Routage pair-à-pair pour la fiabilité des communications

Simon Delamare

-

Remerciements

Je tiens tout d'abord à exprimer ma reconnaissance au professeur Michel Riguidel, pour ses conseils avisés et pour m'avoir permis de réaliser cette thèse dans les meilleures conditions. Je remercie aussi Gwendal Le Grand, avec qui ces travaux ont débuté.

Mes remerciements vont également aux professeurs Damien Magoni, Dritan Nace, Béatrice Paillassa et Noëmie Simoni d'avoir eu l'amabilité de prendre part au jury de cette thèse.

Je tiens également à remercier les membres du département Informatique et Réseau que j'ai côtoyés ces dernières années, en particulier Claude Chaudet et Artur Hecker pour l'aide précieuse qu'ils m'ont toujours apportée. Je tiens aussi à saluer mes collègues doctorants ou jeunes docteurs avec qui j'ai partagé mon bureau : Alpha, Amy, Bin, David, Hany et Samy. J'ai aussi une pensée pour les personnels et les étudiants de l'université Paris 12 et de l'IUT de Villetaneuse que j'ai rencontrés au cours de mes activités d'enseignement.

Toute ma reconnaissance et mon affection vont à mes amis et à ma famille, en particulier à mon cousin Clément, pour m'avoir «montré la voie», ainsi qu'à ma chère maman, qui m'a toujours supporté et encouragé. Je remercie enfin Cécile, pour son soutien au quotidien et pour être si merveilleuse.

Résumé

L'utilisation généralisée des réseaux IP pour les télécommunications entraîne de nouvelles préoccupations quant à leurs fiabilités. L'apparition d'incidents affectant le réseau et perturbant l'acheminement des communications aux utilisateurs ne peut en effet pas toujours être évitée. Des mécanismes de rétablissement réseau sont alors utilisés pour rediriger les communications vers une partie du réseau non affectée par l'incident. L'enjeu de cette opération est qu'elle soit réalisée suffisamment rapidement pour maintenir la bonne délivrance des services réseau aux utilisateurs.

Dans cette thèse, nous avons étudié un nouveau type de mécanisme de rétablissement réseau basé sur le routage pair-à-pair. Ce système utilise un réseau virtuel formé par un ensemble de noeuds participants qui réalisent des opérations de routage. Il a pour avantage de pouvoir être utilisé pour le rétablissement de n'importe quel type de communication IP et de ne pas dépendre de l'infrastructure du réseau. De plus, il permet la protection de bout en bout d'une communication. Ce n'est pas le cas des mécanismes de rétablissement usuels, tels que les protocoles de routage, qui opèrent uniquement à l'intérieur des différents réseaux traversés lors de l'acheminement d'une communication, et ne coopèrent pas entre eux. Enfin, ce mécanisme est déployé par les utilisateurs des réseaux, ce qui permet d'adapter son fonctionnement à leurs besoins de fiabilité spécifiques à chacune de leurs communications.

Ce document est consacré à l'étude et à la conception d'un tel système, dont le but est d'améliorer la fiabilité des communications des utilisateurs lorsqu'elles sont affectées par un incident. Nous étudions tout d'abord la détection d'incident, qui est une étape préalable à toute opération de rétablissement. Ensuite, nous présentons le fonctionnement de notre système de rétablissement réseau basé sur le routage pair-à-pair, ainsi que son implémentation. Nous évaluons son efficacité lors d'expérimentations réalisées sur différentes plateformes de test ainsi que par simulation, et le comparons à d'autres mécanismes de rétablissement.

Nous montrons que notre système permet le rétablissement rapide des communications si les besoins de l'utilisateur le justifient. De plus, les ressources nécessaires à son fonctionnement sont modérées et en rapport avec ces besoins. Nous pensons que notre système permet une amélioration significative de la fiabilité des communications lorsqu'un incident se déclare dans le réseau, en particulier si les mécanismes de rétablissement déployés par les réseaux des opérateurs ne permettent pas d'apporter le niveau de fiabilité recherché par l'utilisateur.

Mots clefs : Fiabilité, Pair-à-pair, Rétablissement, Routage.

Peer-to-peer routing for communications dependability

Simon Delamare

Abstract : The extended use of IP networks for telecommunication leads to new dependability concerns. Incidents, which hit the network and disturb communications delivery to users, cannot be always prevented. Network recovery mechanisms are used to redirect communications to a non-failing part of the network. This process goal is to be fast enough to keep good delivery of network services to users.

In this thesis, we studied a new kind of recovery mechanism based on peer-to-peer routing (also called overlay routing). This system uses a virtual network, made of a network nodes set performing routing operations. Its advantages are that it can be used to recover any kind of IP communications, and to not depend on the network infrastructure. It also allows an end-to-end protection of a communication. This is contrary to usual recovery mechanisms, such routing protocols, which only operate inside the various networks used to forward a communication and do not cooperate between themselves. Finally, this mechanism is deployed by network users, and thus can be adapted to the dependability needs for each of their communications.

This document is dedicated to the study and the conception of such a system, which main goal is to enhance the users' communications dependability when affected by an incident. We first study the incident detection, which is the first step of any recovery operation. We then introduce our peer-to-peer based network recovery system, and its implementation. We evaluate its efficiency with experiments performed in various test beds and simulations, and we compare it to other recovery mechanisms.

We show that our system allows fast communication recovery if users need it. Moreover, its resources consumption are moderated and related to users needs. We think that our system can significantly improve communications dependability when incidents hit the network, particularly if recovery mechanisms deployed by network operators cannot bring the dependability level pursued by a user.

Key words : Dependability, Overlay, Peer-to-peer, Recovery, Reliability, Routing.

Table des matières

Table des figures

Liste des tableaux

Liste des sigles

ADSL Asymmetric Digital Subscriber Line
AODV Ad-hoc On-demand Distance Vector
AS Autonomous System
BFD Bidirectional Forwarding Detection
BGP Border Gateway Protocol
CDN Content Delivery Network
DDoS Distributed Denial of Service
DoS Denial of Service
DHT Distributed Hash Table
DNS Domain Name System
DSR Dynamic Source Routing
EGP Exterior Gateway Protocol
EIGRP Enhanced Interior Gateway Routing Protocol
FSR Fisheye State Routing
FTP File Transfer Protocol
HTTP Hypertext Transfer Protocol
I3 Internet Indirection Infrastructure
ICMP Internet Control Message Protocol
IETF Internet Engineering Task Force
IGP Interior Gateway Protocol
IGRP Interior Gateway Routing Protocol
IP Internet Protocol
IPSec Internet Protocol Security
IS-IS Intermediate System to Intermediate System
ISO International Organization for Standardization
LSP Label Switched Path
LSR Label Switched Router
MAC Media Access Control
MANET Mobile Ad-hoc NETworks

MPLS Multiprotocol Label Switching

MTTF Mean Time To Failure

MTTR Mean Time To Repair

MTU Maximum Transmission Unit

NAT Network Address Translation

OLSR Optimized Link State Routing protocol

OSI Open Systems Interconnection

OSPF Open Shortest Path First

P2P Peer-To-Peer

PPRR Path Probing Relay Routing

RIP Routing Information Protocol

RON Resilient Overlay Network

RTO Retransmission Time Out

RTT Round Trip Time

SLA Service-Level Agreements

SMTP Simple Mail Transfer Protocol

TCP Transmission Control Protocol

TMD Durée maximale de présence d'un incident non détecté tolérée

TMIT Temps Maximum d'Interruption Toléré

UDP User Datagram Protocol

WDM Wavelength-Division mMltiplexing

Chapitre 1

Introduction

1.1 Contexte et motivation

Au cours de ces dernières années, l'utilisation des réseaux IP s'est banalisée, Internet est aujourd'hui utilisé par plus d'un milliard de personnes[31], ainsi que par de nombreuses administrations publiques et entreprises. Parallèlement, les progrès techniques des infrastructures ont permis de délivrer de nouveaux services, tels que le transport de contenus multimédias, aux utilisateurs de ces réseaux. L'acheminement de ces services a nécessité la création de nouveaux protocoles de communication afin de satisfaire leurs nouveaux besoins, tels que la qualité de service. De plus, la gestion de ces services et de leurs utilisateurs a entraîné une complexification des logiciels déployés sur les serveurs des réseaux délivrant ces services. On peut donc constater la sophistication croissante des réseaux IP et d'Internet, qui ne va pas sans une difficulté accrue de gestion.

