moyenne de l'erreur. Les oscillations engendrées par $k = 2$ sont plus considérables lors de l'observation de $\dot{d}(t)$ comme nous pouvons déduire du Tableau 4.16a. La dérivée seconde est aussi bien observée dans la Tableau 4.16b avec des erreurs moyennes et des écarts très petits.

série	Temps de convergence [s]		Erreur moyenne [paquets/s]		Écart-type [paquets/s]	
	$k = 1$	$k = 2$	$k = 1$	$k = 2$	$k = 1$	$k = 2$
1	5.18	4.1	-4.08	-3.79	0.26	0.47
2	4.2	4.1	-2.73	-2.5	0.63	0.55
3	4.27	4.47	-4.25	-4.38	0.38	0.55
4	2.8	5	-1.98	-1.42	0.77	0.57
5	6.98	5.9	-3.4	-3.35	2.8	2.82
6	8.4	5.2	-3.13	-3.2	2.91	2.9
7	5.7	4.7	-7.4	-7.33	2.29	2.29

TAB. 4.15: Les caractéristiques de l'observation des attaques du rejeu n^o1 par les observateurs DD.

Nous remarquons que dans la Figure 4.14, la reconstruction des attaques de plus longues durées est perturbée : dans l'intervalle de temps [350, 430]s de la série 5, dans [620, 750]s de la série 6 et durant toute la série 7. Nous expliquons ces phénomènes par la mauvaise observation de la fenêtre de congestion durant les intervalles de temps correspondants. En dépit des perturbations, les observateurs arrivent à observer 12 paquets/s de flux d'attaques dans les séries 5 et 6 et 8 paquets/s dans la série 7 parmi les 15 paquets/s envoyés originalement.

Pour la rapidité de détection de ces dernières attaques, les faux négatifs et positifs générés pour chacune des séries sont montrés dans le Tableau 4.17. Nous pouvons voir dans les graphes de la Figure 4.14 qu'en l'absence de l'anomalie, l'observateur présente de fortes oscillations autour de 0.5 paquets/s. Nous définissons donc les faux négatifs au moment de l'apparition réelle de l'attaque jusqu'à l'observation d'une anomalie supérieure au seuil de 0.5 paquets/s. De même, les faux positifs sont déterminés au moment de la disparition réelle

série	Erreur moyenne [paquets/s]		Écart-type [paquets/s]	
	$k=1$	$k=2$	$k=1$	$k=2$
1	1.28	2.23	0.05	0.36
2	0.72	0.79	0.1	0.4
3	0.1	-0.02	0.11	0.33
4	0.7	0.35	0.88	0.24
5	0.01	0.03	0.35	0.84
6	0	0.01	0.32	0.67
7	0.01	0.02	0.35	0.69

(a) La dérivée première $\dot{d}(t)$.

série	Erreur moyenne [paquets/s]	Écart-type [paquets/s]
1	0.17	0.11
2	0.01	0.1
3	-0.04	0.09
4	0.05	0.06
5	0	0.16
6	0	0.15
7	0	0.15

(b) La dérivée seconde $\ddot{d}(t)$.

TAB. 4.16: Les caractéristiques de l'observation des dérivées des signaux d'attaques du rejeu n^o1 par les observateurs DD.

série	$k=1$	$k=2$
1	0.27	0.6
5	0.36	0.33
6	0.33	0.2
7	1.7	0.7

(a) Faux négatifs.

série	$k=1$	$k=2$
1	5.45	5.45
2	3.1	3.1
3	4.7	4.7
4	1.5	1.1
5	11.8	8.5
6	11.5	11.1
7	0.7	5

(b) Faux positifs.

TAB. 4.17: Les faux négatifs et positifs induits par les observateurs DD pour le rejeu n^o1 (exprimés en secondes).

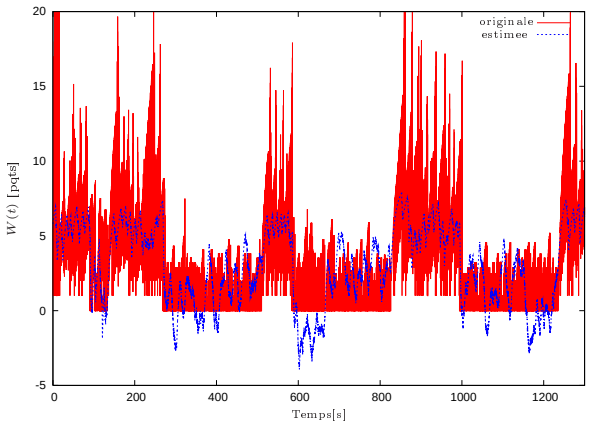
de l'attaque jusqu'à l'instant où l'observation rejoint ce seuil. Nous notons dans le Tableau 4.17 l'absence de faux négatifs avant la reconstruction des séries d'attaques 2, 3 et 4. Cela est dû aux faux positifs relatifs aux séries 1, 2 et 3 qui continuent sur toutes les périodes entre les séries. En effet les intervalles de temps qui séparent deux séries d'attaques consécutives ne sont pas suffisants pour assurer la convergence de l'observateur. De plus, de la Figure 4.14 nous pouvons conclure que des faux positifs apparaissent aussi après la série 4 dans

[200, 220]s et après la série 6 entre [860, 880]s à cause des estimations de $W(t)$ qui ne suivent pas la moyenne réelle.

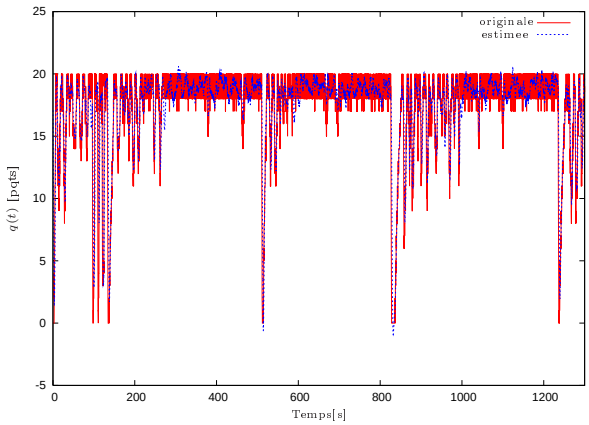
Dans les conditions du rejeu n^o1, l'observateur DD avec $k = 2$ est le meilleur pour la reconstruction de l'anomalie et ses dérivées en terme de temps de convergence et d'erreurs moyennes d'observation. Durant la reconstruction de $d(t)$, cet observateur produit des écarts proches de l'observateur avec $k = 1$ mais les écarts deviennent considérables durant la reconstruction de $\dot{d}(t)$.

4.7.3.3 Avec l'observateur glissant d'ordre 2

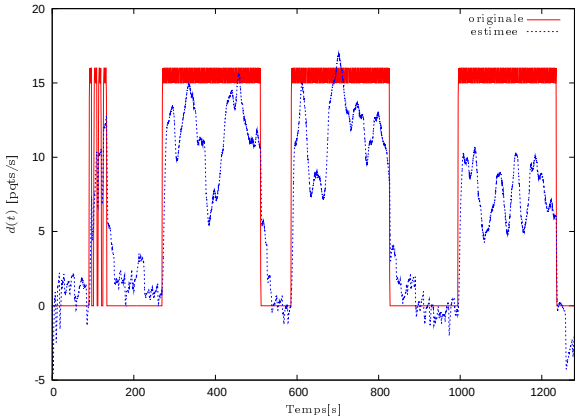
L'observateur glissant d'ordre 2 qui vérifie le Théorème 3.3 appliqué à la topologie du rejeu n^o1 donne le gain d'observation $L = [2.9795, 2.6934]^T$.



(a) La fenêtre de congestion $W(t)$.



(b) La longueur de la file d'attente $q(t)$.



(c) L'attaque $d(t)$.

FIG. 4.15: Estimations du rejeu n^o1 avec l'observateur glissant.

Les résultats de l'obervation de la trace de trafic sont montrés dans la Figure 4.15. L'analyse des reconstructions des attaques est résumée dans le Tableau 4.18. La convergence de l'observation par les modes glissants est plus lente que celle des observateurs DD. Les erreurs de reconstruction des attaques sont par conséquent plus grandes pour les attaques de courtes

série	Temps de convergence [s]	Erreur moyenne [paquets/s]	Écart-type [paquets/s]
1	6.01	-7.7	0.01
2	6.3	-4.8	0.1
3	1.7	-4.74	0.11
4	2.3	-2.79	0.28
5	11.7	-3.4	2.44
6	9.3	-3.13	2.46
7	11	-7.38	1.73

TAB. 4.18: Les caractéristiques de l'observation des attaques du rejeu n^o1 par l'observateur glissant.

durées. Pour les 3 dernières attaques qui sont de longues durées, les erreurs sont très proches des observateurs DD. Nous pouvons ainsi voir dans les graphes de la Figure 4.15c que la reconstruction des attaques représente des perturbations durant les mêmes intervalles qu'avec les observateurs DD dues à l'estimation erronée de la fenêtre de congestion $W(t)$ dans la Figure 4.15a.

série	Faux négatifs
1	0.1
5	0.4
6	0.6
7	2.2

(a)

série	Faux positifs
1	5.45
2	3.1
3	4.7
4	1.8
5	16.8
6	17.6
7	1

(b)

TAB. 4.19: Les faux négatifs et positifs induits par l'observateur glissant pour le rejeu n^o1 (exprimés en secondes).

Les faux négatifs et positifs présentés dans le Tableau 4.19 sont déterminés à partir du seuil fixé à 0.5 paquets/s. Nous remarquons qu'il n'y a pas de faux négatifs pour les séries 2, 3 et 4 vu que les intervalles entre les attaques ne sont pas suffisamment longs pour assurer la convergence de l'observateur. Les faux positifs continuent à apparaître entre les attaques de courtes durées montrant des anomalies continuellement présentes. Les faux négatifs et

positifs déduits de l'observateur glissant persistent plus longtemps qu'avec les observateurs DD parce que ces derniers nécessitent dans ce rejeu un temps de convergence plus grand.

4.7.4 $2^{ème}$ rejeu

Cette trace de réseau est captée avec les mêmes conditions que la précédente trace mais en présence d'une anomalie triangulaire. Le routeur est configuré comme précédemment sous TBF avec un débit de 150Kbps. 6 flux TCP sont émis vers le routeur. Le trafic d'attaques est envoyé de Mont-de-Marsan en forme de rampe partant de 0Kbps à 100Kbps (chaque paquet de 950 octets). Les attaques persistent 4 minutes chacune avec un intervalle de temps variable de 1 à 3 minutes entre elles. Dans la Figure 4.16, le trafic capturé entrant au buffer du routeur est présenté. L'existence d'une attaque ne peut pas être déduite de l'observation

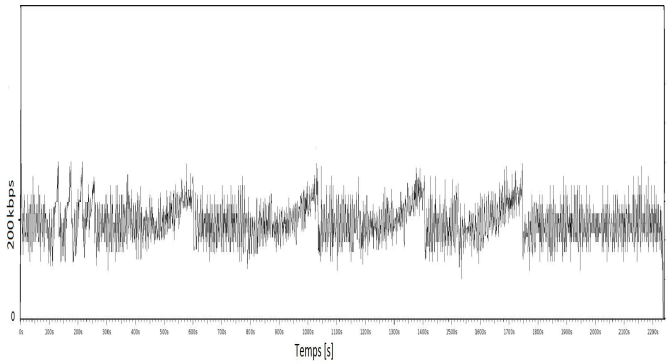


FIG. 4.16: Trafic capturé au niveau du routeur pour le rejeu n^o2.

de l'évolution du trafic. L'analyse hors-ligne de la trace par les procédures décrites dans la section 4.7.1 [Owezarski 2004], montre un taux d'envoi de paquets TCP égal à 0.125Mbps avec des paquets de taille moyenne de 1500 octets. La latence est de 47ms dans les liens menant au routeur.
Pour le point d'équilibre, la valeur de q_0 est modifiée par rapport à la première trace. Le point d'équilibre est détaillé dans le Tableau 4.20.

Point d'équilibre	
W_0	5.8 paquets
q_0	16.66 paquets
p_0	0.05945
R_0	4.037s

TAB. 4.20: Configurations du rejeu n^o2.

4.7.4.1 Avec l'observateur DD

Comme pour la première expérimentation, les gains d'observation selon les valeurs de k.

$$\text{pour } k=1, \quad L=[0.1869, 2.8376, 2.8951, 0.855]^T; \tag{4.6}$$
$$\text{pour } k=2, \quad L=[0.0449, 3.4684, 5.1122, 3.1059, 0.7739]^T; \tag{4.7}$$
$$\text{pour } k=3, \quad L=[-0.0424, 4.6982, 9.3513, 8.6321, 4.3112, 0.9262]^T. \tag{4.8}$$

L'application de ces gains sur la trace est montrée par les graphes de la Figure 4.17. Pour chacune des séries d'attaques, nous analysons le temps de convergence, l'erreur moyenne entre les valeurs observées et réelles, ensuite l'écart-type autour de l'erreur obtenus durant la reconstruction des attaques et leurs dérivées première et seconde. Dans le Tableau 4.21, l'observateur DD avec $k=2$ montre une convergence beaucoup plus lente qu'avec $k=1$. De plus, les erreurs de reconstruction des attaques dans le Tableau 4.21 avec $k=2$ sont plus petites que celles avec $k=1$ mais en restant très proches. Les dérivées premières dans le Tableau 4.22a ne sont pas bien observées vu les erreurs qui sont grandes par rapport à la valeur originale égale à 0.04 paquets/s^2. La dérivée seconde est bien observée, proche de 0 dans la Tableau 4.22b. Nous pouvons remarquer les perturbations durant l'estimation de

série	Temps de convergence [s]		Erreur moyenne [paquets/s]		Écart-type [paquets/s]	
	$k=1$	$k=2$	$k=1$	$k=2$	$k=1$	$k=2$
1	2.2	3.6	2.12	1.93	1.69	1.9
2	3	40.2	-3.52	-3.32	1.41	1.68
3	3.1	35.1	-3.3	-3.2	1.4	1.6

TAB. 4.21: Les caractéristiques de l'observation des attaques du rejeu n^o2 par les observateurs DD.

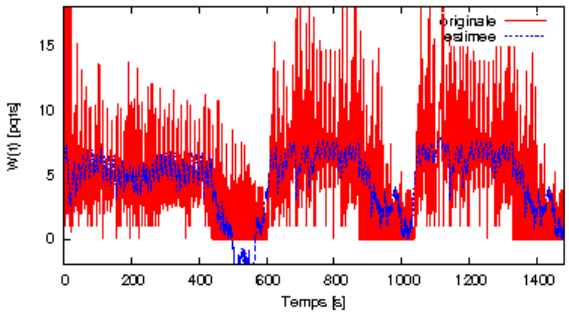
série	Erreur moyenne [paquets/s]		Écart-type [paquets/s]	
	$k=1$	$k=2$	$k=1$	$k=2$
1	0.03	0	0.21	0.64
2	0	-0.05	0.19	0.65
3	-0.1	-0.01	0.19	0.68

(a) La dérivée première $\dot{d}(t)$.

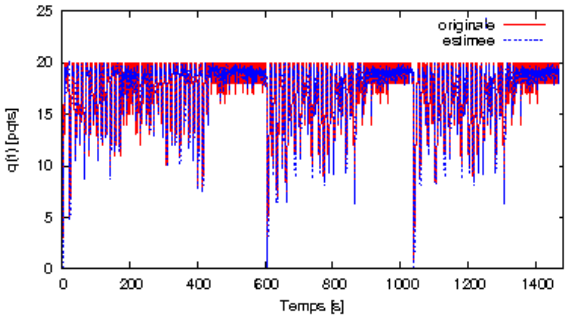
série	Erreur moyenne [paquets/s]	Écart-type [paquets/s]
1	-0.01	0.15
2	0	0.15
3	0.01	0.15

(b) La dérivée seconde $\ddot{d}(t)$.

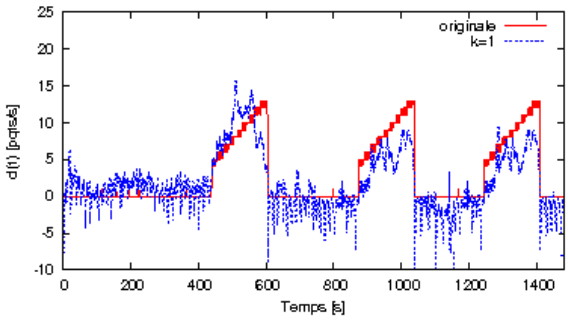
TAB. 4.22: Les caractéristiques de l'observation des dérivées des signaux d'attaques du rejeu n^o2 par les observateurs DD.



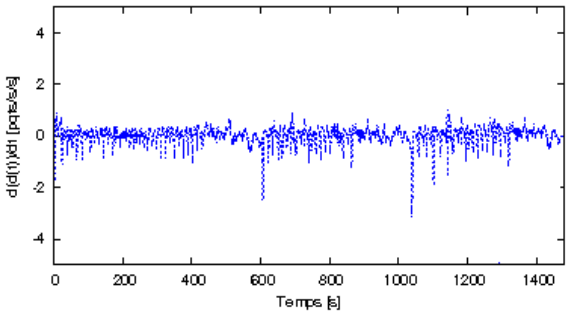
(a) La fenêtre de congestion $W(t)$.



(b) La longueur de la file d'attente $q(t)$.