L'utilisation d'Internet s'est répandue à travers le monde et son utilisation s'est intensifiée, pour faire pleinement partie de la vie courante. Ce phénomène entraîne une dépendance de plus en plus forte aux services fournis par les réseaux. Ainsi, les conséquences de la non-disponibilité de l'accès à certains services Internet sont importantes. De plus, les attaques malicieuses visant à empêcher la délivrance des services sont de plus en plus communes. On a par exemple vu des attaques par déni de service[35] être menées contre les institutions et les grandes entreprises d'un pays suite à un conflit politique. Il est difficile de prévenir et déjouer ces attaques, car elles sont de plus en plus élaborées et exploitent la complexité du réseau. Elles affectent aussi un grand nombre de personnes qui utilisent aujourd'hui massivement le réseau pour accéder aux services courants.

De plus, l'utilisation des réseaux IP est parfois critique, c'est-à-dire qu'elle met en jeu le bien-être des personnes ou des biens. C'est notamment le cas lorsque le réseau est utilisé pour des communications à caractère médical, policier ou de secours. Il est dans ce cas inacceptable qu'un incident survenant dans le réseau perturbe les communications de manière prolongée.

Ces constatations mettent en évidence l'existence d'un fort besoin de fiabilité pour les communications réseau des utilisateurs. Nous considérons que la complexité croissante des communications réseau empêche la prévention systématique des incidents. Ainsi, il y aura toujours des incidents affectant l'accès aux services par les utilisateurs. Lorsqu'un incident apparaît dans le réseau et que celui-ci perturbe les communications, il est par conséquent nécessaire de mettre en place des mécanismes qui permettent de les rétablir. Ceci permet de maintenir la livraison des services réseau aux utilisateurs. C'est le rôle des mécanismes de rétablissement réseau, qui, lors de l'apparition d'un incident, modifient les chemins utilisés dans le réseau pour acheminer les communications de manière à ce que la partie du réseau affectée par l'incident soit contournée. C'est par exemple le protocole de routage qui assume cette fonction dans les réseaux IP aujourd'hui.

Nous pensons cependant que les protocoles de routage classiques, tels que Open Shortest Path First[62] (OSPF), ne permettent pas toujours d'assurer un rétablissement rapide et efficace en cas d'incident. En effet, le rôle de ces protocoles est tout d'abord de renseigner les tables de routage afin d'assurer la connectivité entre les noeuds d'un réseau en état de fonctionnement normal. Cependant, ces protocoles ne sont pas spécifiquement conçus pour détecter et réagir rapidement en cas d'incident dans le réseau.

D'autres mécanismes de rétablissement, tels que ceux utilisés dans les réseaux Multi Protocol Label Switching[84] (MPLS), existent. Ceux-ci sont parfois spécifiquement conçus pour réagir à un incident. Cependant, nous pensons qu'ils ne tiennent pas suffisamment compte des besoins de fiabilité des communications des utilisateurs. En effet, ils ne prennent pas en considération les besoins spécifiques pour l'acheminement d'une communication, qui sont liés au service délivré. De plus, ils ne tiennent pas compte de la criticité, pour l'utilisateur, de la bonne livraison de ce service. Par conséquent, les mécanismes de rétablissement existants ne peuvent satisfaire pleinement les divers besoins

des utilisateurs pour la fiabilité de leurs communications.

Enfin, les différents mécanismes de rétablissement existant sont déployés par les opérateurs des réseaux, et les utilisateurs n'ont aucune garantie de leur bon fonctionnement. Sur Internet, les communications entre utilisateurs sont acheminées par plusieurs réseaux. Le bon acheminement des communications va ainsi dépendre des différents mécanismes de rétablissement déployés par les opérateurs des réseaux traversés. La fiabilité des communications ne dépend donc pas uniquement, par exemple, du réseau du fournisseur d'accès à Internet d'un utilisateur. De plus, lorsque des incidents affectent plusieurs de ces réseaux, les différents mécanismes de rétablissement impliqués ne coopèrent généralement pas pour rétablir au mieux les communications. Par conséquent, il est nécessaire de proposer aux utilisateurs un autre système de rétablissement réseau.

1.2 Problème et solution proposée

Les différentes constatations évoquées ci-dessus mettent en avant la nécessité de concevoir un autre mécanisme pour la fiabilité des communications des utilisateurs. Ce système se doit d'être réactif et de permettre le rétablissement des communications lors de l'apparition d'un incident. De plus, il ne doit pas dépendre de l'infrastructure utilisée pour acheminer les communications. Par exemple, ce mécanisme doit pouvoir être déployé sur tout type de réseau IP, quel que soit le support physique assurant la communication. De la même manière, le système doit pouvoir être utilisé pour tout type de communication, quel que soit le service délivré. Par exemple, il doit aussi bien pouvoir être utilisé pour la fiabilité des communications d'une vidéoconférence que pour la fiabilité des communications Web. Enfin, le mécanisme doit être adapté aux besoins réels en terme de fiabilité exprimés par l'utilisateur pour ses communications. Il doit pour cela être configurable afin de garantir au mieux ces besoins, tout en consommant le minimum de ressources.

Dans cette thèse, nous nous sommes intéressés à l'utilisation d'un système de routage pair-à-pair (P2P) dédié au rétablissement réseau. Les réseaux P2P permettent le déploiement, par les utilisateurs, de nouveaux services sans avoir à modifier l'infrastructure réseau sous-jacente. Pour cela, les noeuds participants à ce réseau forment un réseau virtuel qui déploie un service de la même manière que le ferait un réseau normal. Les réseaux P2P permettent par exemple le déploiement d'un service multicast[7] ou de partage de fichier[13]. Les systèmes de routage P2P tels que Resilient Overlay Network[1] (RON) sont composés de noeuds routeurs qui modifient l'acheminement des communications en les faisant transiter dans le réseau P2P. Ce système permet ainsi la redirection d'une communication jusqu'à sa destination lorsque le réseau IP normalement utilisé est défaillant.

Nous pensons que l'utilisation du routage P2P permet de satisfaire les exigences exprimées plus haut. En effet, cette solution est réactive, puisque les communications sont redirigées en cas d'apparition d'un incident pour permettre la continuité de la délivrance du service. De plus, comme nous l'avons vu, le routage P2P est indépendant de l'infrastructure. En effet, les noeuds participants à ce système sont les machines des utilisateurs dont la seule contrainte est qu'elles soient reliées au réseau. Enfin, le routage P2P est indépendant du contenu à protéger. En effet, dans le modèle que nous envisageons d'utiliser, le routage se substitue au routage IP, mais est indépendant des données acheminées. Il peut donc être aussi bien être utilisé pour protéger une communication Web qu'une diffusion vidéo.

Ainsi, nous envisageons d'utiliser le routage P2P pour la fiabilité des communications des utilisateurs. Nous envisageons l'utilisation de cette technique par un ensemble limité de noeuds qui collaborent afin de protéger leurs communications les plus sensibles. Par exemple, une entreprise pourrait utiliser ce système en formant un réseau P2P avec des machines présentes sur ses différents sites géographiques et reliés à Internet. Il serait alors possible pour l'entreprise d'accéder de manière fiable à

ses services distants les plus sensibles.

1.3 Contributions du document

Ce document présente les différents travaux de recherche qui ont été menés pour concevoir et mesurer l'efficacité d'un système de routage P2P pour la fiabilité des communications des utilisateurs. Trois sujets principaux ont ainsi été étudiés.

Tout d'abord, afin de réagir lors de l'apparition d'un incident, il est nécessaire qu'il soit détecté. Puisque notre système est déployé par les utilisateurs, ce sont eux qui ont la charge de détecter les incidents affectant leurs communications. Afin d'améliorer la fiabilité d'une communication, il est nécessaire de disposer de mécanismes de détection d'incident précis, en mesure de détecter un incident en un temps déterminé et qui consomme le minimum de ressources. Nous étudions par conséquent ces mécanismes et proposons d'utiliser l'envoi de messages sondes pour détecter un incident. À l'aide d'une simulation de leurs comportements, nous montrons comment les utiliser afin qu'ils fonctionnent au mieux, en prenant en considération les attentes des utilisateurs pour la détection. Nous mesurons qu'il est possible de détecter un incident de manière fiable, même lorsque l'utilisateur a besoin de détecter cet incident en moins d'une seconde. Les résultats de ces travaux peuvent être utilisés ensuite par notre système de routage P2P.

Nous présentons ensuite le système de routage P2P qui répond à nos attentes. Notre système déploie des mécanismes pour protéger une communication d'un incident, en fonction d'un temps maximum d'interruption de sa livraison toléré par l'utilisateur. Lorsqu'un incident affecte cette communication, une fois celui-ci détecté, notre système réachemine la communication par un des chemins de secours qui transite par le réseau P2P. Diverses techniques, telles que le routage par la source et le double acheminement, sont utilisées pour assurer un rétablissement plus rapide. Nous proposons une implémentation de notre système. Nous réalisons ensuite une expérimentation afin de mesurer ses performances. Les temps de rétablissement permis par notre système peuvent être inférieurs à une seconde. De plus, nous montrons que sa consommation de ressource dans le réseau est raisonnable, et est en rapport avec la fiabilité demandée par l'utilisateur.

Nous étudions enfin la portée du routage P2P, et la comparons aux autres mécanismes de routage existant. La portée est la capacité d'un système de routage à rétablir une communication affectée par un incident dans le réseau. En effet, en fonction du réseau IP, du nombre de noeuds participant au routage et de l'étendue de l'incident, un système de routage n'est pas toujours en mesure de rétablir une communication. Afin de mesurer cette propriété, nous réalisons une simulation des mécanismes de routage P2P dans un réseau interconnecté soumis à plusieurs scénarios d'incident. Nous comparons notre système au système RON, ainsi qu'aux protocoles de routage habituellement utilisés dans les réseaux IP. Nous montrons que les capacités de rétablissement de notre système sont proches de celles du système RON et du routage IP. De plus, nous mesurons qu'il n'est pas nécessaire qu'un grand nombre de noeuds participent au réseau P2P. Ces travaux permettent de confirmer l'utilité du routage P2P pour améliorer la fiabilité des communications sur Internet.

Cette thèse est donc consacrée à l'étude d'une solution pour la fiabilité des communications des utilisateurs. Nous proposons un système de rétablissement réseau complet pour atteindre ce but. Ce système peut être utilisé aujourd'hui, sur tout type de réseau IP. Il prend en compte les besoins de fiabilité de l'utilisateur pour fonctionner. Les performances de ce système sont mesurées, et il est comparé avec d'autres mécanismes de rétablissement. Nous pensons qu'il permet une amélioration significative de la fiabilité des communications lorsqu'un incident se déclare dans le réseau, en particulier si les mécanismes de rétablissement déployés par les réseaux des opérateurs ne permettent pas

d'apporter le niveau de fiabilité recherché par l'utilisateur.

Il nous a paru nécessaire de nous intéresser à l'amélioration de la fiabilité dans les réseaux IP. Ce domaine a été moins étudié ces dernières années, en particulier pour ce qui concerne les communications de bout en bout. Pourtant, les besoins de fiabilité n'ont jamais été aussi grands. Les nouvelles utilisations des réseaux, parfois critiques, justifient ce besoin. Nous pensons que nos travaux pourront être utiles à ceux qui recherchent une meilleure fiabilité de leurs communications.

Une des originalités de ces travaux est qu'ils sont orientés vers les utilisateurs des réseaux. Trop souvent, les problèmes tels que la fiabilité ne sont abordés que du point de vue du réseau et de son architecture. Le développement des réseaux P2P ces dernières années a montré que les utilisateurs pouvaient et voulaient se réapproprier le réseau. C'est en particulier vrai lorsqu'il existe un besoin qui n'est pas satisfait par les opérateurs des réseaux. Nous pensons que l'utilisation d'un système de rétablissement dans un réseau P2P permet de répondre en partie au besoin de fiabilité des utilisateurs.

Enfin, nos travaux contribuent au domaine des réseaux P2P. Ces réseaux ont été très étudiés ces dernières années, et de nombreux systèmes qui s'appuient sur ce type de réseau ont vu le jour. Nos travaux portent sur la mise au point d'un tel système. Nous pensons qu'ils pourront être réutilisés par d'autres, qui ne seraient pas nécessairement liés au problème de la fiabilité des communications.

1.4 Organisation du document

Les différents chapitres de ce document sont les suivants : le deuxième chapitre sera consacré à la description du contexte de ces travaux. Le chapitre suivant sera consacré à l'étude des mécanismes de détection d'incident par envoi de messages sondes. Notre système de routage P2P sera présenté et évalué dans le quatrième chapitre. Le cinquième chapitre sera consacré à l'étude de la portée des systèmes de routage P2P. Enfin, nous conclurons nos travaux dans le sixième chapitre.

Chapitre 2

Contexte des travaux

Afin de mieux appréhender les travaux qui seront présentés dans les chapitres suivants, nous allons introduire ici le contexte dans lequel ils s'appliquent. Pour cela, nous allons tout d'abord discuter des grandes définitions et des principes généraux d'utilisation et de fonctionnement des réseaux IP. Nous aborderons ensuite la problématique de fiabilité des communications, en nous intéressant aux incidents qui affectent les réseaux. Puis, nous étudierons les mécanismes de rétablissement réseau, dont le but est de rétablir la délivrance des communications affectées par les incidents. Enfin, nous nous intéresserons au routage P2P, qui est la solution étudiée ici pour fiabiliser les communications affectées par ces incidents.

2.1 Applications et services dans les réseaux

Nous allons expliquer dans cette partie ce qu'est un service délivré par le réseau et quelles contraintes existent pour sa délivrance.