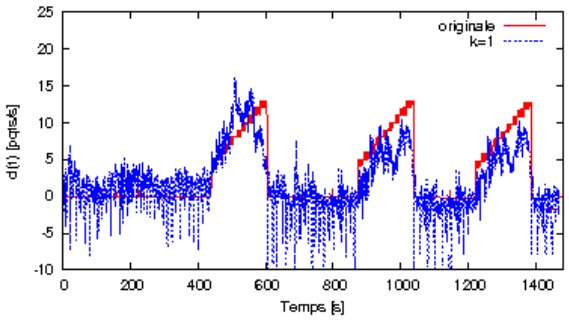


(c.1) $d(t)$

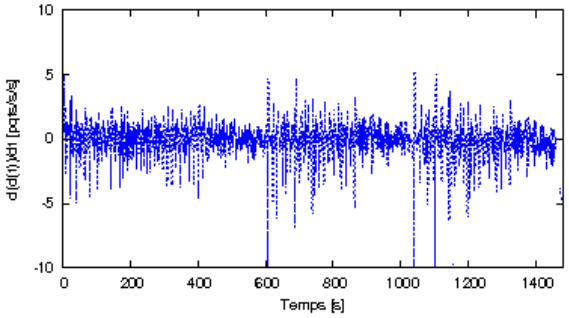


(c.2) $\dot{d}(t)$

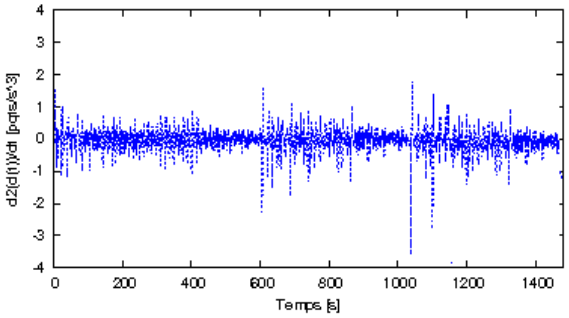
(c) L'attaque et sa dérivée pour $k = 1$.



(d.1) $d(t)$



(d.2) $\dot{d}$



(d.3) $\ddot{d}(t)$

(d) L'attaque et ses dérivées pour $k = 2$.

FIG. 4.17: Estimations DD du rejeu n^o2.

la fenêtre de congestion $W(t)$ (c.f. Figure 4.17a) durant les intervalles de temps [565, 600]s, [875, 1032]s et [1310, 1340]s, affectant ainsi la reconstruction des 3 séries d'attaques. Malgré ces perturbations, les observateurs DD arrivent à reconstruire le débit de 17 paquets/s de l'attaque pour la série 1 et 12 paquets/s pour les séries 2 et 3 au lieu du débit original égal à 15 paquets/s.

Durant la détection des attaques, les faux négatifs et positifs pour chacune des séries sont déterminés dans le Tableau 4.23 en comparant avec la période avant le début des attaques où nous avons des oscillations d'amplitudes entre 2.5 et -2 paquets/s et de moyenne 0.6 paquets/s. Nous remarquons que les faux négatifs et positifs sont bien réduits avec $k = 1$. Les oscillations procréées par $k = 2$ rendent difficile la détermination des faux surtout les positifs.

série	$k = 1$	$k = 2$
1	0.51	0.7
2	2.2	31
3	2.6	20

(a) Faux négatifs.

série	$k = 1$	$k = 2$
1	0.93	5.7
2	0.7	5.2
3	0.5	4.4

(b) Faux positifs.

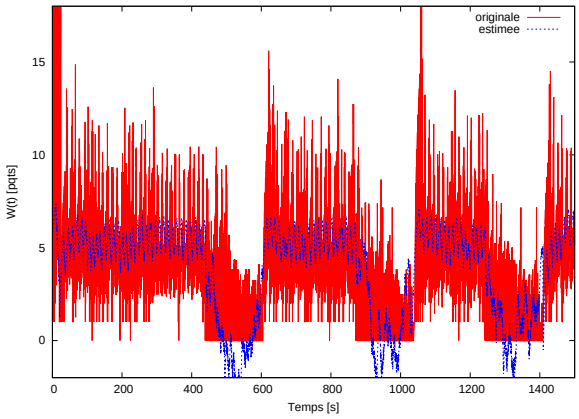
TAB. 4.23: Les faux négatifs et positifs induits par les observateurs DD pour le rejeu n^o2 (exprimés en secondes).

Nous pouvons conclure sur les observateurs DD que pour cette expérimentation, $k = 1$ est le plus favorable pour la reconstruction et la détection des attaques.

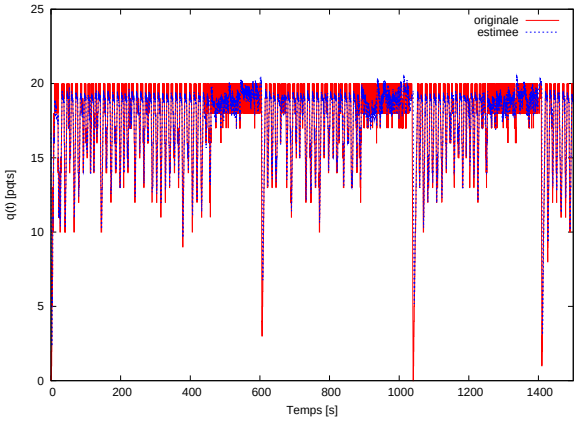
4.7.4.2 Avec l'observateur glissant d'ordre 2

Comme les caractéristiques du routeur dans cette trace sont les mêmes que la trace précédente avec le même nombre de sources TCP dans la topologie, l'observateur glissant d'ordre 2 est identique au premier où $L = [2.9795, 2.6934]^T$. Les résultats du rejeu de cette trace montrés dans la Figure 4.18 sont analysés quantitativement dans le Tableau 4.24. Sous les mêmes conditions d'expérimentation qu'avec les observateurs DD, nous pouvons remarquer que la convergence de l'observation par les modes glissants est lente par rapport à celle de l'observateur DD pour $k = 1$. Cependant, les erreurs moyennes de reconstruction des attaques sont très petites en comparant avec les observateurs DD avec des écart-types plus grands. Dans les graphes de la Figure 4.18c, la reconstruction des attaques représente des perturbations durant les mêmes intervalles qu'avec les observateurs DD. L'observateur glissant reconstruit 16 paquets/s pour la série 1, 14 paquets/s pour la série 2 et 13 pour la série 3 au lieu du débit original égal à 15 paquets/s.

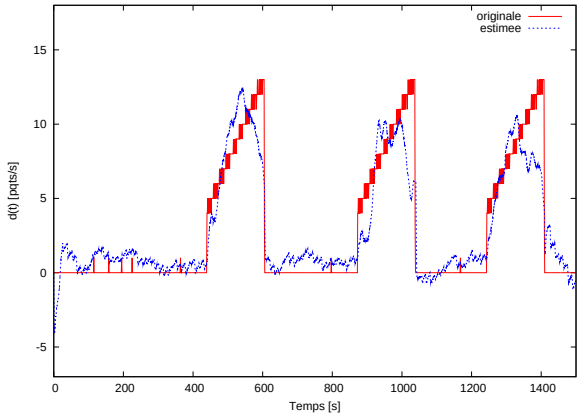
En l'absence d'anomalies, les observateurs glissants reconstruisent des signaux oscillants entre 0 et 1.2 paquets/s avec une moyenne de 0.45 paquets/s. Cette moyenne est prise comme seuil pour la détermination des faux négatifs et positifs résultants. Dans le Tableau 4.25, nous pouvons remarquer la détection rapide de la présence des attaques vu les faux négatifs de



(a) La fenêtre de congestion $W(t)$.



(b) La longueur de la file d'attente $q(t)$.



(c) L'attaque $d(t)$.

FIG. 4.18: Estimations du rejeu n^o2 avec l'observateur glissant.

série	Temps de convergence [s]	Erreur moyenne [paquets/s]	Écart-type [paquets/s]
1	6.9	0.82	1.56
2	5.67	-0.93	1.99
3	2.7	-1.44	2.37

TAB. 4.24: Les caractéristiques de l'estimation des attaques du rejeu n^o2 par l'observateur glissant.

courtes durées. Ces faux négatifs sont plus courts que ceux induits par les observateurs DD, par contre les faux positifs sont plus longs que l'observateur DD avec $k = 1$ vu le temps de convergence qui est lent pour l'observateur glissant.

série	Faux négatifs
1	0.2
2	1.5
3	2.2

(a)

série	Faux positifs
1	4.3
2	5.4
3	19.1

(b)

TAB. 4.25: Les faux négatifs et positifs induits par l'observateur glissant pour le rejeu n^o2 (exprimés en secondes).

4.7.5 $3^{ème}$ rejeu

Durant la trace collectée sur notre routeur, 9 flux TCP sont envoyés dont la taille moyenne des paquets est égale à 1455 octets. Le taux d'envoi moyen de paquets est égal à 0.225Mbps avec une latence de 25.8ms.
Le trafic d'attaques est à débit constant égal à 100Kbps (chaque paquet de 950 octets). Les courtes attaques sont de 20 secondes chacune avec 20 à 30 secondes entre elles, les longues attaques sont de 2 minutes chacune avec une durée variable de 1 à 2 minutes entre elles.
Les configurations du routeur ainsi que le point d'équilibre sont dans le Tableau 4.26.

Point d'équilibre	
W_0	5.0318 paquets
q_0	18.695 paquets
p_0	0.0789
R_0	2.6359s

Configuration du routeur	
C	0.2Mbps
q_{max}	20 paquets

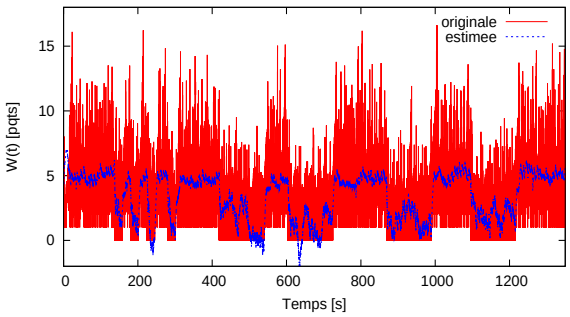
TAB. 4.26: Configurations du rejeu n^o3

4.7.5.1 Avec l'observateur DD

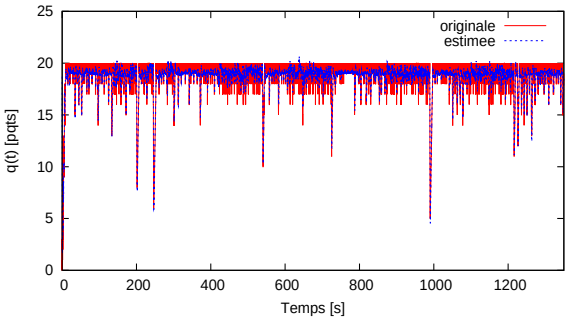
En appliquant le Théorème 2.5 aux paramètres de la topologie présentés dans le Tableau 4.26, nous obtenons les gains d'observation suivants dont les performances sont montrées dans la Figure 4.19.

$$\text{pour } k = 1, \quad L = [0.1979, 2.8355, 2.2711, 0.4804]^T; \tag{4.9}$$
$$\text{pour } k = 2, \quad L = [0.0895, 3.6022, 5.0355, 2.4116, 0.4977]^T; \tag{4.10}$$
$$\text{pour } k = 3, \quad L = [-0.0022, 4.6664, 9.2291, 7.3622, 3.1628, 0.6132]^T. \tag{4.11}$$

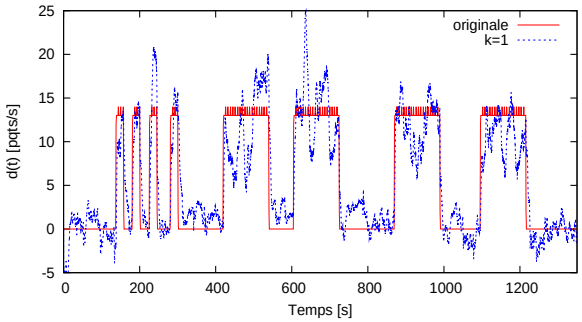
Nous pouvons remarquer dans les graphes de la Figure 4.19, les imperfections durant les estimations de la fenêtre de congestion simultanées avec des changements brusques durant la reconstruction du profil des attaques. Comme par exemple, durant la série 3 d'attaque,



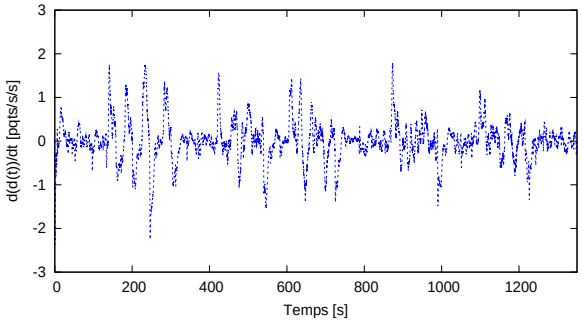
(a) La fenêtre de congestion $W(t)$.



(b) La longueur de la file d'attente $q(t)$.

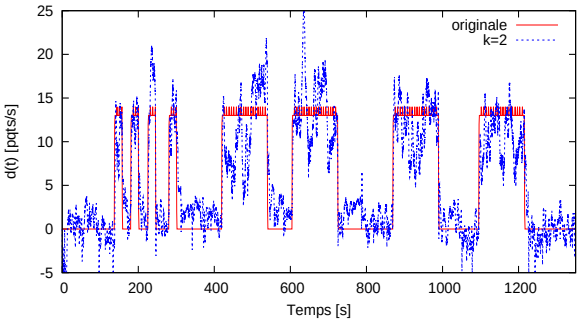


(c.1) $d(t)$

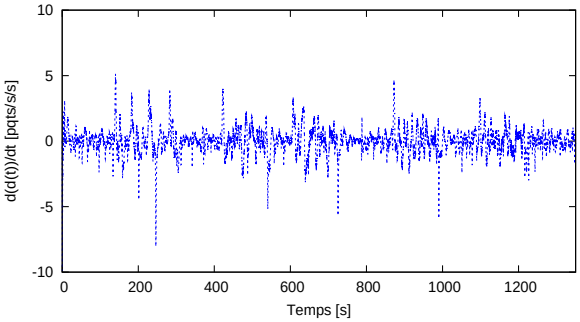


(c.2) $\dot{d}(t)$

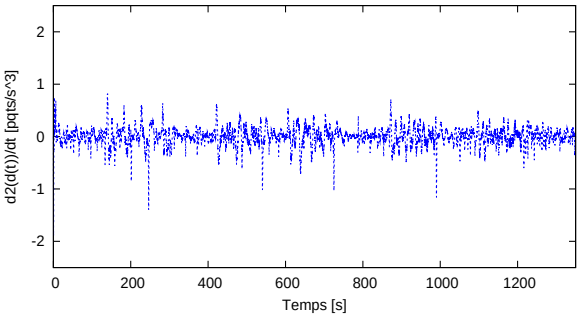
(c) L'attaque et sa dérivée pour $k = 1$.



(d.1) $d(t)$



(d.2) $\dot{d}(t)$



(d.3) $\ddot{d}(t)$

(d) L'attaque et ses dérivées pour $k = 2$.

FIG. 4.19: Estimations DD du rejeu n^o3.

dans les intervalles de temps [455, 480]s au milieu de la série 5, [627, 644]s de la série 6, et [935, 950]s de la série 7.

série	Temps de convergence [s]		Erreur moyenne [paquets/s]		Écart-type [paquets/s]	
	$k=1$	$k=2$	$k=1$	$k=2$	$k=1$	$k=2$
1	3.9	2.93	-4.43	-4.6	1.7	2.5
2	4.4	2.5	-2.38	-2.69	1.11	1.35
3	10.07	3.4	4.79	3.23	0.7	1.41
4	4.07	2.5	-1.78	-2.06	2.43	2.29
5	4.3	3.1	-2.2	-1.33	4.44	4.67
6	6.67	3.23	-1.24	-1.33	3.94	4.12
7	3.7	2.7	-2.09	-3.21	2.63	2.91
8	4.3	2.5	-4.6	-3.65	2.31	2.54

TAB. 4.27: Les caractéristiques de l'observation des attaques du rejeu n^o3 par les observateurs DD.

série	Erreur moyenne [paquets/s]		Écart-type [paquets/s]	
	$k=1$	$k=2$	$k=1$	$k=2$
1	0.64	0.3	0.38	1.27
2	0.54	0.1	0.38	0.85
3	0.67	-0.51	0.52	0.68
4	0.68	0.26	0.39	0.94
5	0.12	0.08	0.43	1.02
6	0.05	-0.01	0.55	1.19
7	0.07	0	0.41	1
8	0.07	0.03	0.36	0.82

(a) La dérivée première $\dot{d}(t)$.

série	Erreur moyenne [paquets/s]	Écart-type [paquets/s]
1	-0.08	0.25
2	-0.09	0.16
3	-0.23	0.12
4	-0.09	0.17
5	-0.01	0.19
6	-0.01	0.21
7	-0.01	0.19
8	-0.01	0.15

(b) La dérivée seconde $\ddot{d}(t)$.

TAB. 4.28: Les caractéristiques de l'observation des dérivées des signaux d'attaques du rejeu n^o3 par les observateurs DD.

Les valeurs dans le Tableau 4.27 montrent qu'en dépit des imperfections des estimations de la

fenêtre de congestion, l'observateur pour $k = 2$ réduit significativement, par rapport à $k = 1$, le temps de convergence pour la reconstruction de la plupart des séries d'attaques. L'erreur moyenne est diminuée avec $k = 2$ par rapport à $k = 1$ pour certaines séries d'attaques, mais avec une augmentation de l'écart de la valeur moyenne pour toutes les attaques. L'observateur reconstruit des attaques de débit entre 10 et 14 paquets/s. Les dérivées $\dot{d}(t)$ et $\ddot{d}(t)$ sont bien observées avec $k = 2$, par contre les oscillations sont plus considérables lors de l'estimation de $\dot{d}(t)$ comme nous pouvons déduire des Tableaux 4.16a et 4.28b.

série	$k = 1$	$k = 2$
1	1.2	0.9
2	0.73	0.65
3	0.9	0.5
4	0.91	0.5
5	0.9	0.62
6	0.8	0.6
7	0.47	0.6
8	1.1	0.8

(a) Faux négatifs.

série	$k = 1$	$k = 2$
1	14.5	17.1
2	14.7	20.1
3	7	22.7
4	15.1	20.4
5	20.7	42.7
6	5.7	10.9
7	7.3	61
8	1	10

(b) Faux positifs.

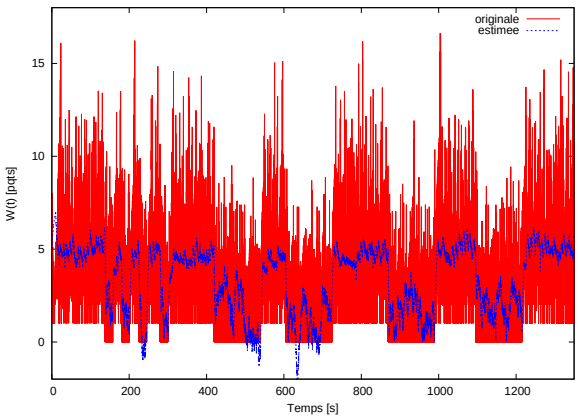
TAB. 4.29: Les faux négatifs et positifs induits par les observateurs DD pour le rejeu n^o3 (exprimés en secondes).