2.1.1 Définitions

Nous allons aborder ici quelques termes qui seront utilisés tout au long de ce document. Les utilisateurs des réseaux sont des entités, telles que des personnes ou des organisations, qui sont connectées au réseau à l'aide d'une machine. Une communication entre utilisateurs est un échange d'information sous forme électronique et informatique. Pour permettre la mise relation des utilisateurs, le réseau achemine ces communications et les délivre aux utilisateurs. Le flux d'information qui transite alors dans le réseau est appelé trafic.

Les communications entre les utilisateurs ont pour but de rendre un certain service, qui répond à une attente de ces utilisateurs. Les applications, déployées sur les machines, ont pour rôle de « traduire » les informations contenues dans une communication en informations utilisables par les utilisateurs pour se voir délivrer le service. Les utilisateurs qui communiquent ne sont pas toujours égaux : il existe des fournisseurs de service qui sont des utilisateurs des réseaux dont le rôle est de délivrer un service à d'autres utilisateurs ou clients. Pour cela, le fournisseur de service dispose d'une ou plusieurs machines dédiées à la délivrance du service appelées serveurs.

2.1.2 Les services et leurs besoins

Aujourd'hui et de plus en plus, les réseaux IP permettent la délivrance d'une grande variété de services, tels que la téléphonie, la télévision, la vidéoconférence. L'acheminement de ces services par le réseau amène des contraintes variées. Ces contraintes sont souvent liées aux prérequis suivants :

- Le débit minimum requis pour acheminer le trafic de la communication
- Le délai de livraison maximum avec lequel acheminer le trafic pour que la qualité de la communication soit satisfaisante
- La sensibilité de l'application à une interruption de la communication (ou à une augmentation du délai de livraison). Ceci est en général lié à la durée de vie des informations contenues dans la communication, c'est-à-dire le temps maximum pour qu'une information soit utile pour rendre le service une fois celle-ci produite..

La table 2.1 [42, 103], présente succinctement certains de ces services et les contraintes liées à leur livraison. Nous avons qualifié de « faible » le débit minimum requis (« Débit ») lorsque celui-ci est inférieur à 500 kbit/s, « fort » lorsqu'il est supérieur à 500 kbit/s. Nous avons qualifié de « court » le délai minimum de livraison (« Délai de livraison ») lorsque celui-ci est inférieur à 200 ms, « moyen » lorsqu'il est compris entre 200 ms et 2 secondes, et « long » au-delà. Nous avons qualifié de « faible »

TAB. 2.1: Exemples de services délivrés dans les réseaux et leurs besoins

Service	**Débit**	**Délai de livraison**	**Interruption**
Navigation Internet	Faible	Moyen	Moyenne
Messagerie électronique	Faible	Long	Faible
Jeux interactifs	Faible	Court	Forte
Diffusion audio	Faible	Moyen	Moyenne
Diffusion vidéo	Fort	Moyen	Moyenne
Téléphonie par Internet	Faible	Court	Forte
Téléphonie vidéo par Internet	Fort	Court	Forte
Partage de fichier	Fort	Long	Faible
Informations à la demande	Faible	Long	Faible

la sensibilité de l'application à une interruption de la communication (« Interruption ») lorsqu'une interruption de plusieurs secondes de la communication peut-être tolérée, « moyenne » lorsqu'une interruption comprise entre 200 ms et quelques secondes peut être tolérée et « forte » lorsqu'une interruption d'au maximum 200 ms peut être tolérée.

On peut constater que les contraintes pour la livraison des différents services, en particulier celles liées à la sensibilité de l'application à l'interruption de la communication, sont très variées. On peut donc déduire que la nature d'un service doit être prise en compte lors de la mise en place les mécanismes qui assureront sa bonne délivrance à l'utilisateur.

2.1.3 Besoins des utilisateurs et criticité d'un service

Différents utilisateurs n'utilisent pas un même service de la même façon. En effet, en fonction de la criticité de l'utilisation du service, c'est-à-dire de l'importance pour l'utilisateur à ce que ce service lui soit correctement délivré, on peut déduire quatre classes d'utilisation[99] d'un service :

- Services critiques pour la sécurité des personnes : Ce groupe concerne les services utilisés dans le milieu médical ou policier, par exemple. Pour des raisons de sûreté, ces services doivent absolument être délivrés. C'est particulièrement le cas lors des accidents majeurs provoquant des conditions de communication difficiles (catastrophes naturelles, etc.).
- Services critiques pour raisons économiques : Ce groupe concerne les services qui entraîneraient d'importantes pertes financières s'ils étaient interrompus.
- Services de confort : Ce groupe concerne des services qui sont fournis aux utilisateurs de manière fiable. L'interruption exceptionnelle de ces services peut être tolérée si elle n'est pas trop fréquente.
- Services de base : Ce groupe concerne les services dont la fiabilité n'est pas une préoccupation centrale.

On voit que la criticité d'un service est très variable et par conséquent, les conséquences de l'interruption de la délivrance d'un service le sont aussi. On peut en déduire que la criticité d'un service est à prendre en considération avant de mettre en place les mécanismes qui assureront sa bonne délivrance à l'utilisateur.

2.1.4 Les Service-Level Agreements

Nous l'avons vu, la fiabilité du réseau peut être essentielle pour certains utilisateurs. Les Service-Level Agreements[104] (SLA) sont un contrat passé entre un utilisateur et son prestataire de service

qui définit la qualité avec laquelle le prestataire est tenu de délivrer le service. Par exemple, le prestataire peut être engagé à fournir une certaine qualité de service, ou à assurer une certaine garantie de fonctionnement pour l'accès à ce service, avec un temps maximum d'interruption toléré limité. L'établissement de SLA a un coût pour l'utilisateur, mais ce dernier obtient des dédommagements en cas de non-respect du SLA par le prestataire.

Ainsi, des SLA peuvent être utilisés pour définir la fiabilité avec laquelle un service doit être délivré à son utilisateur. Dans ce cas, la bonne délivrance du service est recherchée aussi bien par l'utilisateur, qui a souscrit aux SLA, que par le prestataire, qui est tenu par contrat de respecter ses engagements.

2.2 Les réseaux IP

Nous allons rappeler les principes généraux de fonctionnement des réseaux IP déployés aujourd'hui.

2.2.1 Les communications dans les réseaux IP

Internet est le réseau de télécommunication qui connait le plus de succès aujourd'hui. Le nombre de ses utilisateurs n'a cessé d'augmenter depuis sa création [31]. Une tendance appelée convergence vers les réseaux numériques[63] montre que les réseaux IP sont de plus en plus utilisés pour transporter des communications qui traditionnellement disposaient d'un réseau dédié. C'est le cas par exemple des communications téléphoniques ou télévisuelles qui sont désormais transportées par le réseau IP de certains fournisseurs d'accès à Internet.

2.2.2 La représentation en couche

Pour fonctionner, un réseau IP utilise différentes technologies ou différents protocoles de communication. Chacun de ces protocoles a un rôle précis pour permettre d'assurer les communications. On parle d'organisation en couche des réseaux, où chacune des couches correspond à un certain niveau d'abstraction, car chaque protocole est associé à une couche. Les couches les plus basses correspondent aux protocoles les plus dépendants du matériel et de l'infrastructure utilisés, tandis que les couches les plus hautes sont en relation avec le service rendu à l'utilisateur au travers de l'application. Au milieu se trouve le protocole IP, dont le rôle est de permettre les communications entre l'ensemble des équipements réseaux de la planète, quel que soit l'infrastructure matérielle utilisée ou le service rendu à l'utilisateur.

Différents organismes ont proposé une standardisation du nombre et du rôle de chaque couche. Par exemple, l'Organisation Internationale de Normalisation a proposé une représentation en couche appelée modèle Open Systems Interconnection (OSI)[32], tandis que l'Internet Engineering Task Force (IETF) promeut une représentation en couche appelée modèle TCP/IP[93]. Nous nous concentrerons sur cette dernière (figure 2.1), qui est la plus communément admise, en partie car elle est la plus simple.

La couche la plus basse est la couche physique. Elle a la charge de la transmission d'un flot de bits sur un média de transport physique, mais ne s'occupe pas de la façon dont sont organisés ces bits. La couche suivante est la couche liaison de donnée, qui assure un contrôle de la transmission par le lien physique sous-jacent. Le service apporté par cette couche à la couche supérieure est donc la détection et le contrôle des erreurs. La troisième couche est la couche réseau. C'est elle qui assure l'indépendance des communications envers la technologie physique utilisée ainsi que la technologie

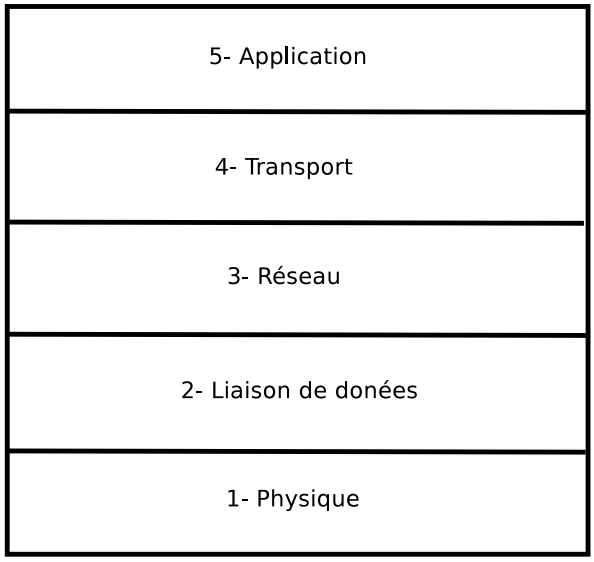


FIG. 2.1: Le modèle en couche TCP/IP

d'acheminement pour relier l'ensemble des équipements. La couche transport assure le transport et éventuellement le contrôle d'une communication entre deux points du réseau. Elle peut fournir des services de corrections en cas d'erreurs, de pertes ou de déséquencement de paquets. La couche la plus haute est la couche application. Elle définit la façon dont doit être organisée l'information de manière à permettre la compréhension des communications entre les applications.

En réalité, comme montre la figure 2.2, les réseaux utilisent aujourd'hui de nombreuses technologies différentes qui le plus souvent remplissent des fonctions qui appartiennent à différentes couches du modèle TCP/IP. Ceci est expliqué par le fait que le choix de l'utilisation de ces technologies s'est effectué au fur et à mesure de l'apparition de nouveaux besoins ainsi qu'en fonction des technologies utilisées par les équipements déjà déployés. Cependant, la représentation en couche d'une infrastructure reste très utilisée, car elle permet de visualiser les différentes technologies d'un réseau, même si le rôle de chaque technologie n'est pas aussi clairement délimité que dans les représentations en couche des modèles standards.

2.2.3 Architecture des réseaux

Avant d'aborder les protocoles utilisés dans les réseaux IP, nous allons introduire quelques notions liées à l'architecture de ces réseaux.

Les réseaux sont composés principalement de deux catégories d'équipement : les machines et les liens qui relient ces machines. Dans le contexte des réseaux, les machines sont couramment appelées noeuds. On peut séparer les noeuds en deux catégories : ceux dédiés à l'acheminement des communications et les autres, qui sont par exemple les machines des utilisateurs ou des machines dédiées à des taches d'administration du réseau. On dit que ces dernières sont situées « en bout de réseau », puisqu'elles ne sont pas utilisées par d'autres noeuds pour acheminer leur communication. Dans notre document, qui se concentre sur les réseaux IP, les noeuds dédiés à l'acheminement seront appelés routeurs, bien que la définition de routeur soit normalement plus stricte.

Les liens sont des médiums physiques utilisés pour le transport des communications. Ils sont généralement associés à un protocole de couche physique afin d'acheminer correctement les successions

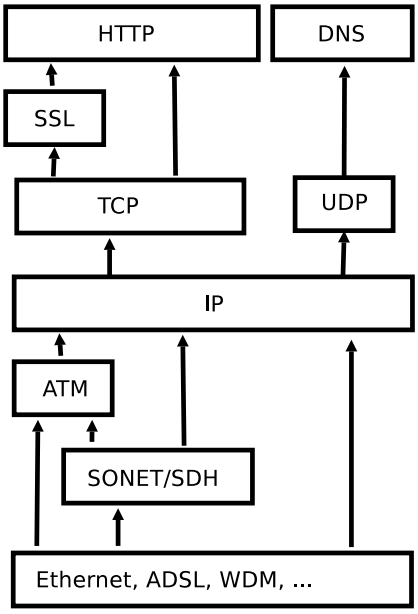


FIG. 2.2: Protocoles couramment rencontrés

de bits d'un noeud à un autre. La paire de fils torsadés est couramment employée pour le transport de données à des débits moyens[79]. Elle est utilisée par exemple dans les réseaux locaux ou par la technologie Asymmetric Digital Subscriber Line (ADSL)[3]. La fibre optique permet des débits plus importants [79] : la technique Wavelength Division Multiplexing (WDM) permet de faire transiter différentes longueurs d'onde sur une même fibre et d'obtenir un débit total de plusieurs térabits par seconde. Enfin, les réseaux basés sur un support radio (réseaux sans fil) se sont fortement développés grâce aux technologies de téléphonie cellulaire et aux technologies informatiques Wifi. À de rares exceptions près (par exemple, les réseaux Ad-Hoc), les liens sans fil sont utilisés pour relier l'utilisateur au premier routeur lui permettant d'accéder au réseau (ce sont des réseaux cellulaires). Les routeurs sont ensuite connectés entre eux par des liens filaires.

La taille d'un réseau est variable, selon son utilisation. Il peut ainsi être qualifié de local, pour un réseau domestique ou d'entreprise, ou encore de national, lorsqu'il s'étend sur un pays, ou international, lorsqu'il s'étend sur plusieurs pays. Le rôle des réseaux de grande taille est souvent d'interconnecter des réseaux de plus petite taille. Par exemple, les réseaux de coeur ont pour rôle de connecter de plus petits réseaux pour permettre les communications entre utilisateurs de ces petits réseaux sur de très longues distances. Internet est ainsi une interconnexion de réseaux à l'échelle mondiale. L'entité qui a la responsabilité de l'administration d'un réseau est appelée opérateur d'un réseau. Dans le contexte de réseaux interconnectés, un réseau administré par un même opérateur est souvent appelé système autonome (ou AS pour Autonomous System). Un exemple d'opérateur de réseau est le fournisseur d'accès à Internet qui connecte les utilisateurs, via son réseau, à Internet.

2.2.4 Protocole de couche liaison de données

Nous allons aborder ici deux des principaux protocoles de couche liaison de données. Le rôle de ces protocoles est d'acheminer les paquets entre deux noeuds reliés par un lien. Pour cela, les paquets sont incorporés dans des trames qui sont des blocs composées d'une succession de bits.