Durant la détection des attaques, les faux négatifs et positifs pour chacune des séries sont déterminés dans le Tableau 4.29 en comparant avant le début les attaques où nous observons des oscillations autour de 1.5 paquets/s. Nous remarquons que les faux négatifs avec $k = 2$ sont très réduits, ils n'atteignent pas 1s. Inversement, $k = 1$ montre une efficacité à réduire les faux négatifs.

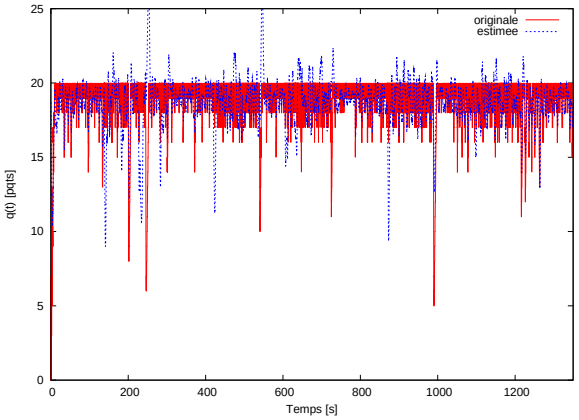
Nous pouvons remarquer aussi des faux positifs qui apparaissent entre les séries 4 et 5 et entre les séries 6 et 7 pour les deux observateurs DD, et les faux positifs qui persistent vers 43s après la série 5.

4.7.5.2 Avec l'observateur glissant d'ordre 2

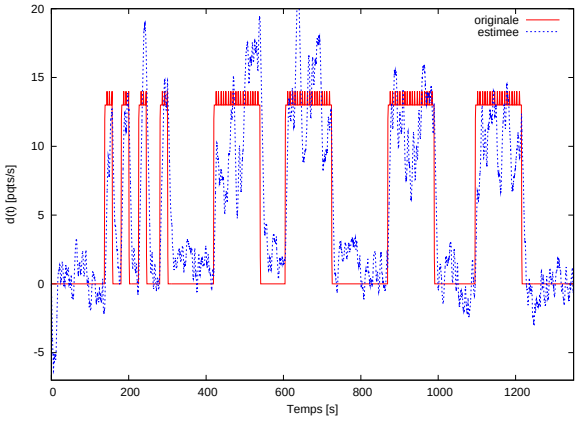
Après l'application des caractéristiques de la topologie du rejeu n^o3 au Théorème 3.3, nous obtenons $L = [4.3843, 6.8196]^T$. L'observateur glissant d'ordre 2 mène aux estimations de $W(t)$, $q(t)$ et $d(t)$ dans la Figure 4.20.



(a) La fenêtre de congestion $W(t)$.



(b) La longueur de la file d'attente $q(t)$.



(c) L'attaque $d(t)$.

FIG. 4.20: Estimations du rejeu n^o3 avec l'observateur glissant.

Comme il est montré dans le Tableau 4.30, sous les mêmes conditions d'expérimentation qu'avec les observateurs DD, la convergence de l'observation par les modes glissants est plus lente avec des erreurs de reconstruction des attaques plus grandes que celle des observateurs DD. L'observateur reconstruit, malgré les perturbations, des attaques de débit entre 10 et 14 paquets/s.

Les faux négatifs et positifs durant la détection de chacune des séries d'attaques sont déterminés dans le Tableau 4.31 avec un seuil égal à 0.3 paquets/s. L'observateur glissant est plus efficace à réduire les faux négatifs que l'observateur DD avec $k = 2$. Par contre, $k = 1$ reste le meilleur dans cette expérimentation pour réduire les faux positifs. Nous notons que comme pour les observateurs DD, les faux positifs persistent 47s après la série 5 et ils apparaissent de nouveau entre les séries 4 et 5 et entre les séries 6 et 7.

série	Temps de convergence [s]	Erreur moyenne [paquets/s]	Écart-type [paquets/s]
1	9.3	-4.81	1.71
2	7.8	-2.74	0.86
3	14.3	3.4	0.44
4	10.8	-1.22	0.72
5	7.16	-3.3	4.28
6	8.1	-1.33	3.44
7	8.2	-3.28	2.47
8	6	-3.68	2.19

TAB. 4.30: Les caractéristiques de l'observation des attaques du rejeu n^o3 par l'observateur glissant.

série	Faux négatifs
1	1.8
2	0.8
3	0.1
4	0.4
5	0.85
6	0.8
7	0.6
8	0.4

(a)

série	Faux positifs
1	18.5
2	21.5
3	24.5
4	66.3
5	47.3
6	11.2
7	18.5
8	12.3

(b)

TAB. 4.31: Les faux négatifs et positifs induits par l'observateur glissant pour le rejeu n^o3 (exprimés en secondes).

4.7.6 $4^{ème}$ rejeu

Pour le trafic TCP, 9 flux sont émis des sources ayant des paquets de taille moyenne de 1500 octets. Le taux d'envoi moyen trouvé est égal à 0.188Mbps avec une latence de 31.71ms. L'attaque envoyée pour ce rejeu est à débit triangulaire décroissant partant de 100Kbps à 0Kbps (chaque paquet de 950 octets). Les courtes attaques sont de 20 secondes chacune avec une durée de 10 à 20 secondes entre deux attaques consécutives, les longues attaques sont de 2 minutes chacune avec une durée variable de 1 à 2 minutes entre elles.

Les caractéristiques du routeur ainsi que le point d'équilibre sont présentés dans le Tableau 4.32.

Point d'équilibre	
W_0	4.0618 paquets
q_0	18.4536 paquets
p_0	0.1212
R_0	2.9245s

Configuration du routeur	
C	0.2Mbps
q_{max}	20 paquets

TAB. 4.32: Configurations du rejeu n^o4.

4.7.6.1 Avec l'observateur DD

Pour cette expérimentation, les gains d'observation résultant du Théorème 2.5 sont les suivants :

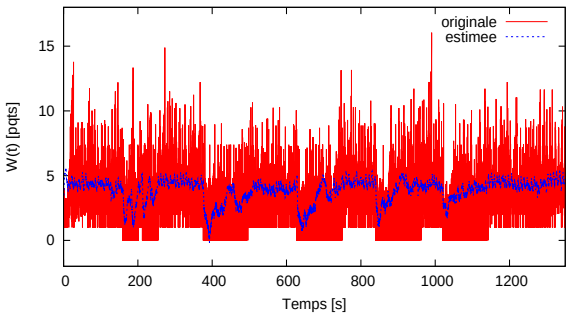
$$\text{pour } k = 1, \quad L = [0.1793, 2.6279, 1.9819, 0.4158]^T; \tag{4.12}$$

$$\text{pour } k = 2, \quad L = [0.0434, 3.3562, 4.5184, 2.1657, 0.4296]^T; \tag{4.13}$$

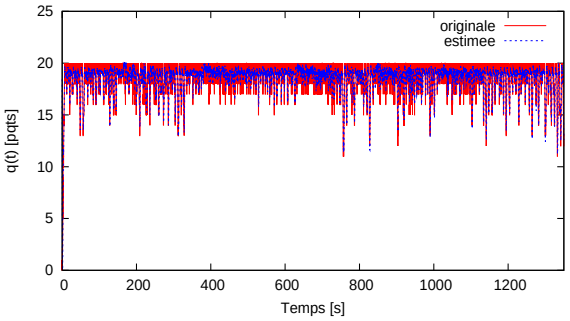
$$\text{pour } k = 3, \quad L = [-0.0451, 4.3824, 8.3031, 6.5641, 2.8199, 0.5309]^T. \tag{4.14}$$

Les graphes obtenus sont représentés dans la Figure 4.21. Nous pouvons remarquer de fortes perturbations durant les estimations de la fenêtre de congestion qui provoquent des changements brusques durant la reconstruction du profil des attaques. Nous sommes réduits à analyser les résultats pour l'observateur DD avec $k = 1$ puisque les observations avec $k = 2$ impliquent de fortes oscillations rendant l'analyse des graphes inepte. Les Tableaux 4.33a et 4.33b résument les caractéristiques de reconstruction des attaques et leurs dérivées premières. Les faux négatifs et positifs sont déterminés pour chacune des séries d'attaques en prenant un seuil égal à -0.9 paquets/s. Dans les Tableaux 4.34, les faux négatifs et positifs présentés sont relatifs aux faux qui apparaissent au début et à la fin de chaque attaque réelle. Les caractéristiques de reconstruction des attaques et leurs dérivées premières dépendent fortement des perturbations qui apparaissent durant l'estimation de la fenêtre de congestion, ainsi que d'autres faux négatifs et positifs apparaissant durant ces attaques.

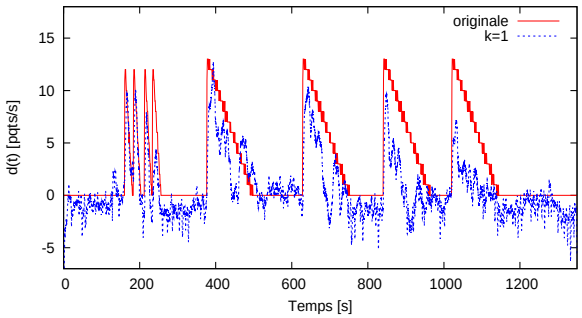
Durant la série d'attaque 4 qui est courte (de 22s), une perturbation est parvenue à la fin de la série dans l'intervalle de temps [252, 255.6]s, provoquant une chute de la reconstruction de l'anomalie en-dessous au seuil. Des faux négatifs sont ainsi générés pendant 3s jusqu'à la disparition de la série, et nous n'aurons pas de faux positifs après sa disparition (Tableaux 4.34b). Ensuite durant la longue série 5, des perturbations dans [425, 460]s diminuent brusquement la reconstruction de l'anomalie sans avoir un impact sur des faux négatifs ou positifs. Pour la série 6, l'attaque est diminuée fortement dans [694.6, 706]s provoquant des faux négatifs, ainsi qu'avant la fin de la série, nous aurons des faux négatifs pendant 3.9s. De plus, des faux négatifs sont induits durant l'observation de la série 7 dans [902, 925]s et [941.4, 953.8]s. Enfin, l'observation de la série 8 est totalement perturbée et des faux négatifs



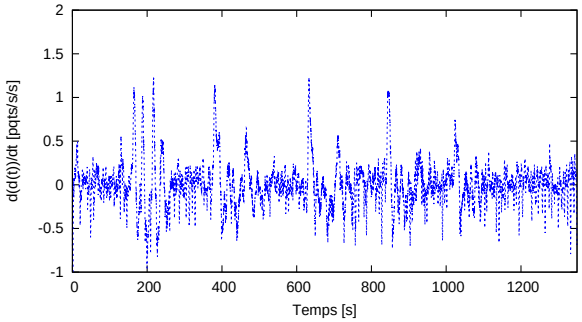
(a) La fenêtre de congestion $W(t)$.



(b) La longueur de la file d'attente $q(t)$.

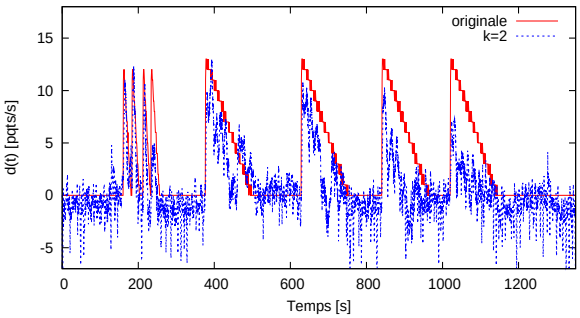


(c.1) $d(t)$

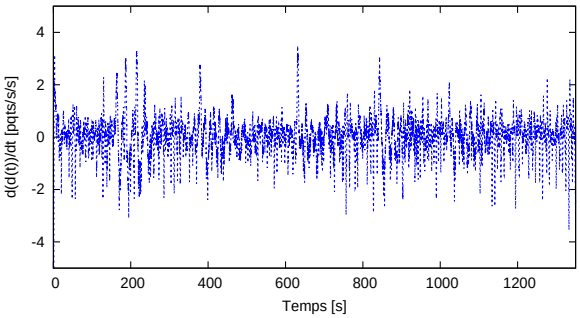


(c.2) $\dot{d}(t)$

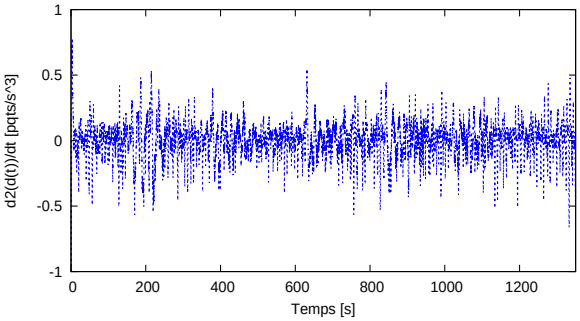
(c) L'attaque et sa dérivée pour $k = 1$.



(d.1) $d(t)$



(d.2) $\dot{d}(t)$



(d.3) $\ddot{d}(t)$

(d) L'attaque et ses dérivées pour $k = 2$.

FIG. 4.21: Estimations DD du rejeu n^o4.

série	Temps de convergence [s]	Erreur moyenne [paquets/s]	Écart-type [paquets/s]
1	4.7	0.81	0.98
2	4.9	0.15	1.29
3	4.6	-1.69	1.02
4	9.15	-1.45	0.38
5	9.1	-1.36	2.55
6	7.1	-3.09	1.68
7	6.56	-5.22	1.62
8	7.5	-4.7	2.04

(a) L'attaque $d(t)$.

série	Erreur moyenne [paquets/s]	Écart-type [paquets/s]
1	0.67	0.6
2	0.48	0.6
3	0.72	0.63
4	0.54	0.3
5	0.07	0.3
6	0.05	0.32
7	0.04	0.35
8	0.06	0.2

(b) La dérivée première $\dot{d}(t)$.

TAB. 4.33: Les caractéristiques de l'observation des attaques et leurs dérivées premières du rejeu n^o4 par l'observateur DD.

durant les dernières 2.5s de la série.
Les erreurs moyennes observées sont principalement affectées par ces perturbations comme cela est montré dans le Tableau 4.33a. 12 paquets/s sont observées dans la série 6 et 10 paquets/s dans les séries 7 et 8, tandis que l'observation arrive à 15 paquets/s pour les autres. Les dérivées premières devant être égales à -0.6 paquets/s^2, ne sont pas bien observées par valeurs moyennes pour les courtes attaques, alors qu'elles se rapprochent de la valeur réelle pour les longues attaques (c.f. Tableau 4.33b). Les grands écart-types rendent l'observation des dérivées trop difficile.

Nous pouvons conclure aussi des graphes de la Figure 4.21 et des Tableaux 4.34, que les faux négatifs ne sont pas générés avant la série 2 à cause des faux positifs qui persistent entre les séries 1 et 2 où la durée est très petite de 3s. Des faux positifs apparaissent aussi entre les séries 5 et 6 évitant les faux négatifs après le début de la série 6. Des faux positifs sont déduits entre les séries 7 et 8.

série	Faux négatifs
1	0.1
3	1.2
4	1.1
5	0.2
7	0.3
8	0.1

(a)

série	Faux positifs
1	3
2	5.5
3	1.6
5	2.5
7	24

(b)

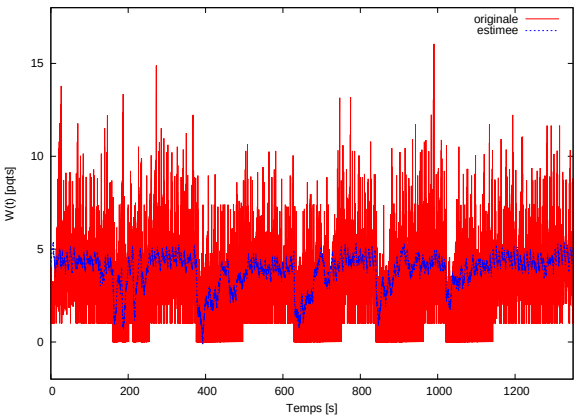
TAB. 4.34: Les faux négatifs et positifs induits par les observateurs DD pour le rejeu n^o4 (exprimés en secondes).

4.7.6.2 Avec l'observateur glissant d'ordre 2

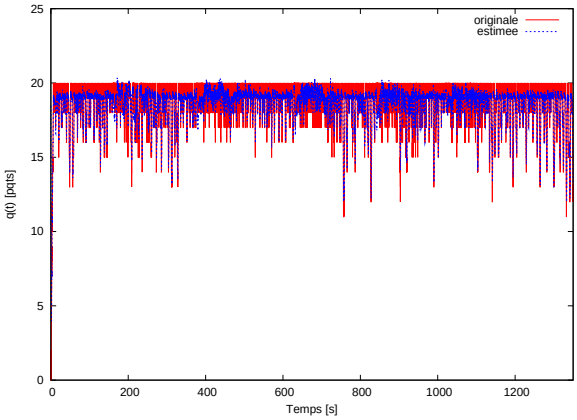
La conception de l'observateur glissant d'ordre 2 à partir du Théorème 3.3 mène au gain $L = [4.3968, 6.8636]^T$. Les graphes d'estimation sont présentés dans la Figure 4.22.

Comme pour les observateurs DD, les graphes de la Figure 4.22 montrent que l'estimation de la fenêtre de congestion et la reconstruction du profil des attaques sont affectées par de fortes perturbations. Dans le Tableau 4.35, les caractéristiques de reconstruction des attaques sont présentées. Les faux négatifs et positifs pour chacune des séries sont déterminés en prenant un seuil égal à -0.8 paquets/s. Dans les Tableaux 4.36, les faux négatifs et positifs présentés sont relatifs aux faux qui apparaissent au début et à la fin de chaque attaque réelle. L'observateur glissant montre des performances qui sont différentes de l'observateur DD face aux perturbations.