Ethernet

Ethernet[30] est le protocole le plus répandu pour le transport des paquets dans les réseaux locaux et est aujourd'hui de plus en plus utilisé dans les réseaux de plus grandes tailles. Originellement, les trames Ethernet étaient diffusées à l'ensemble des machines connectées au réseau local et seule la station à laquelle la trame était destinée lisait cette trame. Aujourd'hui, l'utilisation d'Ethernet commuté permet de ne joindre que la station désirée.

Chaque équipement Ethernet est associé à une adresse unique, l'adresse Medium Access Control (MAC), codée sur 6 octets. Chaque trame Ethernet contient l'adresse MAC de l'équipement destinataire et émetteur. Les commutateurs Ethernet maintiennent des tables qui associent à chacun de leur port l'adresse MAC de l'équipement qui y est relié. Ainsi, lorsqu'un commutateur Ethernet reçoit une trame, il sait sur quel port la réémettre afin de la transmettre à la station destinataire.

MPLS

Dans un but d'uniformisation, les opérateurs se tournent de plus en plus vers des réseaux tout IP (pour lesquels tous les éléments du réseau sont adressés par IP). Le protocole Multiprotocol Label Switching [84] (MPLS) est apparu pour permettre la commutation de paquets dans ce type de réseau.

MPLS a connu un fort intérêt depuis le début des années 2000, car il possède les atouts suivants :

- MPLS est conçu de manière à « s'insérer » entre la technologie de couche liaison de données utilisée dans un réseau et le protocole IP. Il est ainsi compatible avec l'ensemble des technologies pour le transport des paquets IP utilisées aujourd'hui.
- La technique de commutation de paquets permet une vitesse de traitement plus rapide des paquets dans les routeurs du réseau
- La technique de commutation de paquets permet d'établir des circuits virtuels, c'est-à-dire des chemins prédéterminés dans le réseau qui seront empruntés pour acheminer les paquets IP entre les différents noeuds. L'utilisation de circuits virtuels permet ainsi à l'opérateur de mieux contrôler le trafic dans son réseau (c'est l'ingénierie de trafic).

MPLS utilise la notion d'étiquette, qui est une information ajoutée à un paquet IP lorsque celui-ci entre dans le réseau MPLS. Le choix de l'étiquette ajoutée à l'arrivée du paquet va déterminer entièrement le circuit virtuel, appelé Label Switch Path (LSP), qu'il va emprunter pour parvenir jusqu'à sa destination. Pour cela, chaque noeud du réseau MPLS (appelé Label Switch Router (LSR)) maintient une table qui pour chaque étiquette d'un paquet arrivant associe une étiquette « de sortie » ainsi que le prochain LSR à emprunter dans le circuit virtuel. Ainsi, lorsqu'un LSR reçoit un paquet, il consulte sa table de commutation et en fonction de l'étiquette de ce paquet, il la remplace par l'étiquette de sortie et transmet le paquet au LSR indiqué. Ce processus est répété par tous les LSR traversés par le paquet jusqu'au noeud destination (ou noeud de sortie du réseau MPLS).

2.2.5 Les protocoles des réseaux IP

Nous allons maintenant présenter les protocoles couramment utilisés dans les réseaux IP.

Internet Protocol

Le protocole Internet Protocol[74] (IP) est la base d'Internet. Il permet d'interconnecter les réseaux entre eux, quel que soit le lien physique et le protocole de la couche liaison de données utilisés dans ces réseaux. Pour cela, IP définit des adresses attribuées à chaque machine accessible par le réseau. Avec le protocole IP, chaque message à envoyer est fractionné dans différents paquets qui sont

envoyés un par un indépendamment les uns des autres. De plus, il est sans connexion, c'est-à-dire qu'aucune demande de communication n'est effectuée par un noeud auprès de son correspondant avant le début de l'envoi des paquets. Le service rendu par IP est non fiable, car IP n'apporte aucune garantie sur la bonne livraison des paquets.

Il existe deux versions du protocole IP : IPv4 et IPv6. IPv4 est utilisé par toutes les machines connectées à Internet aujourd'hui. Son remplaçant IPv6, adopté en 1995, peine encore à se généraliser dans le réseau.

Les protocoles de la couche transport

Un des rôles des protocoles de la couche transport est de multiplexer les possibilités de communication d'une machine. En effet, une machine dispose généralement d'une seule adresse IP et doit être en mesure d'établir plusieurs communications en même temps. C'est le rôle des ports, utilisés par la couche transport pour définir une communication entre deux applications exécutées entre deux machines. Les deux principaux protocoles de la couche transport sont le protocole TCP et le protocole UDP.

Le protocole Transmission Control Protocol[76] (TCP) fonctionne en mode connecté, c'est-à-dire qu'avant tout échange d'information avec un correspondant, une demande d'ouverture de session est effectuée. TCP est fiable : il garantit la bonne délivrance des données à la machine distante. Chaque paquet envoyé dont la réception n'a pas été confirmée est réémis. Enfin, TCP inclut un mécanisme de contrôle de flux afin de permettre d'adapter le débit d'une transmission à la capacité du réseau.

Le protocole User Datagram Protocol[77] UDP permet l'échange de messages appelés datagrammes. Contrairement à TCP, c'est un protocole sans connexion, non fiable, sans contrôle de flux. Le seul rôle de UDP est donc le transport des paquets IP d'une machine à une autre en utilisant les ports, afin de permettre la communication vers une application d'une machine distante. Bien qu'UDP ne possède pas toutes les fonctionnalités de TCP, il est très utilisé pour certaines communications. C'est le cas lorsqu'elles n'ont pas besoin d'une fiabilité de leurs échanges, mais aussi lorsque les communications doivent être effectuées rapidement et qu'il n'est pas possible d'attendre l'ouverture d'une session. Enfin, UDP est utilisé lorsque l'information contenue dans le paquet a un caractère temporel et ne serait pas utilisable après une réémission consécutive à la perte de ce paquet. Dans ce cas, l'information serait reçue trop tard. C'est le cas par exemple pour les communications relatives à un service de téléphonie.

Protocoles de la couche application

Les protocoles de la couche application définissent les règles à respecter lors de la communication entre deux applications distantes afin de permettre l'interprétation des informations à échanger. Les protocoles de la couche application s'appuient sur les couches inférieures et considèrent donc que la connectivité entre les applications distantes est assurée. Cependant, lorsque les applications ont des besoins autres que la connectivité, comme des besoins de qualité de service, le choix des protocoles de couches inférieures utilisés sera déterminant pour satisfaire ces besoins.

Les protocoles normalisés les plus utilisés sont associés à un port, qui est le port à contacter lorsque l'on souhaite établir une communication avec l'application associée. Voici quelques-uns des protocoles les plus connus :

- Hypertext Transfer Protocol (HTTP - Port 80) : Le protocole d'accès au Web
- File Transfer Protocol (FTP - Port 20 et 21) : Un protocole pour l'échange de fichiers

– Domain Name System (DNS - Port 53) : Le protocole permettant de rechercher la correspondance entre un nom de domaine et une adresse IP
– Simple Mail Transfer Protocol (SMTP - Port 25) : Le protocole pour l'envoi d'e-mails.

2.3 Incidents dans les réseaux

Dans cette partie, nous allons nous intéresser aux incidents susceptibles d'affecter les réseaux. Nous appelons incident un événement inattendu affectant le réseau qui va empêcher la délivrance d'une ou plusieurs communications aux utilisateurs. Un incident affecte généralement une partie d'un réseau seulement : il peut s'agir par exemple d'un ou plusieurs routeurs ou liens. Les communications qui utilisaient ces équipements pour être acheminées ne seront plus en mesure d'être délivrées. Nous étudierons deux catégories d'incident : les pannes et les attaques. Nous essayerons d'expliquer les causes d'apparition des incidents et de caractériser leurs effets. Ceci nous permettra de mieux comprendre les problèmes de fiabilité rencontrés dans les réseaux IP avant d'aborder les solutions envisageables à ces problèmes.