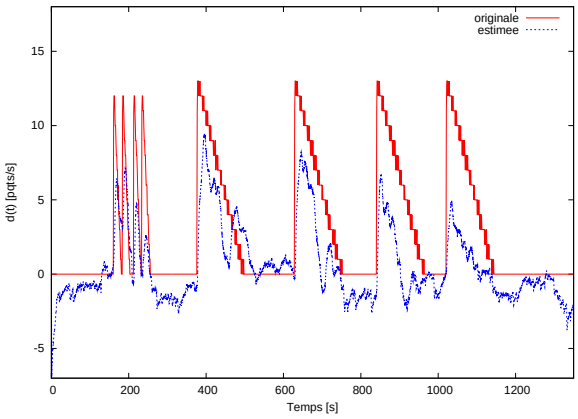
L'observation de la série d'attaque 4 n'est pas affectée par la perturbation contrairement à l'observateur DD (c.f. Tableaux 4.36). Durant la série 5, la reconstruction de l'anomalie diminue brusquement sans avoir un impact sur des faux négatifs et positifs comme dans le cas de l'observateur DD. Durant la série 6, des faux négatifs apparaissent dans l'intervalle [698.3, 707.3]s, ainsi que durant 0.5s juste avant la disparition de la série. Les faux positifs n'existent pas après la fin de la série. De plus, des faux négatifs sont induits durant l'observation de la série 7 dans [908, 925]s et [942, 952]s. L'observation de la série 8 est totalement perturbée et des faux négatifs de durée 2.5s sont induits jusqu'à la fin. Nous pouvons remarquer que les faux négatifs apparus durant l'apparition des attaques sont plus réduits avec l'observateur glissant parce que l'estimation de la fenêtre de congestion est moins affectée



(a) La fenêtre de congestion $W(t)$.



(b) La longueur de la file d'attente $q(t)$.



(c) L'attaque $d(t)$.

FIG. 4.22: Estimations du rejeu n^o4 avec l'observateur glissant.

par ces perturbations.
La convergence est plus lente avec l'observateur glissant que l'observateur DD pour toutes les séries d'attaques comme nous pouvons voir du Tableau 4.35. Par contre, les erreurs moyennes observées pour les séries 3 jusqu'à 8 sont plus petites que celles trouvées avec l'observateur DD avec des écarts aussi plus faibles. 12 paquets/s sont observées dans la série 6 et 10 paquets/s dans la série 7 et 11 paquets/s dans la série 8, tandis que l'observation arrive à 15 paquets/s pour les autres.
En comparant les faux négatifs et positifs entre l'observateur glissant dans les Tableaux 4.36 et l'observateur DD dans les Tableaux 4.34, nous pouvons remarquer que l'observateur glissant réduit les faux négatifs au début des attaques et augmente les faux positifs par rapport à l'observateur DD. De plus, comme nous le montrons dans la Figure 4.22c et les Tableaux 4.36, les faux négatifs ne sont pas générés avant les séries 2 et 4 à cause des faux positifs qui persistent entre les séries 1 et 2, et entre 3 et 4. Des faux positifs apparaissent aussi entre les séries 5 et 6 évitant les faux négatifs après le début de la série 6, et entre 7 et 8.

Nous pouvons conclure aussi sur l'avantage de l'observateur glissant à diminuer les taux

série	Temps de convergence [s]	Erreur moyenne [paquets/s]	Écart-type [paquets/s]
1	8.3	1.3	1.05
2	6.4	0.9	0.84
3	5.6	-1.57	1.14
4	9.2	-1.17	1.45
5	16.3	-1.07	1.99
6	13.9	-2.81	1.09
7	8.7	-5.11	0.69
8	9.15	-4.48	1.8

TAB. 4.35: Les caractéristiques de l'observation des attaques du rejeu n^o4 par l'observateur glissant.

série	Faux négatifs
1	0.07
3	0.9
5	1.1
7	0.1
8	0.01

(a)

série	Faux positifs
1	3
2	6.3
3	2
4	4.1
5	30.6
7	26.2

(b)

TAB. 4.36: Les faux négatifs et positifs induits par l'observateur glissant pour le rejeu n^o4 (exprimés en secondes).

des faux négatifs durant la reconstruction de l'attaque en présence des imperfections dans l'observation, ce qui est un point important pour la sécurité du réseau.

4.7.7 Analyse des résultats des expérimentations

Les analyses des résultats des traces de trafic montrent l'efficacité des observateurs basés sur l'approche DD et sur les modes glissants de second ordre évalués pour chacun des trafics TCP rejoués. Les observateurs réagissent rapidement aux évolutions de $W(t)$ et des anomalies $d(t)$ de type CBR et TBR avec des suivis parfois perturbés des formes originales. En effet, nous remarquons que dans des fenêtres de temps distinctes, l'estimation de la fe-

nêtre de congestion moyenne $W(t)$ ne suit plus l'évolution moyenne. Si les perturbations arrivent pendant la présence d'une attaque, la reconstruction de cette dernière est fortement perturbée, menant dans certains cas à des faux négatifs. En outre, pendant l'absence d'une attaque, des faux positifs peuvent être générés. Ces perturbations reflètent l'aspect réaliste des trafics mis en rejeu. En comparant les performances des observateurs nous constatons les imperfections qui apparaissent aux mêmes moments. Ce phénomène peut être expliqué par les deux aspects expérimentaux suivants :

- A cause de l'utilisation du mécanisme de TBF, pour réguler le débit d'entrée au routeur, qui est à base du Drop Tail, les paquets sont éjectés dès que la file d'attente atteint sa limite maximale. De ce fait, en présence des attaques, surtout de longues durées, les sources sont fréquemment forcées à réduire leur fenêtre d'émission en atteignant zéro pour un certain laps de temps. Ce qui contredit l'hypothèse fondamentale du modèle non linéaire de TCP adopté qui est la persistance d'émission des paquets dans la phase d'évitement de congestion.
- Les tailles des paquets TCP réelles, injectées au générateur de trafic sous NS-2, comprennent des valeurs qui sont loin de la taille moyenne prise pour l'équilibre autour duquel nos observateurs sont construits.

Par conséquent, ces deux faits pratiques perturbent l'évaluation de $W(t)$, la reconstruction de $d(t)$ et ses dérivées successives. Ces erreurs d'observation mettent en cause les limitations du modèle fluide de TCP adopté qui reste valide autour du point d'équilibre pris en compte. L'avantage principal de nos observateurs réside dans la détection très rapide de l'attaque par les deux théories traitées. De plus, le flux attaquant est reconstruit en grande partie en dépit des conditions pessimistes. Les résultats que nous avons exposés montrent la pertinence des observateurs construits pour des conditions réelles, en se basant sur les hypothèses du modèle théorique.

4.8 Conclusion

Au cours de ce chapitre, nous avons mis en application les résultats théoriques des chapitres précédents. Deux niveaux d'expérimentation ont été envisagés pour chacun des observateurs étudiés : observateurs de Luenberger IOD minimal/DD et observateurs par modes glissants d'ordre 1 et 2. Tout d'abord, nous avons principalement effectué des tests à l'aide du simulateur NS-2. Ces simulations ont nécessité l'implémentation des techniques d'observation dans le noyau du logiciel, associés aux différents AQM : RED, PI et Gain-K. Nous avons ainsi pu montrer, d'une part, que les observateurs de Luenberger conçus par l'approche DD avec des anomalies polynômiales et l'observateur glissant d'ordre 2 permettent de reconstruire les profils rectangulaire et triangulaire. D'autre part, pour la détection, les faux négatifs sont mieux gérés avec l'observateur glissant d'ordre 2. Les modes glissants d'ordre 2 permettent la reconstruction d'un profil d'anomalies quelconque sans avoir besoin ni du réglage de filtrage, ni de l'ordre du polynôme d'anomalies. A contrario, les observateurs de Luenberger sont capables d'apporter plus d'informations sur l'anomalie en observant simultanément ses dynamiques supérieures. Pour choisir le type d'observation à adopter pour le modèle TCP/IP, nous sommes face à un compromis entre le maximum d'information consentie et les durées d'émission de faux négatifs et positifs tolérables.

Le second niveau d'expérimentation concerne les tests de rejeu de traces, dans lesquels des caractéristiques de flux réels sont simulées sous le simulateur de réseaux NS-2 ayant les fonctionnalités adéquates. Plus réaliste, cette méthode requiert un traitement spécial pour la trace collectée du réseau au niveau d'un routeur. Les observateurs de Luenberger DD et les observateurs par modes glissants d'ordre 2 détectent la présence et l'apparition des attaques malgré les contraintes reliées à l'aspect réel des expérimentations. Les débits des attaques sont aussi observés en grande partie. Les résultats prometteurs pour les observateurs de Luenberger et les observateurs glissants nous encouragent à poursuivre notre étude pour réaliser des expérimentations avec d'autres mécanismes d'AQM et d'autres types d'attaques que les UDP floodings.
Ayant élaboré des méthodologies de détection et reconstruction des anomalies, l'étape suivante consisterait à proposer une architecture autonome de supervision du modèle TCP/IP dans une perspective de garantie d'une meilleure QdS ainsi qu'une durée de vie maximale du modèle TCP/IP en présence d'une anomalie.

Conclusions et Perspectives

Les anomalies sont de plus en plus diverses dans les réseaux de communication de façon que leur détection devient une tâche critique dans l'administration des réseaux. La problématique de ce mémoire est la détection et la reconstruction d'une classe d'anomalies survenant au niveau d'un routeur dans une architecture de réseau TCP/IP. Notre travail de thèse s'inscrit dans deux thématiques, l'Automatique et les réseaux de communication. Les techniques d'observation dans le domaine d'Automatique sont apportées aux représentations mathématiques du comportement des modèles TCP/IP afin d'améliorer la Qualité de Service en terme de sécurité. Pour un modèle adopté, nous élaborons des observateurs capables de détecter et reconstruire les profils des anomalies au niveau d'un routeur. Notre contribution est novatrice au contraire des techniques de l'Informatique qui sont basées soit sur la détection des signatures des anomalies connues, soit sur le résultat d'un apprentissage effectué sur un échantillon de trafic.

Le premier chapitre a été consacré au cadre de travail de la thèse. Le fonctionnement du protocole TCP a été brièvement présenté selon le modèle de couches gérant la communication entre deux machines dans un réseau. Nous nous sommes ensuite intéressés au comportement du routeur en phase de congestion. Un modèle proposé par [Misra 2000] est représenté par un modèle non linéaire à retards. Ce modèle mathématique nous sera utile pour développer des techniques d'observation issues de la théorie de contrôle. Effectivement, dans le domaine de l'Informatique, la plupart des outils de détection des anomalies provenant des utilisations légitimes ou illégitimes se limite sur des types spécifiques d'anomalies. Notre travail s'est focalisé sur l'observation des flux d'anomalies au niveau du routeur. Différentes techniques d'observation sont exposées en les classant selon la connaissance du profil des défauts, en observateurs à entrées connues et à entrées inconnues.

Dans le Chapitre 2, les formes polynômiales couvrant une large gamme de profils d'anomalies sont prises en compte dans le modèle TCP/IP. Nous avons déduit que l'observateur de Luenberger conçu pour le modèle comprenant l'anomalie et ses dérivées successives n'est pas faisable par la deuxième méthode de Lyapunov-Krasovskii indépendant du retard (IOD). Ce premier résultat nous a amené à proposer un observateur d'ordre réduit pour la partie inconnue du modèle TCP qui est constituée de la fenêtre de congestion des sources et des dynamiques de l'anomalie. Sous Simulink, les anomalies sont détectées instantanément et leurs profils sont bien suivis indépendamment du retard et du degré du polynôme considéré pour l'anomalie. Par l'approche dépendant du retard (DD), un observateur appliqué au système complet montre une bonne sensibilité aux présences et disparitions des anomalies. Par conséquent, les faux positifs et négatifs diminuent en augmentant le degré de l'anomalie. Une approche robuste est ensuite envisagée pour permettre de considérer un retard incertain entre deux bornes minimale et maximale. La possibilité de reconstruire les dérivées supérieures de l'anomalie reste le principal avantage de la modélisation polynômiale des anomalies. Par contre, cette approche est limitée par l'anomalie considérée sous forme polynômiale et le choix du degré du polynôme.

Une approche alternative traitant l'observation de telles perturbations d'une façon géné-

rique a fait l'objet du Chapitre 3. Les observateurs par modes glissants sont construits pour estimer l'état inconnu en dépit de la présence des anomalies complètement inconnues. Ces dernières sont ensuite reconstruites à partir de la commande équivalente par des techniques de filtrage. Sous Simulink, le filtrage passe-bas d'ordre 3 introduit à l'observation du premier ordre réduit le phénomène de réticence mais ralentit la détection de l'anomalie. Par rapport au premier ordre, l'observateur glissant de second ordre révèle de meilleures performances en termes de réponse plus rapide à la présence/absence de l'anomalie et un meilleur suivi de la forme du flux anormal. Le glissement d'ordre 2 apporte des améliorations sur la détection et la reconstruction de l'anomalie par rapport à l'ordre 1 et aux observateurs de Luenberger.

Les applications des résultats théoriques des chapitres précédents sont mises en place sous le simulateur des réseaux NS-2 dans le chapitre 4. Deux niveaux d'expérimentation ont été envisagés pour chacun des observateurs étudiés : les observateurs de Luenberger IOD minimal/DD et les observateurs par modes glissants d'ordre 1 et 2. Tout d'abord, nous avons simulé sous NS-2 des topologies équivalentes à Simulink. Les techniques d'observation sont associées aux différents mécanismes de contrôle de cogestion ou AQM : RED, PI et Gain-K. Nous avons ainsi pu montrer que selon l'AQM associé au protocole TCP, les observateurs de Luenberger conçus par l'approche DD avec des anomalies polynômiales ou l'observateur glissant d'ordre 2 permet de mieux reconstruire les profils rectangulaires et triangulaires. Pour la détection, les faux négatifs et positifs sont plus réduits avec l'observateur glissant d'ordre 2. Par contre, les observateurs de Luenberger sont capables d'observer simultanément l'anomalie et ses dérivées supérieures. Nous sommes face à un compromis entre le maximum d'information consentie et les durées d'émission de faux négatifs et positifs tolérables.
La vérification des performances des observateurs est validée à l'aide des logiciels de modélisation Matlab/Simulink et NS-2 où les trafics sont synthétiques. Nous sommes concernés par un niveau d'expérimentation plus réaliste où des caractéristiques des flux TCP réels sont présentes. Nous adoptons la méthode de rejeu de traces de trafic capturées au niveau d'un routeur dans la topologie de réseau. Les caractéristiques des flux réels sont extraites des traces après un traitement spécial hors-ligne, puis simulées dans NS-2 qui contient les fonctionnalités adéquates au rejeu. Les observateurs de Luenberger DD et les observateurs par modes glissants d'ordre 2 détectent la présence et l'apparition des attaques, ainsi qu'ils observent une grande partie du débit d'attaque envoyé malgré les contraintes reliées à l'aspect réel des expérimentations.

Les travaux de recherche que nous avons exposés dans ce mémoire, surtout concernant la partie des expérimentations des rejeux de traces de trafic, aboutissent à des résultats prometteurs qui nous encouragent à poursuivre à court terme notre étude dans la thématique d'observation des anomalies. A long terme, nous proposons d'ouvrir la voie à de techniques de supervision des réseaux apportées par l'Automatique. Plusieurs thématiques de recherche peuvent être impliquées dans le domaine de l'Informatique comme les protocoles de routage des paquets, et dans l'Automatique comme les systèmes hybrides et les modèles non linéaires.

Perspectives à court terme

Concernant l'observation des anomalies au niveau d'un routeur dans une topologie de modèles TCP/IP, nous proposons plusieurs points à approfondir :

- Comme nous avons mentionné dans le Chapitre 2, pour déterminer les gains d'observation relatifs à l'approche DD pour l'observateur minimal, il s'agit de résoudre une équation qui comprend des termes dépendant des gains multipliés exponentiellement par les pôles. La résolution analytique de cette équation est à étudier.
- Pour les expérimentations sur les rejeux de traces réelles, d'autres topologies peuvent être envisagées afin de mieux évaluer l'efficacité et les limites de nos observateurs. D'autres mécanismes d'AQM peuvent être implantés sur le routeur réel. Cependant, la difficulté réside dans l'association d'un mécanisme de gestion de la file d'attente comme RED avec une limitation de la capacité du routeur. Ce problème reste ouvert et à résoudre.
- Les attaques de type UDP flooding générés par le logiciel TFN2K en forme plates et en rampes sont prises en compte pour les rejeux. Les tailles de ces attaques peuvent être manipulées. Nous pouvons proposer d'autres types d'attaques comme les ICMP flooding ou les SYN flooding. Pour ces attaques, les paquets sont de petites tailles fixes puisqu'ils ne comprennent que des entêtes. Ce qui n'est pas pris en compte dans le modèle adopté pour TCP. Pour des attaques spécifiques, nous pouvons envisager de modéliser les comportements des sources attaquantes et les réponses des destinataires. Ce modèle d'attaque pris comme signature peut être corrélé au comportement normal du protocole TCP. Des observateurs, conçu chacun pour un type d'attaques, serviront ensemble à identifier et classifier les attaques.
- Au cours de cette thèse, le temps d'aller-retour moyen des paquets TCP, par conséquent le retard, est supposé constant et incertain. Nous souhaitons construire un observateur afin d'identifier le retard variable $R(t)$. L'observation peut être ensuite améliorée en concevant des observateurs *Linéaires à Paramètres Variables* ou LPV en fonction du retard ou robustes vis-à-vis des variations du retard.
- La synthèse d'un observateur pour le système non linéaire apparaît être une bonne solution pour se libérer du point de fonctionnement nécessitant l'implémentation d'un mécanisme d'AQM.
- En ce qui concerne la modélisation du protocole TCP, nous pouvons mettre en jeu des flux qui sont hétérogènes [Low 2002], [Ariba 2009]. Des observateurs construits pour de tels modèles peuvent réduire les contraintes durant les expérimentations sur des trafics réalistes.