2.3.1 Les pannes dans les réseaux

Nous allons présenter dans cette partie la première catégorie d'incident affectant les réseaux IP : les pannes.

Fonctionnement correct d'un élément et élément en panne

On dit[43] d'un élément qu'il fonctionne correctement lorsqu'il se comporte conformément à ses spécifications : L'élément rend le service pour lequel il est conçu. À l'inverse, un élément est en panne ou défaillant lorsqu'il ne respecte plus ses spécifications : il ne peut plus rendre correctement le service pour lequel il est conçu. Une panne est la manifestation consécutive à la présence d'une faute dans le système.

L'origine des fautes

Les études[39] sur les réseaux de télécommunication ont permis de mettre en évidence les différentes origines des fautes présentes dans ces réseaux. (Les chiffres de cette section sont issus d'une étude réalisée entre 1992 et 1994 sur un réseau de téléphonie commuté et sont donnés à titre indicatif, mais ne représente pas nécessairement la répartition de l'origine des fautes dans un réseau IP actuel)

Origine humaine interne

Cette catégorie d'erreur regroupe les fautes imputables au personnel chargé d'administrer le réseau. Les fautes de cette catégorie sont généralement dues à une erreur humaine lors de la manipulation d'un élément du réseau, qu'il soit logiciel ou matériel.

Ce type de faute représentait un quart des fautes constatées durant la période de l'étude. Cependant, ce type de faute n'était à l'origine que de 14 % des pannes constatées chez les utilisateurs.

Origine externe

Cette catégorie regroupe les fautes ayant pour cause un événement extérieur au réseau de télécommunication. La faute peut alors avoir une origine humaine, c'est le cas lors des coupures accidentelles de câble, par exemple, ou encore lors des pannes d'approvisionnement en électricité. L'origine de ces fautes peut aussi être environnementale, par exemple lors des catastrophes naturelles telles que les tremblements de terre ou les inondations.

Ce type de faute représentait plus du tiers des fautes constatées durant la période de l'étude. Il était à l'origine du tiers des pannes constatées chez les utilisateurs.

Origine matérielle

Cette catégorie regroupe les fautes imputables à une défaillance matérielle d'un équipement utilisé dans le réseau. Les fautes de cette catégorie peuvent être la conséquence de l'usure naturelle des composants utilisés, comme les câbles ou l'alimentation électrique de ces composants par exemple.

Ce type de faute représentait 20 % des fautes constatées durant la période de l'étude, mais n'était à l'origine que de 14 % des pannes constatées chez les utilisateurs.

Origine logicielle

Cette catégorie regroupe les fautes imputables à une défaillance logicielle d'un équipement utilisé dans le réseau. Les fautes de cette catégorie peuvent être la conséquence d'une erreur de conception du logiciel : c'est le cas lorsque le service rendu par le logiciel ne correspond pas aux besoins de ses utilisateurs. L'erreur peut provenir de l'implémentation du logiciel lorsque le logiciel ne se comporte pas de manière attendue. Enfin, la faute peut provenir d'une erreur lors du déploiement du logiciel, s'il est mal utilisé par exemple.

Ce type de faute représentait moins de 15 % des fautes constatées durant la période de l'étude, mais n'était à l'origine que de 2 % des pannes constatées chez les utilisateurs.

Surcharge du réseau

Les fautes de cette catégorie se produisent lorsque la demande des utilisateurs est plus forte que la capacité du réseau à répondre à cette demande : la capacité de traitement ou d'acheminement des informations par les éléments du réseau n'est pas suffisante aux besoins. Ces fautes peuvent se produire si le réseau n'a pas été correctement dimensionné : il n'a pas la capacité de traitement ou d'acheminement des informations nécessaires à la délivrance du service pour lequel il a été conçu. C'est souvent le cas lors d'un événement exceptionnel qui provoque une forte augmentation de la quantité de messages dans le réseau.

Ce type de faute représentait 6 % des fautes constatées durant la période de l'étude. Cependant, l'impact de ces fautes était important, car elles étaient à l'origine que de 44 % des pannes constatées chez les utilisateurs.

Attaque du réseau

Les fautes de cette catégorie se produisent lorsqu'un utilisateur malveillant porte atteinte au bon fonctionnement du réseau. L'objectif de l'attaque est d'empêcher la délivrance du service rendu par

le réseau aux autres utilisateurs. Pour cela, l'attaquant peut endommager un élément du réseau, de manière physique ou logicielle, ou peut provoquer une surcharge.

Ce type de faute représentait 1 % des fautes constatées durant la période de l'étude et étaient à l'origine que de 1 % des pannes constatés chez les utilisateurs. Il faut cependant noter dans les réseaux IP, on constate une augmentation continue de ces attaques, et elles sont devenues un problème de sécurité majeur dans ces réseaux[59].

La section 2.3.2 sera consacrée à une étude plus approfondie de cette catégorie de fautes.

Les conséquences des fautes

La présence d'une faute entraîne l'apparition d'une panne. Les pannes qui affectent l'accès aux services distant par un utilisateur peuvent être classées en deux catégories :

- La panne qui affecte le réseau, c'est-à-dire les routeurs ou les liens qui sont utilisés pour acheminer les communications entre les utilisateurs
- La panne qui affecte un équipement en bout de réseau, dont dépend l'utilisateur pour se voir délivrer le service. Ce peut être la machine serveur ou le premier lien qui relie la machine utilisateur au réseau, par exemple.

Plusieurs études[65, 54] ont été menées pour caractériser ces pannes.

Panne d'un équipement en bout de réseau

Une analyse[65] effectuée sur les équipements de plusieurs grands fournisseurs d'accès à Internet montre que plus d'un tiers des pannes affectant ces équipements ont une origine humaine interne et que la durée de l'impossibilité d'accès au service est en moyenne de plusieurs heures pour ce type de panne. Les pannes d'origine matérielles ou logicielles sont moins communes et mènent à des interruptions de service d'environ une heure en moyenne. Dans cette étude, les pannes d'origine externe, due à la surcharge des équipements ou à une attaque étaient en nombre négligeable.

Il faut souligner que lorsque ce type de panne apparait, la livraison du service ne sera à nouveau possible que lorsque la panne sera corrigée et l'utilisateur n'a donc aucun moyen à sa disposition pour accéder au service avant cette correction. Une exception existe parfois lorsque les équipements délivrant le service sont répliqués et accessibles en plusieurs points du réseau. Dans ce cas, l'utilisateur peut se connecter à un équipement non touché par la panne afin de se voir délivrer le service. Nous verrons dans la section 2.5.1 des exemples de systèmes permettant la redirection automatique des communications de l'utilisateur vers un équipement valide afin d'assurer la continuité de livraison d'un service.

Panne dans le réseau

Différentes études réalisées dans Internet vont nous permettre de mieux comprendre les conséquences de l'apparition d'une panne dans le réseau IP. On peut en effet caractériser une panne selon différentes caractéristiques : son étendue, sa fréquence, sa durée et sa localisation.