Perspectives à long terme

Ayant élaboré des méthodologies d'observation des anomalies dans le réseau, nous pouvons orienter notre recherche vers une architecture autonome de supervision du modèle TCP/IP. Les objectifs sont la garantie d'une meilleure Qualité de Service ainsi qu'une durée de vie maximale du modèle en présence d'une anomalie. Afin de limiter l'impact des anomalies sur le routeur et la partie du réseau concernés, la stratégie proposée consiste à dérouter une partie des flux TCP vers un(des) autre(s) routeur(s) non touché(s) par l'anomalie. Ce changement de routage de certains paquets TCP peut aider le routeur à assurer un bon

fonctionnement vis-à-vis des flux restants.
Dans le modèle dynamique adopté dans notre travail, le changement de routage affectera le nombre de flux TCP (N) dépendamment de l'amplitude de l'anomalie estimée. Nous pouvons définir ainsi des sous-systèmes commutant entre eux selon l'état de l'anomalie. Si l'anomalie consomme un pourcentage de la capacité du routeur ($\frac{d}{C}\%$), notre proposition est de diminuer le nombre de flux du même pourcentage. Nous définissons 5 seuils correspondant à l'anomalie par rapport à la capacité du routeur. Une configuration est définie dans le Tableau 4.37.
A noter que pour le dernier état où le routeur est face à une attaque égale à sa capacité,

Amplitude d'anomalie		Flux restants	Sous-système
Nulle ou faible	$0 \leq d(t) < 0.25C$	$N_1 = N_{max}$	S_1
Moyenne	$0.25C \leq d(t) < 0.5C$	$N_2 = 0.75N_{max}$	S_2
Forte	$0.5C \leq d(t) < 0.75C$	$N_3 = 0.5N_{max}$	S_3
Très forte	$0.75C \leq d(t) < 0.95C$	$N_4 = 0.25N_{max}$	S_4
Attaque	$0.95C \leq d(t)$	$N_5 = 0$	S_5

TAB. 4.37: Configurations des sous-systèmes.

$N_5 = 0$ n'est pas envisageable puisque le modèle mathématique ne sera plus applicable. Nous passons donc à définir mathématiquement les 4 sous-systèmes définis à partir de la topologie nominale (2.1) :

$$\begin{cases} \dot{W}(t) &= \sum_{i=1}^{4} \gamma_i \frac{1}{R(t)} - \frac{W(t)W(t-R(t))}{2R(t-R(t))} p(t-R(t)), \\ \dot{q}(t) &= \sum_{i=1}^{4} \gamma_i \frac{W(t)}{R(t)} N_i - C + d(t). \end{cases} \tag{4.15}$$

La configuration du routeur n'est pas changée, la capacité C reste donc intacte. Comme les flux sont homogènes, le temps d'aller retour ne changera pas d'un sous-système à un autre. D'où $R_i(t) = R(t)$.
La présence des aspects continus et discrets définit un modèle global hybride. Pour assurer la stabilité globale, il est nécessaire de passer par la notion de région d'attraction. Ayant des points d'équilibre et une région d'attraction définis pour chacun des sous-systèmes, la convergence vers un second sous-système après une transition n'est assurée que si le point d'équilibre du dernier appartient à la région d'attraction du premier. Comme les transitions entre les systèmes sont aléatoires, pour garantir la stabilité globale du modèle hybride, il sera nécessaire de définir une région d'attraction commune dans laquelle les 4 points d'équilibre sont localisés.

La région d'attraction exacte est difficile à trouver. Nous proposons de l'estimer par une fonctionnelle de Lyapunov-Krasovskii. Avant de considérer le modèle hybride, nous visons linéariser le système nominal de TCP en conservant ses propriétés non linéaires. En d'autres termes, le modèle non linéaire peut être mis sous la forme d'une somme des termes linéaires et des termes non linéaires comme suit :

$$\dot{X}(t) = AX(t) + A_dX(t-h) + B_du(t-h) + \bar{A}X(t) + \bar{A}_dX(t-h) + \bar{B}_du(t-h) \tag{4.16}$$

où les non linéarités résident dans les matrices $\bar{A}, \bar{A}_d$ et $\bar{B}_d$ qui sont bornées.
L'application de la fonctionnelle de Lyapunov-Krasovskii

$$V(t) = x^T(t)Px(t) + \int_{-h}^{0} x^T(s)Qx(s)ds,$$

résulte des conditions infaisables en prenant en compte l'approche polytopique et l'approche bornée en norme pour les termes non linéaires bornés. La définition de la région d'attraction garantissant la stabilité du modèle hybride reste un problème ouvert à résoudre vu la complexité du modèle TCP non linéaire.

Annexe A

Linéarisation de TCP

Le système (2.1) modélisant le comportement de TCP étant non linéaire, il est nécessaire de le linéariser autour d'un point d'équilibre (W_0, q_0, p_0). Pour des raisons de simplicité, nous supposons dans un premier temps que le facteur de charge N et la capacité du lien C sont constants. Le point d'equilibre est défini par le couple d'équations :

$$\begin{cases} \dot{W}(t) = 0 \;\Rightarrow\; W_0^2 p_0 = 2, \\ \dot{q}(t) = 0 \;\Rightarrow\; W_0 = \frac{R_0 C}{N},\; R_0 = \frac{q_0}{C} + T_p. \end{cases} \tag{A.1}$$

Pour simplifier les détails de calcul, considérons les fonctions f et g représentant $W(t)$ et $q(t)$ telles que

$$\begin{cases} f &= \frac{1}{\frac{q}{C}+T_p} - \frac{WW_R}{2(\frac{q_R}{C}+T_p)} p_R, \\ g &= \frac{W}{\frac{q}{C}+T_p} N - C, \end{cases} \tag{A.2}$$

où $W_R = W(t - R(t))$, $q_R = q(t - R(t))$ et $p_R = p(t - R(t))$ sont les variables retardées d'un retard variable $R(t) = \frac{q}{C} + T_p$.

Les dérivées partielles de f et g par rapport à (W, W_R, q, q_R, p_R) sont exprimées en fonction du point d'équilibre comme le suivant :

$$\begin{aligned} \frac{\partial f}{\partial W} &= -\frac{W_R}{2(\frac{q_R}{C}+T_p)} p_R = -\frac{W_0}{2(\frac{q_0}{C}+T_p)} p_0 = -\frac{W_0 p_0}{2R_0} = -\frac{N}{R_0^2 C}, \\ \frac{\partial f}{\partial W_R} &= \frac{\partial f}{\partial W}, \\ \frac{\partial f}{\partial p_R} &= -\frac{WW_R}{2(\frac{q_R}{C}+T_p)} p_R = -\frac{W_0^2}{2R_0} = -\frac{C^2 R_0}{2N^2}, \\ \frac{\partial f}{\partial q} &= -\frac{1}{C(\frac{q}{C}+T_p)^2} = -\frac{1}{C(\frac{q_0}{C}+T_p)^2} = -\frac{1}{CR_0^2}, \\ \frac{\partial f}{\partial q_R} &= \frac{WW_R}{2C(\frac{q}{C}+T_p)^2} p_R = \frac{W_0^2 p_0}{2C(\frac{q_0}{C}+T_p)^2} = \frac{1}{CR_0^2}, \\ \frac{\partial g}{\partial W} &= \frac{N}{\frac{q}{C}+T_p} = \frac{N}{\frac{q_0}{C}+T_p} = \frac{N}{R_0}, \\ \frac{\partial g}{\partial q} &= -\frac{NW}{C(\frac{q}{C}+T_p)^2} = -\frac{NW_0}{C(\frac{q_0}{C}+T_p)^2} = -\frac{1}{R_0}. \end{aligned} \tag{A.3}$$

Autour d'un point d'équilibre donné par (A.1), les variations de $\delta W = W - W_0$ et $\delta q = q - q_0$ seront de la forme :

$$\begin{cases} \delta\dot{W}(t) &= -\frac{N}{R_0^2 C}(\delta W(t) + \delta W(t - h(t))) \\ & \quad -\frac{1}{R_0^2 C}(\delta q(t) - \delta q(t - h(t))) - \frac{R_0^2 C^2}{2N^2} \delta p(t - h(t)) \\ \delta\dot{q}(t) &= \frac{N}{R_0}\delta W(t) - \frac{1}{R_0}\delta q(t) + d(t) \end{cases} \tag{A.4}$$

Annexe B

Théorèmes

Dans cette partie sont présentés quelques résultats relatifs aux inégalités matricielles utilisées dans ce mémoire.

B.1 Complément de Schur

Le complément de Schur est une technique utilisée pour reformuler certaines inégalités matricielles non linéaires en terme d'*Inégalités Matricielles Linéaires* (LMI) comme décrit dans le lemme suivant [Boyd 1994].

Lemme B.1. *Soient les matrices $Q(x) \in \mathbb{R}^{m\times m}, R(x) \in \mathbb{R}^{n\times n}$ et $S(x) \in \mathbb{R}^{m\times n}$ dépendant d'une variable $x \in \mathbb{R}^p$. Les inégalités matricielles suivantes sont équivalentes :*

$$\begin{cases} R(x) > 0, \\ Q(x) - S(x)R^{-1}(x)S^T(x) > 0. \end{cases} \tag{B.1}$$

et

$$\begin{bmatrix} Q(x) & S(x) \\ S^T(x) & R(x) \end{bmatrix} > 0. \tag{B.2}$$

La dernière inégalité matricielle est une LMI si les matrices $Q(x), R(x)$ et $S(x)$ dépendent affinement de la variable x.

B.2 Inégalité de Jensen

Cette inégalité est souvent utilisée pour majorer des termes intégraux lors du calcul de la dérivée de certaines fonctionnelles de Lyapunov-Krasovskii.

Lemme B.2. *Pour tout scalaire $\rho > 0$ et une matrice définie positive $M \in \mathbb{R}^{m\times m}$, l'inégalité suivante est vraie :*

$$\rho \int_0^\rho x^T(\eta)Mx(\eta)d\eta \geq \left(\int_0^\rho x^T(\eta)d\eta\right) M \left(\int_0^\rho x(\eta)d\eta\right). \tag{B.3}$$

Annexe C

Exemple d'une trace

Une trace contient le détail du trafic visible depuis une interface réseau. C'est une succession de trames capturées par un outil de capture donné. Nous présentons un extrait d'une trace capturée par le logiciel Wireshark [Orebaugh 2007]. Les trames sont visualisées sous la forme suivante :

```
No.     Time           Source        Destination  Protocol   Info
23804 11:10:18.235252 140.93.xx.xxx   193.49.uu.uuu  TCP    49238 > ssh [ACK] Seq=430641 Ack=19966097 Win=65535 Len=0
23805 11:10:18.236051 140.93.xx.xxx   193.49.uu.uuu  SSH    Encrypted request packet len=80[Packet size limited during capture]
23806 11:10:18.262957 193.55.yyy.yy   193.49.uu.uuu  IP     Bogus IP header length (0, must be at least 20)
23807 11:10:18.275037 193.55.zzz.zz   193.49.uu.uuu  TCP    49466 > commplex-link [PSH, ACK] Seq=1378075 Ack=1 Win=5840 Len=1434
23808 11:10:18.280130 140.93.xx.xxx   193.49.uu.uuu  TCP    49238 > ssh [ACK] Seq=430721 Ack=19969017 Win=65535 Len=0
23809 11:10:18.292111 140.93.xx.xxx   193.49.uu.uuu  SSH    Encrypted request packet len=80[Packet size limited during capture]
23810 11:10:18.294256 140.93.xx.xxx   193.49.uu.uuu  TCP    [TCP Dup ACK 23809#1] 49238 > ssh [ACK] Seq=430801 Ack=19970477 Win=65535 Len=0
```

Dans Wireshark, les détails des trames sont sauvegardés dans un fichier sous le format libpcap. Dans le fichier, chaque trame prend une forme bien particulière : l'entête sous forme d'une structure qui contient tous les champs des protocoles suivie des données contenues dans la trame. Le fichier résultant est la succession des trames organisées suivant ce format. Prenons par exemple la première trame. L'entête est formé d'octets comme suit :

```
00 04 23 c5 a8 5b 00 12 1e 28 74 00 08 00 45 00
00 78 3d 48 40 00 7b 06 09 2c 8c 5d 0a 72 c1 31
61 0b c0 56 00 16 e0 73 f4 1b 04 95 c2 ef 50 18
ff ff 3b 80 00 00 1f c2 21 16 4e 91 a8 24 5a 39
2d 27 06 c3 da 56 b0 d3 58 49 dd 5a a9 94 56 79
38 83 44 1e 64 ea 74 aa a0 db b9 34 f6 e8 ef 3f
```

Sous le format libpcap, Wireshark décode et affecte des valeurs aux champs relatifs à chaque protocole utilisé dans la transmission de la trame.

```
Ethernet II,
    Destination: Intel_xx:xx:xx
    Source: Intel_xx:xx:xx
    Type: IP (0x0800)

Internet Protocol,
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 40
    Flags: 0x04 (Don't Fragment)
    Fragment offset: 0
    Time to live: 123
    Protocol: TCP (0x06)
    Header checksum: 0x097f [correct]
    Source: 140.93.xx.xxx
    Destination: 193.49.uu.uuu

Transmission Control Protocol,
    Source Port: 49238
    Destination Port: ssh (22)
```

```
    Sequence number: 430641
    Ack number: 19966097
    Header length: 20
    Checksum: 0xabb3 [correct]

Data (0 bytes)
```

Annexe D

Scripts et codes dans NS-2

Cet annexe contient différents scripts pour des tests de simulations et de rejeux avec le simulateur de réseaux NS-2, ainsi que des codes qui correspondent aux observateurs (de Luenberger IOD et DD, glissants d'ordre 1 et 2) que nous avons construits dans les chapitres précédents.

- Les scripts des simulations et des rejeux sont des programmes Tcl utilisés pour la description de la topologie du modèle TCP/IP, la configuration des liens, des agents émetteurs et récepteurs et la description du scénario.
- Les modèles discrétisés des observateurs sont introduits aux codes sources C++ des différents AQM testés (RED, PI, retour d'état Gain-K, TBF). Dans cette annexe, nous présentons les extraits ajoutés au code de l'AQM RED : red.cc.
  NS-2 est un simulateur à événements discrets où les événements sont orientés commutation de paquets : arrivée d'un paquet, envoie des ACK,... [Fall 2002]. Puisque notre étude est centralisée sur un routeur dans le modèle TCP/IP, nous avons introduit les observateurs dans la partie du code qui sera exécutée lors de l'arrivée d'un paquet au buffer du routeur. L'échantillonnage sera donc aux instants d'arrivée d'un paquet au routeur.

D.1 Script de simulations

```
set ns [new Simulator]

#Ouvrir les fichiers des resultats
set file1 [open out.tr w]
$ns trace-all $file1

#Ouvrir le fichier graphique NAM
set file2 [open out.nam w]
$ns namtrace-all $file2

#Noeuds du routeur et la destination
set N [$ns node]
set D [$ns node]

#Parametres du PI
Queue/PI set a_ 1.822e-5
Queue/PI set b_ 1.816e-5
Queue/PI set w_ 160
Queue/PI set qref_ 175

#Parametres du GAIN_K2
Queue/GAIN_K2 set k1_ -0.321e-3
Queue/GAIN_K2 set k2_ 0.0204e-3
Queue/GAIN_K2 set w_ 160
Queue/GAIN_K2 set NbConnect_ 60
Queue/GAIN_K2 set qref_ 175
Queue/GAIN_K2 set wref_ 15.4166666666667
Queue/GAIN_K2 set pref_ 0.00841490138787

#Parametres du RED
Queue/RED set bytes_ false
Queue/RED set queue_in_bytes_ false
Queue/RED set gentle_ false
Queue/RED set thresh_ 150
Queue/RED set maxthresh_ 700
Queue/RED set q_weight_ 0.0000133
Queue/RED set linterm_ 10
Queue/RED set adaptive_ 0

#Liens entre le routeur et la destination
$ns duplex-link $N $D 15Mb 50ms RED
$ns queue-limit $N $D 800

set NumbSrc 60
set Duration 200

#Creation des noeuds des sources
for {set j 1} {$j<=$NumbSrc} { incr j } {
set S($j) [$ns node]
}

#Liens entre les sources et le routeur
for {set j 1} {$j<=$NumbSrc} { incr j } {
$ns duplex-link $S($j) $N 15Mb 50ms DropTail
$ns queue-limit $S($j) $N 100
}

#Sources TCP
for {set j 1} {$j<=$NumbSrc} { incr j } {
set tcp_src($j) [new Agent/TCP/Newreno]
}
```

```
#Destinations TCP
for {set j 1} {$j<=$NumbSrc} { incr j } {
set tcp_snk($j) [new Agent/TCPSink]
}

#Creations des agents et connexions
for {set j 1} {$j<=$NumbSrc} { incr j } {
$ns attach-agent $S($j) $tcp_src($j)
$ns attach-agent $D $tcp_snk($j)
$ns connect $tcp_src($j) $tcp_snk($j)
}

#Agents FTP pour les sources TCP
for {set j 1} {$j<=$NumbSrc} { incr j } {
set ftp($j) [$tcp_src($j) attach-source FTP]
}
for {set j 1} {$j<=$NumbSrc} { incr j } {
$tcp_src($j) set packetSize_ 500
$tcp_src($j) set window_ 160
$tcp_src($j) set ecn_ 0
}

#Sources UDP
for {set j 1} {$j<=$NumbSrc} { incr j } {
set udp_src($j) [new Agent/UDP] }


#Destinations UDP
for {set j 1} {$j<=$NumbSrc} { incr j } {
set udp_snk($j) [new Agent/Null]
}

#Creations des agents et connexions
for {set j 1} {$j<=$NumbSrc} { incr j } {
$ns attach-agent $S($j) $udp_src($j)
$ns attach-agent $D $udp_snk($j)
$ns connect $udp_src($j) $udp_snk($j)
$udp_src($j) set fid_ 2
}


#Agents CBR pour les sources UDP
for {set j 1} {$j<=3} { incr j } {
set cbr($j) [new Application/Traffic/CBR]

$cbr($j) attach-agent $udp_src($j)
$cbr($j) set type_ PT_CBR
$cbr($j) set packet_size_ 500
$cbr($j) set rate_ 2mb
$cbr($j) set random_ 0
}


#Scenario pour les agents FTP et CBR
for {set i 1} {$i<=$NumbSrc} { incr i } {
$ns at 0.001 "$ftp($i) start"
$ns at Duration "$ftp($i) stop"
}

for {set j 1} {$j<=3} { incr j } {
$ns at 50 "$cbr($j) start"
$ns at 100 "$cbr($j) stop"
}

#Procedure de fin
proc finish {} {
        global ns file1 file2
        $ns flush-trace
        close $file1
        close $file2
        exec nam out.nam &
        exit 0
}
$ns at [expr $Duration] "finish"

#Lancement de la simulation
$ns run
```

D.2 Script de rejeux

```
# Recuperation du nom du fichier info contenant le nombre de flux a rejouer,
# le debut, les RTT et les pertes pour chacun des flux TCP.

set nom_fichierInfo [lindex $argv 0]
set nom_fichierTr [lindex $argv 1]

# Ouverture du fichier
set fichierInfo [open $nom_fichierInfo r]

# Lecture du fichier
set nb_flux [gets $fichierInfo]
set duree_simulation [gets $fichierInfo]
for {set indice 0} {$indice < $nb_flux} {incr indice} {
    set nom_fichierTrace($indice) [gets $fichierInfo]
}

for {set indice 0} {$indice < $nb_flux} {incr indice} {
    set debutFlux($indice) [gets $fichierInfo]
}

for {set indice 0} {$indice < $nb_flux} {incr indice} {
    set rttFlux($indice) [gets $fichierInfo]
}
```

```
for {set indice 0} {$indice < $nb_flux} {incr indice} {
    set pertesFlux($indice) [gets $fichierInfo]
}

puts " - flux TCP a rejouer $nb_flux"
puts " - duree simu $duree_simulation"

# Parametres de la simulation
set debit_lien_coeur 10Mb
set debit_lien_acces 10Mb

set debit_lien_C 0.122Mb
set 1_2_RTT_C 94.3ms

set facteur_taille_file 0.12
# Facteur de diminution trouve de facon empirique.
# Taille paquet min=taille acquittement= 40octets
# => nb max de packets dans la file d'attente

set autre_delai 1ms
set granularite_delai_calcul 5
set unite ms
set granularite_delai_ms $granularite_delai_calcul$unite

# unite granularite rtt : ms
set granularite_rtt 10
set nb_maxi_de_noeuds 1001

for {set indice 0} {$indice < $nb_maxi_de_noeuds} {incr indice} {
    set pertesRTT($indice) [gets $fichierInfo]
}

puts " - nombre de noeuds dans la topologie [expr $nb_maxi_de_noeuds-1] "

 set nb_paquets_par_rtt 10000
 set taille_fenetre_ctrl_flux 10000

#-------------------------------------------------------------------------------#
#                          Description de la simulation                         #
#                                                                               #
#-------------------------------------------------------------------------------#

# Creation un objet simulateur
set ns [new Simulator]

# Creation d'un fichier NAM
set fichierNam [open resultat_simulation.nam w]
$ns namtrace-all $fichierNam

puts " - Creation du fichier resultat"

# Creation d'un fichier de resultats de la simulation
set fichierResultat [open $nom_fichierTr w]
set tout [open out.tr w]
$ns trace-all $tout

puts " - Creation des generateurs de trafic TCP"

# Creation des generateurs de trafic TCP
Tracefile set debug_ false
for {set indice 0} {$indice < $nb_flux} {incr indice} {
    # Fichier de traces
    set fichierTrace($indice) [new Tracefile]
    $fichierTrace($indice) filename $nom_fichierTrace($indice)

    # Generateur de trafic
    set generateur_trafic($indice) [new Application/Traffic/Trace]
    $generateur_trafic($indice) attach-tracefile $fichierTrace($indice)
}
```

```
#-----------------------------------------#
#    Description de la topologie :        #
#   creation des noeuds et des liens      #
#-----------------------------------------#

puts " - Creation de la topologie"

# Brin de coeur
set node_coeur(0) [$ns node]
set node_coeur(1) [$ns node]

$ns duplex-link $node_coeur(0) $node_coeur(1) $debit_lien_coeur $autre_delai DropTail

# Parametrage de la file d'attente
$ns duplex-link-op $node_coeur(0) $node_coeur(1) queuePos 0.5

# Orientation du noeud
$ns duplex-link-op $node_coeur(0) $node_coeur(1) orient  right

# On recupere tout !
$ns trace-all $fichierResultat

# Creation des liens d'acces
set node_acces_C(0) [$ns node]
set N [$ns node]

$ns duplex-link $N $node_coeur(0) 0.1Mb 73.4ms DropTail
$ns duplex-link $node_acces_C(0) $node_coeur(0) $debit_lien_C $1_2_RTT_C DropTail

#-----------------------------------------#
# Creation des differents agents et sources  #
# (avec connection des sources aux agents et  #
#  des agents aux noeuds)                 #
# Connection emetteurs/recepteurs         #
#-----------------------------------------#

puts " - Creation des agents emetteurs"

# Creation des agents et sources emetteurs
for {set indice 0} {$indice < $nb_flux} {incr indice} {
     set agentEmetteur($indice) [new Agent/TCP/Newreno]

    # Connexion des agents sur les noeuds en fonctions des pertes du flux

    puts " Flux $indice ok"

    $ns attach-agent $node_acces_C(0) $agentEmetteur($indice)
    $agentEmetteur($indice) set fid_ $indice
    $agentEmetteur($indice) set tcp_flags_ TF_NODELAY | TF_ACKNOW

   # Important lorsque les files d'attente sont limitees d'avoir
   # une taille basse de la fenetre d'emission
   # $agentEmetteur($indice) set window_ 2
   # $agentEmetteur($indice) set maxpkts_ 2

    $generateur_trafic($indice) attach-agent $agentEmetteur($indice)
}

#source attaquante
set udp_src [new Agent/UDP]
$ns attach-agent $N $udp_src

set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp_src
$cbr set type_ PT_CBR
$cbr set packet_size_ 950
$cbr set rate_ 0.1mb
$cbr set random_ 0

puts " - Creation des agents recepteurs"
```

```
# Creation des agents recepteurs : les recepteurs sont sur le noeud 0
for {set indice 0} {$indice < $nb_flux} {incr indice} {
    set agentRecepteur($indice) [new Agent/TCPSink]

    $ns attach-agent $node_coeur(1) $agentRecepteur($indice)
}

# agents recepteurs de l'attaque
set udp_snk [new Agent/Null]
$ns attach-agent $node_coeur(1) $udp_snk

puts " - Connexion des differents agents"

# Connection emetteurs/recepteurs
for {set indice 0} {$indice < $nb_flux} {incr indice} {
    $ns connect $agentEmetteur($indice) $agentRecepteur($indice)
}

# Connection attaque
$ns connect $udp_src $udp_snk

#configuration de TBF
set tbf [new TBF2]
$tbf set bucket_ 12160
$tbf set rate_ 0.15m
$tbf set qlen_  20

set lnk [[$ns link $node_coeur(0) $node_coeur(1)] set link_]
$tbf target [$lnk target]
$lnk target $tbf

#-------------------------#
# Mise en place du scenario  #
#-------------------------#

puts " - Dynamique de la simulation"

# Description du scenario
for {set indice 0} {$indice < $nb_flux} {incr indice} {
    $ns at $debutFlux($indice) "$generateur_trafic($indice) start"
}

# Description de l'attaque
$ns at 90 "$cbr start"
$ns at 95.84 "$cbr stop"
$ns at 101.3 "$cbr start"
$ns at 108.9 "$cbr stop"
$ns at 112.3 "$cbr start"
$ns at 119.6 "$cbr stop"
$ns at 124.3 "$cbr start"
$ns at 132.8 "$cbr stop"

$ns at 269 "$cbr start"
$ns at 510 "$cbr stop"
$ns at 585 "$cbr start"
$ns at 826 "$cbr stop"
$ns at 994 "$cbr start"
$ns at 1235 "$cbr stop"
$ns at 1440 "$cbr start"
$ns at 1681 "$cbr stop"


$ns at [expr (1*$duree_simulation)] "finish"

# Choix des resultats de la simulation

puts " - Fin de la definition de la simulation"
puts " - Debut de la simulation"
```

```
# Procedure de fin
proc finish {} {
        global ns
        global fichierNam
        global fichierResultat
        global tout
        $ns flush-trace
# Ferme les fichiers resultats
        close $fichierNam
        close $fichierResultat
        close $tout
# Execute nam on the trace file
        exec nam resultat_simulation.nam &
        exit 0
}

# Lancement de la simulation
$ns run
```

D.3 Codes des observateurs

D.3.1 de Luenberger minimal IOD

```
 void REDQueue::enque(Packet* pkt)
{
double now = Scheduler::instance().clock();

ArrivedPkt=ArrivedPkt+1;
hdr_tcp* tcph = hdr_tcp::access(pkt);
CongFen = tcph->get_fenetre();
double rtt = tcph->last_rtt();
hdr_ip* iph = hdr_ip::access(pkt);
int ip_addr=iph->saddr();
int qlen2=qib_ ? q_->byteLength():q_->length();

////////////Observateur IOD////////////

hdr_cmn* cb = hdr_cmn::access(pkt);
if (cb->ptype() == PT_CBR)
     { d_count++;
       }

if((NOW-old_now)>=0.001){
int i;

for(i=0;i<5;i++){
   z[i]=z[i]+(NOW-old_now)*z_pt[i];
 }

q_pt=(qlen2-qlen_old)/(NOW-old_now);
old_now=NOW;

//////valeurs retardees de W, q, p/////

for(i=80;i>0;i--){
   tab_w[i]=tab_w[i-1];
   tab_q[i]=tab_q[i-1];
   tab_p[i]=tab_p[i-1];
   tab_temps[i]=tab_temps[i-1];
  }

tab_w[0]=z[0];
tab_q[0]=qlen2;
tab_p[0]=edv_.v_prob;
tab_temps[0]=NOW;

int cmpt_h=0;

while((NOW-tab_temps[cmpt_h])<h*0.001){
       cmpt_h=cmpt_h+1;
       }

zd[0]=tab_w[cmpt_h];
zd[1]=tab_q[cmpt_h];
pd=tab_p[cmpt_h];

////////dynamiques d'observation////////

z_pt[0]=(a11-l1*a21)*z[0]-l1*z[1]+ad1*zd[0]
        +a12*(qlen2-qref)+ad2*(zd[1]-qref)
        +b1*(pd-pref)+l1*q_pt
        -l1*a22*(qlen2-qref);
z_pt[1]=-l2*a21*z[0]-l2*z[1]+z[2]
        +l2*q_pt-l2*a22*(qlen2-qref);
z_pt[2]=-l3*a21*z[0]-l3*z[1]+l3*q_pt
        -l3*a22*(qlen2-qref);
z_pt[3]=-l4*a21*z[0]-l4*z[1]+l4*q_pt
        -l4*a22*(qlen2-qref);
z_pt[4]=-l5*a21*z[0]-l5*z[1]+l5*q_pt
        -l5*a22*(qlen2-qref);

qlen_old=qlen2;

///valeurs sauvegardees dans des fichiers///

if(NOW>0.4 && ip_addr>=2){
fichier_w_real <<NOW<<" "<<CongFen<< endl;
fichier_w_est <<NOW<<" "<<(z[0]+wref)<< endl;
fichier_d_est <<NOW<<" "<<(z[1])<< endl;
fichier_dpt_est <<NOW<<" "<<(z[2])<< endl;
fichier_d2pt_est <<NOW<<" "<<(z[3])<< endl;
 }
}
//////////////////////////////////////////

void DctTimer::expire(Event *)
{
double rate = a_->d_count/delay;
if (qId_==0)
     fichier_drate <<NOW<<" "<<(rate)<< endl;
     a_->d_count=0;
     this->resched(delay);
}
```

D.3.2 de Luenberger DD

```
void REDQueue::enque(Packet* pkt)
{
double now = Scheduler::instance().clock();
ArrivedPkt=ArrivedPkt+1;
hdr_tcp* tcph = hdr_tcp::access(pkt);
CongFen = tcph->get_fenetre();
double rtt = tcph->last_rtt();
hdr_ip* iph = hdr_ip::access(pkt);
int ip_addr=iph->saddr();
int qlen2=qib_ ? q_->byteLength():q_->length();

////////////Observateur DD////////////

hdr_cmn* cb = hdr_cmn::access(pkt);
if (cb->ptype() == PT_CBR)
{ d_count++;
  }

if((NOW-old_now)>=0.001){
int i;

for(i=0;i<6;i++){
   z[i]=z[i]+(NOW-old_now)*z_pt[i];
 }

q_pt=(qlen2-qlen_old)/(NOW-old_now);
old_now=NOW;

//////valeurs retardees de W, q, p/////

for(i=80;i>0;i--){
   tab_w[i]=tab_w[i-1];
   tab_q[i]=tab_q[i-1];
   tab_p[i]=tab_p[i-1];
   tab_temps[i]=tab_temps[i-1];
  }

tab_w[0]=z[0];
tab_q[0]=qlen2;
tab_p[0]=edv_.v_prob;
tab_temps[0]=NOW;

int cmpt_h=0;

while((NOW-tab_temps[cmpt_h])<h*0.001){
        cmpt_h=cmpt_h+1;
        }

zd[0]=tab_w[cmpt_h];
zd[1]=tab_q[cmpt_h];
pd=tab_p[cmpt_h];

////////dynamiques d'observation////////

z_pt[0]=a11*z[0]+a12*z[1]+ad1*zd[0]
        +ad2*(zd[1]-qref)+b1*(pd-pref)
        +l1*(qlen2-qref-z[1]);
z_pt[1]=a21*z[0]+a22*z[1]+z[2]
        +l2*(qlen2-qref-z[1]);
z_pt[2]=z[3]+l3*(qlen2-qref-z[1]);
z_pt[3]=z[4]+l4*(qlen2-qref-z[1]);
z_pt[4]=z[5]+l5*(qlen2-qref-z[1]);
z_pt[5]=l6*(qlen2-qref-z[1]);

qlen_old=qlen2;

///valeurs sauvegardees dans des fichiers///

if(NOW>0.4 && ip_addr>=2){
fichier_w_real <<NOW<<" "<<CongFen<< endl;
fichier_w_est <<NOW<<" "<<(z[0]+wref)<< endl;
fichier_q_real <<NOW<<" "<<qlen2<< endl;
fichier_q_est <<NOW<<" "<<(z[1]+qref)<< endl;
fichier_d_est <<NOW<<" "<<(z[2])<< endl;
fichier_dpt_est <<NOW<<" "<<(z[3])<< endl;
fichier_d2pt_est <<NOW<<" "<<(z[4])<< endl;
 }
}
////////////////////////////////////////////

void DctTimer::expire(Event *)
{
double rate = a_->d_count/delay;
if (qId_==0)
    fichier_drate <<NOW<<" "<<(rate)<< endl;
    a_->d_count=0;
    this->resched(delay);
}
```

D.3.3 glissant d'ordre 1 avec le filtre d'ordre 3

```
void REDQueue::enque(Packet* pkt)
{
double now = Scheduler::instance().clock();
ArrivedPkt=ArrivedPkt+1;
hdr_tcp* tcph = hdr_tcp::access(pkt);
CongFen = tcph->get_fenetre();
double rtt = tcph->last_rtt();
hdr_ip* iph = hdr_ip::access(pkt);
int ip_addr=iph->saddr();
int qlen2=qib_ ? q_->byteLength():q_->length();

//////Observateur glissant d'ordre 1//////

hdr_cmn* cb = hdr_cmn::access(pkt);
if (cb->ptype() == PT_CBR)
{ d_count++;
  }

if((NOW-old_now)>=0.001){
int i;

for(i=0;i<2;i++){
    z[i]=z[i]+(NOW-old_now)*z_pt[i];
 }

d_newpt2=d_pt2+(NOW-old_now)*d_pt3;
d_newpt1=d_pt1+(NOW-old_now)*d_pt2;
d=d+(NOW-old_now)*d_pt1;

old_now=NOW;

//////valeurs retardees de W, q, p/////

for(i=80;i>0;i--){
   tab_w[i]=tab_w[i-1];
```

```
    tab_q[i]=tab_q[i-1];
    tab_p[i]=tab_p[i-1];
    tab_temps[i]=tab_temps[i-1];
    }

tab_w[0]=z[0];
tab_q[0]=qlen2;
tab_p[0]=edv_.v_prob;
tab_temps[0]=NOW;

int cmpt_h=0;
double nu;

while((NOW-tab_temps[cmpt_h])<h*0.001){
       cmpt_h=cmpt_h+1;
       }

zd[0]=tab_w[cmpt_h];
zd[1]=tab_q[cmpt_h];
pd=tab_p[cmpt_h];

nu=-k*(z[1]-qlen2+qref)/abs(z[1]-qlen2+qref);
d_pt2=d_newpt2;
d_pt1=d_newpt1;

////////dynamiques d'observation////////
```

```
z_pt[0]=a11*z[0]+ad1*zd[0]+a12*(qlen2-qref)
        +ad2*(zd[1]-qref)+b1*(pd-pref);
z_pt[1]=a21*z[0]+a22*(qlen2-qref)
        +l*(z[1]-qlen2+qref)+nu;
d_pt3=(1/tho*tho*tho)*(nu-3*(tho*tho)*d_pt2
       -3*tho*d_pt1-d);

///valeurs sauvegardees dans des fichiers///

if(NOW>0.4 && ip_addr>=2){
fichier_w_real <<NOW<<" "<<CongFen<< endl;
fichier_w_est <<NOW<<" "<<(z[0]+wref)<< endl;
fichier_q_est <<NOW<<" "<<(z[1]+qref)<< endl;
fichier_q_real <<NOW<<" "<<(qlen2)<< endl;
fichier_d_est <<NOW<<" "<<(d)<< endl;
 }
}
/////////////////////////////////////////

void DctTimer::expire(Event *)
{
double rate = a_->d_count/delay;
if (qId_==0)
    fichier_drate <<NOW<<" "<<(rate)<< endl;
    a_->d_count=0;
    this->resched(delay);
}
```

D.3.4 glissant d'ordre 2

```
void REDQueue::enque(Packet* pkt)
{
double now = Scheduler::instance().clock();
ArrivedPkt=ArrivedPkt+1;
hdr_tcp* tcph = hdr_tcp::access(pkt);
CongFen = tcph->get_fenetre();
double rtt = tcph->last_rtt();
hdr_ip* iph = hdr_ip::access(pkt);
int ip_addr=iph->saddr();
int qlen2=qib_ ? q_->byteLength():q_->length();

//////Observateur glissant d'ordre 2//////

hdr_cmn* cb = hdr_cmn::access(pkt);
if (cb->ptype() == PT_CBR)
  {d_count++;
   }

if((NOW-old_now)>=0.001){
int i;

for(i=0;i<3;i++){
   z[i]=z[i]+(NOW-old_now)*z_pt[i];
 }

old_now=NOW;

//////valeurs retardees de W, q, p/////

for(i=80;i>0;i--){
   tab_w[i]=tab_w[i-1];
   tab_q[i]=tab_q[i-1];
   tab_p[i]=tab_p[i-1];
   tab_temps[i]=tab_temps[i-1];
 }

tab_w[0]=z[0];
```

```
tab_q[0]=qlen2;
tab_p[0]=edv_.v_prob;
tab_temps[0]=NOW;

int cmpt_h=0;
double signe;

while((NOW-tab_temps[cmpt_h])<h*0.001){
       cmpt_h=cmpt_h+1;
        }

zd[0]=tab_w[cmpt_h];
zd[1]=tab_q[cmpt_h];
pd=tab_p[cmpt_h];

signe=(z[1]-qlen2+qref)/abs(z[1]-qlen2+qref);

////////dynamiques d'observation////////

z_pt[0]=a11*z[0]+ad1*zd[0]+a12*(qlen2-qref)
        +ad2*(zd[1]-qref)+b1*(pd-pref);
z_pt[1]=a21*z[0]+a22*(qlen2-qref)+z[2]
      -lambda*sqrt(abs(z[1]-qlen2+qref))*signe;
z_pt[2]=-alpha*signe;

///valeurs sauvegardees dans des fichiers///

if(NOW>0.4 && ip_addr>=2){
fichier_w_real <<NOW<<" "<<CongFen<< endl;
fichier_w_est <<NOW<<" "<<(z[0]+wref)<< endl;
fichier_q_est <<NOW<<" "<<(z[1]+qref)<< endl;
fichier_q_real <<NOW<<" "<<(qlen2)<< endl;
fichier_d_est <<NOW<<" "<<(z[2])<< endl;
 }
}
/////////////////////////////////////////
```

```
void DctTimer::expire(Event *)
{
double rate = a_->d_count/delay;
if (qId_==0)
    fichier_drate <<NOW<<" "<<(rate)<< endl;
    a_->d_count=0;
    this->resched(delay);
}
```

Bibliographie

[Ali Ahammed 2010] G. F. Ali Ahammed et R. Banu. *Analyzing the Performance of Active Queue Management Algorithms.* Computing Research Repository, CoRR, 2010. (Cité en page 13.)

[Altman 2005] E. Altman, K. Avrachenkov et C. Barakat. *A stochastic model of TCP/IP with stationary random losses.* IEEE/ACM Transactions on Networking, vol. 13, no. 2, pages 356–369, April 2005. (Cité en page 20.)

[Ariba 2008] Y. Ariba, Y. Labit et F. Gouaisbaut. *Network anomaly estimation for TCP/AQM networks using an observer.* 3^{rd} ACM International Workshop on Feedback Control Implementation and Design in Computing Systems and Networks, FeBID'08, pages 3818–3823, June 2008. (Cité en pages 20, 78 et 93.)

[Ariba 2009] Y. Ariba, F. Gouaisbaut et Y. Labit. *Feedback control for router management and TCP/IP network stability.* IEEE Transactions on Network and Service Management, vol. 6, no. 4, pages 255–266, December 2009. (Cité en pages 14, 24, 37, 42 et 143.)

[Aussibal 2007] J. Aussibal, P. Borgnat, Y. Labit, N. Dewaele G. Larrieu, L. Gallon, P. Owezarski, P. Abry et K. Boudaoud. *Base de traces d'anomalies légitimes et illégitimes.* In Proceedings of the 2^{nd} Conference on Security in Network Architectures and Information Systems, SAR-SSI 2007, pages 153–168, Annecy, France, June 2007. (Cité en pages 15, 17, 35 et 42.)

[Bai 2007] L. Bai, Z. Tian et S. Shi. *RFDF design for linear time-delay systems with unknown inputs and parameter uncertainties.* International Journal of Systems Science, vol. 38, pages 139–149, February 2007. (Cité en page 32.)

[Barford 2002] P. Barford, J. Kline, D. Plonka et A. Ron. *A signal analysis of network traffic anomalies.* In Proceedings of the 2^{nd} ACM SIGCOMM Workshop on Internet measurment, IMW'02, pages 71–82, Marseille, France, 2002. (Cité en pages 18 et 19.)

[Barlow 2000] J. Barlow et W. Thrower. *TFN2K - An Analysis.* AXENT Security Team, March 2000. URL :http ://www.packetstormsecurity.org/distributed/TFN2k_Analysis-1.3.txt. (Cité en pages 16 et 113.)

[Bartolini 1986] G. Bartolini et T. Zolezzi. *Control of nonlinear variable structure systems.* Journal of Mathematical Analysis and Applications, vol. 118, pages 42–62, 1986. (Cité en page 67.)

[Bartolini 1998] G. Bartolini, A. Ferrara et E. Usani. *Chattering avoidance by second-order sliding mode control.* IEEE Transactions on Automatic Control, vol. 43, no. 2, pages 241–246, February 1998. (Cité en pages 70 et 73.)

[Bernstein 1912] S. Bernstein. *Démonstration du théorème de Weierstrass, fondée sur le calcul des probabilités.* Communications of the Kharkov Mathematical Society, vol. 13, pages 1–2, 1912. (Cité en pages 41 et 42.)

[Borgnat 2007] P. Borgnat, P. Abry, G. Dewaele, A. Scherrer, N. Larrieu, Ph. Owezarski, Y. Labit, L. Gallon et J. Aussibal. *Une Caractérisation non gaussienne et à longue mémoire du trafic Internet et de ses anomalies.* Annales des telecommunications-annals of telecommunications, vol. 62, no. 11-12, pages 1401–1428, December 2007. (Cité en page 18.)

[Borgnat 2009] P. Borgnat, G. Dewaele, K. Fukuda, P. Abry et K. Cho. *Seven Years and One Day : Sketching the Evolution of Internet Traffic.* In Proceedings of the 28^{th} IEEE INFOCOM'09, pages 711–719, May 2009. (Cité en page 18.)

[Boutayeb 2001] M. Boutayeb. *Observers design for linear time-delay systems.* Systems & Control Letters, vol. 44, pages 103–109, 2001. (Cité en page 30.)

[Boyd 1994] S. Boyd, L. El Ghaoui, E. Feron et V. Balakrishnan. Linear Matrix Inequalities in System and Control Theory. Society for Industrial and Applied Mathematics, SIAM, Philadelphia, USA, 1994. (Cité en pages 27 et 149.)

[Braden 1998] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski et L. Zhang. *Recommendations on Queue Management and Congestion Avoidance in the Internet*. Rapport technique 2309, Internet Engineering Task Force, IETF, April 1998. (Cité en page 12.)

[Breda 2009] Dimitri Breda, Stefano Maset et Rossana Vermiglio. *TRACE-DDE : a Tool for Robust Analysis and Characteristic Equations for Delay Differential Equations*. In Topics in Time Delay Systems, volume 388 of *Lecture Notes in Control and Information Sciences*, pages 145–155. Springer Berlin / Heidelberg, 2009. (Cité en pages 52 et 80.)

[Cao 2009] J. Cao et M. Stefanovic. *Switching congestion control for satellite TCP/AQM networks*. In Proceedings of the 2009 conference on American Control Conference, ACC'09, pages 4842–4847, St. Louis, Missouri, USA, 2009. (Cité en pages 14 et 20.)

[Chandola 2009] V. Chandola, A. Banerjee et V. Kumar. *Anomaly detection : A survey*. ACM Computing Surveys (CSUR), vol. 41, pages 1–58, July 2009. (Cité en pages 17, 18 et 19.)

[Chen 1996] J. Chen, R.J. Patton et H.Y. Zhang. *Design of unknown input observers and robust fault-detection filters*. International Journal of Control, vol. 63, no. 1, pages 85–105, 1996. (Cité en page 29.)

[Chen 1999] J. Chen et R.J. Patton. Robust model-based fault diagnosis for dynamic systems. Kluwer Academic Publishers, Norwell, MA, USA, 1999. (Cité en page 28.)

[Chen 2006] Y. Chen et K. Hwang. *Collaborative detection and filtering of shrew DDoS attacks using spectral analysis*. Journal of Parallel Distributed Computing, vol. 66, no. 9, pages 1137–1151, 2006. (Cité en page 15.)

[Clark 1992] D. Clark, S. Shenker et L. Zhang. *Supporting real-time applications in an Integrated Services Packet Network : architecture and mechanism*. In Proceedings of the conference on Communications architectures & protocols, SIGCOMM '92, pages 14–26, Baltimore, Maryland, USA, 1992. (Cité en page 38.)

[Cleary 2000] J. Cleary, S. Donnelly, I. Graham, A. Mcgregor et M. Pearson. *Design principles for accurate passive measurement*. In The First Passive and Active Measurement Workshop, PAM'2000, pages 1–7, Hamilton, New Zealand, April 2000. (Cité en pages 17 et 110.)

[Corless 1998] M. Corless et J. Tu. *State and input estimation for a class of uncertain systems*. Automatica, vol. 34, June 1998. (Cité en page 29.)

[Darouach 1999] M. Darouach, P. Pierrot et E. Richard. *Design of reduced-order observers without internal delays*. IEEE Transactions on Automatic Control, vol. 44, pages 1711–1713, September 1999. (Cité en pages 30, 31 et 32.)

[Darouach 2001] M. Darouach. *Linear functional observers for systems with delays in state variables*. IEEE Transactions on Automatic Control, vol. 46, no. 3, pages 491–496, March 2001. (Cité en pages 30 et 32.)

[Defoort 2009] M. Defoort, F. Nollet, T. Floquet et W. Perruquetti. *A Third-Order Sliding-Mode Controller for a Stepper Motor*. IEEE Transactions on Industrial Electronics, vol. 56, no. 9, pages 3337–3346, September 2009. (Cité en pages 34, 69 et 70.)

[Denning 1987] D.E. Denning. *An Intrusion-Detection Model*. IEEE Transactions on Software Engineering - Special issue on computer security and privacy, vol. 13, pages 222–232, February 1987. (Cité en pages 17 et 18.)

[Ding 2001] S.X. Ding, Zhong Maiying, Tang Bingyong et P. Zhang. *An LMI approach to the design of fault detection filter for time-delay LTI systems with unknown inputs.* In Proceedings of the American Control Conference, ACC'01, volume 3, pages 2137–2142, 2001. (Cité en page 32.)

[Dittrich 1999] D. Dittrich. *The DoS Project's Trinoo distributed denial of service attack tool.* http ://staff.washington.edu/dittrich/misc/tinoo.analysis, 1999. (Cité en page 16.)

[Drakunov 1995] S. Drakunov et V. Utkin. *Sliding mode observers. Tutorial.* In Proceedings of the 34^{th} IEEE Conference on Decision and Control, CDC'95, volume 4, pages 3376–3378, December 1995. (Cité en page 33.)

[Driver 1977] D.R. Driver. Ordinary and delay differential equations. Springer-Verlag, New York, USA, 1977. (Cité en page 26.)

[Edwards 1998] C. Edwards et S. Spurgeon. Sliding mode control : Theory and applications. Taylor & Francis, 1998. (Cité en page 75.)

[Edwards 2000] C. Edwards, S.K. Spurgeon et R.J. Patton. *Sliding mode observers for fault detection and isolation.* Automatica, vol. 36, pages 541–553, 2000. (Cité en page 77.)

[Edwards 2004] C. Edwards. *A comparison of sliding mode and unknown input observers for fault reconstruction.* In Proceedings of the 43^{rd} IEEE Conference on Decision and Control, CDC'04, volume 5, pages 5279–5284, Atlantis, Paradise Island, Bahamas, December 2004. (Cité en pages 29 et 32.)

[Emel'yanov 2005] S. V. Emel'yanov, S. K. Korovin et A. Levant. *High-order sliding modes in control systems.* Computational Mathematics and Modeling, vol. 7, pages 294–318, 2005. (Cité en pages 69, 72 et 83.)

[Fall 2002] K. Fall et K. Varadhan. *The ns Manual.* notes and documentation on the software ns2-simulator, 2002. URL : www.isi.edu/nsnam/ns/. (Cité en page 153.)

[Fall 2010] K. Fall et K. Varadhan. *The NS Manual*, 2010. www.isi.edu/nsnam/ns/. (Cité en pages 9, 34 et 91.)

[Feng 1999] W.-C. Feng, D.D. Kandlur, D. Saha et K.G. Shin. *A self-configuring RED gateway.* In Proceedings of the Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM'99, volume 3, pages 1320–1328, New York, NY, USA, March 1999. (Cité en page 13.)

[Feng 2002] W.-C. Feng, K.G. Shin, D.D. Kandlur et D. Saha. *The BLUE active queue management algorithms.* IEEE/ACM Transactions on Networking, vol. 10, pages 513–528, August 2002. (Cité en page 13.)

[Filippov 1988] A.F. Filippov. Differential equations with discontinuous righthand sides. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1988. (Cité en page 66.)

[Fliess 2005] M. Fliess, C. Join et H. Mounier. Advances in communication control networks, chapitre An Introduction to Nonlinear Fault Diagnosis with an Application to a Congested Internet Router, pages 393–395. Lecture notes in Control and Information Sciences. Springer, 2005. (Cité en pages 24 et 37.)

[Floquet 2000] T. Floquet. *Contributions à la commande par mode glissants d'ordre supérieur.* PhD thesis, Ecole Centrale de Lille Université des Sciences et Technologie de Lille, Lille I, 2000. (Cité en pages 33 et 72.)

[Floquet 2007a] T. Floquet et J.P. Barbot. *Super twisting algorithm based step-by-step sliding mode observers for nonlinear systems with unknown inputs.* International Journal of Systems Science, vol. 38, no. 10, pages 803–815, 2007. (Cité en pages 73 et 83.)

[Floquet 2007b] T. Floquet, C. Edwards et S. Spurgeon. *On sliding mode observers for systems with unknown inputs.* International Journal of Adaptive Control and Signal Processing, vol. 21, no. 8-9, pages 638–656, 2007. (Cité en page 32.)

[Floyd 1993] S. Floyd et V. Jacobson. *Random early detection gateways for congestion avoidance.* IEEE/ACM Transactions on Networking, vol. 1, pages 397–413, August 1993. (Cité en pages 12, 13, 51 et 93.)

[Floyd 2001] S. Floyd et V. Paxson. *Difficulties in simulating the Internet.* IEEE/ACM Transactions on Networking, vol. 9, no. 4, pages 392–403, August 2001. (Cité en page 110.)

[Frank 1994] P. M. Frank. *On-line fault detection in uncertain nonlinear systems using diagnostic observers ; a survey.* International Journal of Systems Science, vol. 25, no. 12, pages 2129–2154, 1994. (Cité en page 29.)

[Fridman 2002] L. Fridman et A. Levant. Sliding mode control in engineering, chapitre Higher-Order Sliding Modes. Marcel Dekker, Inc, New York, 2002. (Cité en page 72.)

[Fridman 2005] E. Fridman et U. Shaked. *Stability and Guaranteed Cost Control of Uncertain Discrete Delay Systems.* International Journal of Control, vol. 78, no. 4, pages 235–246, Mars 2005. (Cité en page 27.)

[Fridman 2006] E. Fridman. *Stability of systems with uncertain delays : a new "Complete" Lyapunov-krasovskii functional.* IEEE Transactions on Automatic Control, vol. 51, no. 5, pages 885–890, May 2006. (Cité en pages 25 et 27.)

[Friedland 1996] B. Friedland. Observers, chapitre 37, pages 607–618. Levine, W.S., edition, CRC Press, 1996. (Cité en page 29.)

[Gertler 1998] J.J. Gertler. Fault detection and diagnosis in engineering systems. Marcel Dekker, 1998. (Cité en page 28.)

[Gouaisbaut 2006] F. Gouaisbaut et D. Peaucelle. *Delay-Dependent stability analysis of linear time delay systems.* In IFAC Workshop on Time Delay System, TDS'06, Aquila, Italy, Juillet 2006. (Cité en page 58.)

[Gu 2003] K. Gu, V. L. Kharitonov et J. Chen. Stability of time-delay systems. Birkhäuser Boston, 2003. Control engineering. (Cité en pages 25, 26, 27, 47, 50 et 76.)

[Guan 1991] Y. Guan et M. Saif. *A novel approach to the design of unknown input observers.* Automatic Control, IEEE Transactions on, vol. 36, no. 5, pages 632–635, May 1991. (Cité en pages 29 et 30.)

[Hale 1993] J.K. Hale et S.M. Verduyn Lunel. Introduction to functional differential equations. Applied Mathematical Sciences, Volume 99. Springer-Verlag, New York, USA, 1993. (Cité en pages 26 et 27.)

[Hammouri 2010] H. Hammouri et Z. Tmar. *Unknown input observer for state affine systems : A necessary and sufficient condition.* Automatica, vol. 46, no. 2, pages 271–278, 2010. (Cité en page 29.)

[Hassan 2004] M. Hassan et R. Jain. High performance TCP/IP networking : Concepts, issues, and solutions. Prentice-Hall, USA, 2004. (Cité en pages 6, 7, 9 et 12.)

[Hodge 2004] V.J. Hodge et J. Austin. *A survey of outlier detection methodologies.* Artificial Intelligence Review, vol. 22, pages 85–126, 2004. (Cité en page 17.)

[Hollot 2001a] C. V. Hollot, V. Misra, D Towsley et W. Gong. *A control theoretic analysis of RED.* In Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM'01, volume 3, pages 1510–1519, April 2001. (Cité en page 14.)

[Hollot 2001b] C. V. Hollot, V. Misra, D Towsley et W. Gong. *On designing improved controllers for AQM routers supporting TCP flows.* In Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM'01, volume 3, pages 1726–1734, Anchorage, Alaska, April 2001. (Cité en pages 23 et 37.)

[Hollot 2001c] C.V. Hollot et Y. Chait. *Nonlinear stability analysis for a class of TCP/AQM networks.* In Proceedings of the 40^{th} IEEE Conference on Decision and Control, CDC'01, volume 3, pages 2309–2314, Orlando, Florida, December 2001. (Cité en page 14.)

[Hollot 2002] C.V. Hollot, V. Misra, D. Towsley et Weibo Gong. *Analysis and design of controllers for AQM routers supporting TCP flows.* IEEE Transactions on Automatic Control, vol. 47, pages 945–959, June 2002. (Cité en pages 14, 20, 51 et 93.)

[Hou 1992] M. Hou et P.C. Muller. *Design of observers for linear systems with unknown inputs.* IEEE Transactions on Automatic Control, vol. 37, no. 6, pages 871–875, June 1992. (Cité en page 29.)

[Hou 1996] M. Hou et R.J. Patton. *An LMI approach to H_-/H_∞ ; fault detection observers.* In International Conference on Control, UKACC'96, volume 1, pages 305–310, September 1996. (Cité en page 28.)

[Hussain 2003] A. Hussain, J. Heidemann et C. Papadopoulos. *A framework for classifying denial of service attacks.* In Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communications, SIGCOMM'03, pages 99–110, Karlsruhe, Germany, August 2003. (Cité en pages 15, 17 et 37.)

[Isidori 1995] Alberto Isidori. Nonlinear control systems. Springer-Verlag New York, Inc., 3^{rd} edition, Secaucus, NJ, USA, 1995. (Cité en page 70.)

[Jacobson 1988] V. Jacobson. *Congestion avoidance and control.* In Proceedings of the Symposium on Communications Architectures and Protocols, SIGCOMM'88, pages 314–329, Stanford, California, United States, 1988. (Cité en page 8.)

[Jacobson 1990] V. Jacobson. *Modified TCP Congestion Avoidance Algorithm.* April 1990. (Cité en page 8.)

[Jiang 2005] C. Jiang et D.H. Zhou. *Fault detection and identification for uncertain linear time-delay systems.* Computers & Chemical Engineering, vol. 30, no. 2, pages 228–242, 2005. (Cité en page 32.)

[Juanole 1995] G. Juanole, A. Serhrouchni et D. Seret. Réseaux de communication et conception de protocoles. Hermès, Paris, France, 1995. (Cité en page 12.)

[Jung 2002] J. Jung, B. Krishnamurthy et M. Rabinovich. *Flash crowds and denial of service attacks : characterization and implications for CDNs and web sites.* In Proceedings of the 11^{th} international conference on World Wide Web, WWW'02, pages 293–304, Honolulu, Hawaii, USA, 2002. (Cité en pages 15, 17 et 18.)

[Kalsi 2010] K. Kalsi, J. Lian, S. Hui et S.H. Zak. *Sliding-mode observers for systems with unknown inputs : A high-gain approach.* Automatica, vol. 46, no. 2, pages 347–353, 2010. (Cité en page 32.)

[Kelly 1998] F.P. Kelly, A.K. Maulloo et D.K.H. Tan. *Rate Control for Communication Networks : Shadow Prices, Proportional Fairness and Stability.* The Journal of the Operational Research Society, vol. 49, no. 3, pages 237–252, March 1998. (Cité en page 14.)

[Krishnamurthy 2003] B. Krishnamurthy, S. Sen, Y. Zhang et Y. Chen. *Sketch-based change detection : methods, evaluation, and applications.* In Proceedings of the 3^{rd} ACM SIGCOMM Conference on Internet measurement, IMC'03, pages 234–247, New York, NY, USA, 2003. (Cité en page 18.)

[Kumar 1998] A. Kumar. *Comparative performance analysis of versions of TCP in a local network with a lossy link.* IEEE/ACM Transactions on Networking, vol. 6, pages 485–498, August 1998. (Cité en page 20.)

[Kunniyur 2001] S. Kunniyur et R. Srikant. *Analysis and design of an adaptive virtual queue (AVQ) algorithm for active queue management.* In Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM'01, pages 123–134, San Diego, California, USA, April 2001. (Cité en page 14.)

[Lakhina 2004] A. Lakhina, M. Crovella et C. Diot. *Diagnosing network-wide traffic anomalies.* In Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communications, SIGCOMM'04, pages 219–230, Portland, Oregon, USA, 2004. (Cité en pages 15 et 37.)

[Lee 1981] E. B. Lee et A. Olbrot. *Observability and related structural results for linear hereditary systems.* International Journal of Control, vol. 34, no. 6, pages 1061–1078, 1981. (Cité en page 44.)

[Levant 1993] A. Levant. *Sliding order and sliding accuracy in sliding mode control.* International Journal of Control, vol. 58, no. 6, pages 1247–1263, 1993. (Cité en pages 69, 70, 72, 73, 74, 83 et 84.)

[Levant 1998] Arie Levant. *Robust exact differentiation via sliding mode technique.* Automatica, vol. 34, pages 379–384, March 1998. (Cité en page 74.)

[Li 1997] X. Li et C. E. de Souza. *Criteria for robust stability and stabilization of uncertain linear systems with state delay.* Automatica, vol. 33, pages 1657–1662, Septembre 1997. (Cité en page 14.)

[Liu 2003] S. Liu, T. Basar et R. Srikant. *Controlling the Internet : a survey and some new results.* In Proceedings of the 42^{nd} IEEE Conference on Decision and Control, CDC'03, volume 3, pages 3048–3057, December 2003. (Cité en page 14.)

[Low 2002] H. S. Low, F. Paganini et J.C. Doyle. Internet congestion control, volume 22, pages 28–43. IEEE Control Systems Magazine, February 2002. (Cité en pages 20 et 143.)

[Luenberger 1971] D. Luenberger. *An introduction to observers.* IEEE Transactions on Automatic Control, vol. 16, no. 6, pages 596–602, December 1971. (Cité en pages 28 et 29.)

[Maki 2005] I. Maki, G. Hasegawa, M. Murata et T. Murase. *Performance analysis and improvement of TCP proxy mechanism in TCP overlay networks.* In IEEE International Conference on Communications, ICC'05, volume 1, pages 184–190, May 2005. (Cité en page 20.)

[Manfredi 2004] S. Manfredi, M. di Bernardo et F. Garofalo. *Robust output feedback active queue management control in TCP networks.* In Proceeding of the 43^{rd} IEEE Conference on Decision and Control, CDC'04, volume 1, pages 1004–1009, Atlantis, Paradise Island, Bahamas, December 2004. (Cité en page 14.)

[Martinez 2008] R. Martinez, J. Alvarez et Y. Orlov. *Hybrid Sliding Mode-Based Control of Underactuated Systems with Dry Friction.* IEEE Transactions on Industrial Electronics, vol. 55, no. 11, pages 3998–4003, 2008. (Cité en page 70.)

[May 2000] Martin May, Christophe Diot, Bryan Lyles et Jean Bolot. *Influence of Active Queue Management Parameters on Aggregate Traffic Performance.* Research Report RR-3995, INRIA, 2000. (Cité en page 13.)

[Meskin 2009] N. Meskin et K. Khorasani. *Robust fault detection and isolation of time-delay systems using a geometric approach.* Automatica, vol. 45, no. 6, pages 1567–1573, 2009. (Cité en page 32.)

[Michiels 2005] W. Michiels et S.-I. Niculescu. *Stability Analysis of a Fluid Flow Model for TCP like Behavior*. International Journal of Bifurcation and Chaos, vol. 15, no. 7, pages 2277–2282, 2005. (Cité en page 14.)

[Mirkovic 2004] J. Mirkovic. *A taxonomy of DDoS attacks and DDoS defense mechanisms*. ACM SIGCOMM Computer Communication Review, vol. 34, pages 39–53, 2004. (Cité en page 16.)

[Misra 1999] V. Misra, W. Gong et D. Towsley. *Stochastic Differential Equation Modeling and Analysis of TCP Windowsize Behavior*. Rapport technique ECE-TR-CCS-99-10-01, University of Massachusetts, Octobre 1999. (Cité en pages 14, 20, 21, 22 et 23.)

[Misra 2000] V. Misra, W. Gong et D. Towsley. *Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED*. In Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, SIGCOMM '00, pages 151–160, Stockholm, Sweden, 2000. (Cité en pages 14, 20, 22, 23, 24, 51, 75 et 141.)

[Morse 1976] A. S. Morse. *Ring models for delay-differential systems*. Automatica, vol. 12, pages 529–531, September 1976. (Cité en page 44.)

[Niculescu 2001] S.-I. Niculescu. Delay effects on stability a robust control approach. Lecture notes in Control and Information Sciences. Springer, 2001. (Cité en pages 25, 27 et 47.)

[Orebaugh 2007] A. Orebaugh, G. Ramirez, J. Burke, L. Pesce, J. Wright et G. Morris. Wireshark & ethereal network protocol analyzer toolkit. Syngress Publishing, Rockland, MA, USA, 2007. (Cité en pages 38, 113 et 151.)

[Ott 1999] T.J. Ott, T.V. Lakshman et L.H. Wong. *SRED : stabilized RED*. In Proceedings of the 18^{th} Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM '99, volume 3, pages 1346–1355, New York, NY, USA, March 1999. (Cité en page 13.)

[Owezarski 2004] P. Owezarski et N. Larrieu. *A trace based method for realistic simulation*. In IEEE International Conference on Communications, volume 4, pages 2236–2239, Paris, France, June 2004. (Cité en pages 35, 109, 110, 111, 112, 114 et 122.)

[Owezarski 2008] P. Owezarski, P. Berthou, Y. Labit et D. Gauchard. *LaasNetExp : a generic polymorphic platform for network emulation and experiments*. In Proceedings of the 4th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities (TRIDENTCOM), pages 1–9, Innsbruck, Austria, 2008. (Cité en pages 38 et 113.)

[Padhye 1998] J. Padhye, V. Firoiu, D. Towsley et J. Kurose. *Modeling TCP throughput : a simple model and its empirical validation*. In Proceedings of the ACM conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, SIGCOMM'98, volume 28, pages 303–314, Vancouver, British Columbia, Canada, October 1998. (Cité en page 19.)

[Park 1996] K. Park, G. Kim et M. Crovella. *On the relationship between file sizes, transport protocols, and self-similar network traffic*. In Proceedings of the International Conference on Network Protocols, pages 171–180, October 1996. (Cité en page 111.)

[Patcha 2007] A. Patcha et J.-M. Park. *An overview of anomaly detection techniques : Existing solutions and latest technological trends*. Computer Networks, vol. 51, pages 3448–3470, August 2007. (Cité en page 18.)

[Pisano 2008] A. Pisano, A. Davila, L. Fridman et E. Usai. *Cascade Control of PM DC Drives via Second Order Sliding Mode Technique*. IEEE Transactions on Industrial Electronics, vol. 55, no. 11, pages 3846–3854, 2008. (Cité en page 34.)

[Rahmé 2009a] S. Rahmé, Y. Labit et F. Gouaisbaut. *Sliding Mode Observer for Anomaly Detection in TCP/AQM Networks.* In Proceedings of the Second International Conference on Communication Theory, Reliability, and Quality of Service, CTRQ'09, pages 113–118, Colmar, France, 2009. IEEE Computer Society. (Cité en pages 76 et 77.)

[Rahmé 2009b] S. Rahmé, Y. Labit et F. Gouaisbaut. *An unknown input sliding observer for anomaly detection in TCP/IP networks.* In International Conference on Ultra Modern Telecommunications Workshops, ICUMT'9, pages 1–7, St.Petersburg, Russia, October 2009. (Cité en pages 76 et 79.)

[Rahmé 2010] S. Rahmé, Y. Labit, F. Gouaisbaut et T. Floquet. *Second order sliding mode observer for anomaly detection in TCP networks : From theory to practice.* In Proceedings of the 49^{th} IEEE Conference on Decision and Control, CDC'10, pages 5120–5125, Atlanta, Georgia, USA, December 2010. (Cité en pages 83, 105 et 109.)

[Rahmé 2011] S. Rahmé, Y. Labit, F. Gouaisbaut et T. Floquet. *Sliding modes for anomaly observation in TCP networks : from theory to practice.* IEEE Transactions on Control Systems Technology (TCST), soumis en mars 2011. (Cité en pages 83, 105 et 109.)

[Riachy 2008] Samer Riachy, Yuri Orlov, Thierry Floquet, Raul Santiesteban et Jean-Pierre Richard. *Second order sliding mode control of underactuated Mechanical systems I : Local stabilization with application to an inverted pendulum.* International Journal of Robust and Nonlinear Control, vol. 18, no. 4-5, pages 529–543, 2008. (Cité en pages 34, 69 et 70.)

[Richard 2003] J.-P. Richard. *Time Delay Systems : An overview of some recent advances and open problems.* Automatica, vol. 39, pages 1667–1694, 2003. (Cité en pages 25 et 44.)

[Ryu 2004] S. Ryu, C. Rump et C. Qiao. *Advances in Active Queue Management (AQM) Based TCP Congestion Control.* Telecommunication Systems, vol. 25, pages 317–351, March 2004. (Cité en page 13.)

[Sename 2001] O. Sename. *New trends in design of observers for time-delay systems.* Kybernetica, vol. 37, no. 4, pages 427 – 458, 2001. (Cité en page 44.)

[Slotine 1986] J.-J.E. Slotine, J.K. Hedrick et E.A. Misawa. *On Sliding Observers for Nonlinear Systems.* In American Control Conference (ACC), pages 1794–1800, Seattle, WA, USA, June 1986. (Cité en page 68.)

[Smirnov 2002] G.V. Smirnov. Introduction to the theory of differential inclusions, volume 41. American Mathematical Society, 2002. (Cité en page 66.)

[Sorenson 1985] H. W. Sorenson. Kalman filtering : Theory and application. IEEE PRESS, 1985. (Cité en page 28.)

[Specht 2004] S. M. Specht et R. Lee. *Distributed Denial of Service : Taxonomies of Attacks, Tools, and Countermeasures.* In Proceedings of the 17^{th} International Conference on Parallel and Distributed Computing Systems, PDCS'04, pages 543–550, Cambridge, Massachusetts, USA, November 2004. (Cité en page 15.)

[Spurgeon 2008] S.K. Spurgeon. *Sliding mode observers : a survey.* International Journal of Systems Science, vol. 39, pages 751–764, August 2008. (Cité en pages 32, 69 et 77.)

[Stevens 1994] W. R. Stevens. Tcp/ip illustrated : The protocols. Addison-Wesley Publishing Company, 1994. (Cité en pages 5, 7 et 8.)

[Tanenbaum 1994] A. Tanenbaum. Computer networks. Prentice-Hall, USA, 1994. Traduction française : Réseaux : Architectures, protocoles, applications. InterEditions. Paris, France. (Cité en pages 5 et 12.)

[Utkin 1992] V.I. Utkin. Sliding modes in control and optimization. Springer-Verlag, Berlin, Germany, 1992. (Cité en pages 33, 34, 66, 77 et 78.)

[Varga 2009] A. Varga. *OMNeT++, User Manual*, 2009. www.omnetpp.org/documentation. (Cité en page 91.)

[Wang 2001] Z. Wang, B. Huang et H. Unbehauen. *Robust H_∞ observer design of linear time-delay systems with parametric uncertainty.* Systems and Control Letters, vol. 42, no. 4, pages 303–312, 2001. (Cité en page 30.)

[Wang 2003] D. Wang et C. V. Hollot. *Robust analysis and design of controllers for a single TCP flow.* In IEEE International Conference on Communication Technology, ICCT'03, volume 1, pages 276–280, April 2003. (Cité en page 14.)

[Wierman 2003] A. Wierman et T. Osogami. *A unified framework for modeling TCP-Vegas, TCP-SACK, and TCP-Reno.* In Proceedings of the 11^{th} IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems, MASCOTS'03, pages 269–278, October 2003. (Cité en page 20.)

[Witrant 2005] E. Witrant, D. Georges, C. de Wit et O. Sename. *Stabilisation of Network Controlled System with a Predictive Approach.* In 1st Workshop on Networked Control System and Fault Tolerant Control, Octobre 2005. (Cité en page 14.)

[Yang 1997] H. Yang et M. Saif. *State observation, failure detection and isolation (FDI) in bilinear systems.* International Journal of Control, vol. 67, August 1997. (Cité en page 29.)

[Yang 1998] H. Yang et M. Saif. *Observer design and fault diagnosis for state-retarded dynamical systems.* Automatica, vol. 34, no. 2, pages 217–227, 1998. (Cité en page 32.)

[Ye 2000] N. Ye. *A markov chain model of temporal behavior for anomaly detection.* In Proceedings of the IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop, pages 171–174, 2000. (Cité en page 19.)

[You 2004] F. You, Z. Tian et S. Shi. *Actuator fault diagnosis for a class of time-delay systems.* In Proceedings of the 5^{th} World Congress on Intelligent Control and Automation, WCICA'04, volume 2, pages 1798–1802, June 2004. (Cité en page 32.)

[Zhong 2003] M. Zhong, S.X. Ding, J. Lam et C. Zhang. *Fault detection filter design for LTI system with time delays.* In Proceedings of the 42^{nd} IEEE Conference on Decision and Control, CDC'03, volume 2, pages 1467–1472, December 2003. (Cité en page 32.)

AUTHOR : Sandy Rahmé
TITLE : Anomaly detection and estimation in a communication network
SUPERVISORS : Frédéric Gouaisbaut, Yann Labit

Summary :
The supervision domain particularly the *anomaly detection* represents an important aspect of guaranteeing a *Quality of Service* to communication networks. A wide variety of disruptions designated as anomalies are often related to physical or technical problems such as power or file server failures, abrupt changes caused by legitimate traffic such as network congestion or flash crowds, and risky illegitimate behavior such as *Denial-of-Service* and *Distributed Denial of Service* (DoS/DDoS) attacks.

We address the problem of anomalies detection and reconstruction in TCP/IP model based on control theory techniques. These anomalies are considered as fault signals in the mathematical model adopted for representing TCP/IP dynamics. For faults detection and according to our knowledge of the faults variations, the observers may be classified into known or unknown input observers. Our first contribution in terms of conceiving known input observers is limited to polynomial forms able to cover a wide range of anomalies. The anomaly and its derivatives are reconstructed by *Luenberger observers* after introducing them in the state space of the system. The construction of these latter observers is limited in terms of specific anomaly profiles and constrained by the polynomial degree associated to the anomaly. Therefore, another detection approach dealing with completely unknown anomalies is proposed. The *sliding modes* of first and higher orders are investigated to guarantee finite time convergence and robustness against parametric uncertainties and faults.

Our proposals have been studied analytically by validating via Matlab/Simulink and the Network Simulator NS-2. Furthermore, in the context of NS-2, these approaches are integrated into a module for replaying traffic traces in order to test them on a TCP traffic captured in real environment.

Keywords : TCP Protocol, Quality of Service, anomalies, time delayed systems, faults detection and reconstruction, observer theory, sliding modes, NS-2, traffic trace replay.