Il est possible qu'une panne affecte plusieurs liens en même temps, notamment lorsqu'elle affecte un équipement réseau utilisé par l'ensemble de ces liens, tel qu'un routeur. En effet, dans ce cas, l'ensemble des liens connectés à cet équipement réseau peuvent être considérés comme affecté par une panne. Une étude[54] réalisée dans un important réseau de coeur durant 6 mois en 2002 montre que ce type de panne représente 16 % de l'ensemble des pannes observées dans le réseau. Dans 50 %

des cas, la panne affecte 2 liens en même temps, mais le nombre de liens affectés peut aller jusqu'à 20.

De plus, lorsqu'une panne affecte une fibre optique, il se peut qu'elle affecte plusieurs liens IP. En effet, il est commun que plusieurs liens distincts dans le réseau IP n'utilisent réellement qu'un même lien optique dans le réseau « physique ». L'étude montre que 10 % des pannes observées dans le réseau ont pour origine une panne d'un équipement optique qui entraine la panne simultanée de plusieurs liens dans le réseau IP.

Enfin, certaines pannes affectant plusieurs liens en même temps ne rentrent dans aucune des catégories ci-dessus, c'est le cas pour 3 % des pannes observées.

Concernant la fréquence d'apparition des pannes, l'étude montre que l'on peut qualifier une panne de « fréquente » si elle affecte régulièrement un même équipement réseau ou bien « d'aléatoire » s'il affecte n'importe lequel des équipements réseau de manière aléatoire. Cette tendance est aussi observée par une autre étude[19] observant les communications entre 31 noeuds situés en bout de réseau.

Il est ainsi montré[54] que plus de 50 % parmi les pannes affectant un unique lien sont de type « fréquentes » et ne concernent que 2,5 % des liens. Une panne de ce type apparait fréquemment : entre 1 et 40 heures en moyenne pour chacun des liens affectés. De plus, ce type de panne persiste moins longtemps que les autres : la panne disparait 200 secondes après son apparition en moyenne. Les auteurs de l'étude déduisent que ces pannes concernent des liens réseau dégradés, à la fin de leur durée de vie.

Les pannes de type « aléatoire », moins fréquentes, ont tout de même été observées plusieurs fois par jour dans le réseau. La durée de persistance de ces pannes est plus longue que pour les pannes de type « fréquentes » : 20 % des pannes sont encore présentes 1000 secondes après leur apparition.

Le temps de persistance des pannes affectant plusieurs liens simultanément est plus long que pour les pannes qui affectent un unique lien. En effet, plus de 25 % des pannes sont encore présentes 1000 secondes après leurs apparitions. La fréquence de ce type de panne est assez élevée, puisque ce type de panne a généralement été observé chaque jour.

Conséquences pour les communications de l'utilisateur

Une panne va interrompre une communication de l'utilisateur si elle affecte un équipement utilisé pour la délivrer. Il faut cependant relativiser la durée de persistance d'une panne lorsque l'on s'intéresse aux conséquences de cette panne sur les communications de l'utilisateur. En effet, il est possible qu'un mécanisme de rétablissement réseau soit utilisé de manière à ce que les communications « contournent » cette panne. Dans ce cas, on peut considérer que du point de vue de l'utilisateur, la panne est terminée.

L'étude[19] des communications en bout de réseau observe ainsi qu'avec les pannes affectant les communications entre deux noeuds en bout de réseau dont la durée est supérieure à 2 minutes, seulement 10 % d'entre elles ont une durée supérieure à 15 minutes et qu'elles durent en moyenne 3 minutes. De plus, une autre étude[2] a mesuré le taux de perte des paquets des communications de bout en bout à 0,42 %, et le taux de perte conditionnel (c'est-à-dire la probabilité qu'après qu'un paquet est perdu, le suivant le soit aussi) à 72 %.

Concernant la localisation des pannes, 62 % des pannes observées [19] étaient situées « à l'intérieur » du réseau et n'affectaient pas les équipements situés en bordure de celui-ci. Par conséquent, les communications affectées par ces pannes pouvaient être rétablies par un mécanisme de rétablissement réseau. D'autres travaux[27] montrent que lorsqu'une panne affecte une communication vers

un noeud relié à Internet par une connexion haut débit domestique, dans 60 % des cas celle-ci affecte le dernier lien, celui qui relie ce noeud au réseau. Cependant, le dernier lien n'est affecté que dans 16 % des pannes touchant les communications vers un serveur Web important. Si le dernier lien reliant l'utilisateur au réseau est affecté par une panne, il n'est pas possible d'utiliser un mécanisme de rétablissement réseau pour rétablir les communications.

2.3.2 Les attaques sur la disponibilité

Nous allons nous intéresser dans cette section aux attaques visant à perturber le bon fonctionnement d'un réseau afin d'empêcher la délivrance de service à l'utilisateur. Ces attaques sont nommées déni de service (ou DoS pour Denial Of Service). En effet, nous allons voir que les conséquences de ces attaques sont assez similaires aux conséquences des pannes.

Les attaques par déni de service visent à rendre indisponible un élément du réseau délivrant un service, ou rendre l'accès à celui-ci impossible. On peut classifier[92] les attaques par déni de service en deux grandes catégories :

- Les attaques qui exploitent une vulnérabilité, c'est-à-dire un défaut de spécification, de conception ou d'implémentation d'un protocole ou d'une application.
- Les attaques par surcharge de l'équipement délivrant le service visé ou de l'infrastructure réseau permettant l'accès à cet équipement.

Les attaques exploitant une vulnérabilité

Ces attaques résultent souvent de la non-prise en compte du problème de sécurité lors de la conception d'un protocole ou d'une application. En effet, la problématique de la sécurité n'est apparue que tardivement dans l'histoire des réseaux IP dont la conception se préoccupait avant tout d'assurer la connectivité[73].

Parmi ces attaques, on peut distinguer celles qui exploitent un défaut de spécification, de conception ou d'implémentation d'une application accessible par le réseau de celles qui exploitent un défaut de spécification, de conception ou d'implémentation d'un protocole.

Les attaques par surcharge

Nous allons aborder ici les attaques de déni de service par surcharge, dont le but est de rendre indisponible l'accès à un service en bloquant l'accès à la machine délivrant le service ou en empêchant la machine de délivrer le service. Ces attaques peuvent être classifiées[92] en trois catégories : les attaques sur les couches IP et Transport, les attaques sur la couche applicative, les attaques par saturation des liens.

Les attaques sur les protocoles

Cette catégorie regroupe les attaques qui exploitent les protocoles afin de surcharger la machine cible en terme de consommation de ressources mémoires et processeur afin de la rendre indisponible pour répondre à des requêtes légitimes.

Il existe de nombreuses attaques sur les couches IP, transport ou application. Elles se basent sur le fait que la machine cible doit consommer des ressources (en réservant une zone mémoire ou en générant et en envoyant un paquet, par exemple) chaque fois qu'un paquet particulier est reçu. On

peut citer par exemple l'attaque par « SYN Flood » [90], ou par établissement de sessions HTTP illégitimes.

Dans ce dernier cas, le principe de l'attaque est d'initier le maximum de sessions possibles avec le serveur de la cible afin que les requêtes légitimes pour établir une nouvelle session ne puissent pas être assurées par le serveur. Il est ainsi possible de concevoir des attaques temporelles[40], qui prennent en compte le temps au bout duquel une session inactive expire, pour ensuite immédiatement initier une nouvelle session, de manière à ce que le serveur ait toujours le nombre de maximums de sessions ouvertes. Contrairement aux attaques par surcharge, les attaques temporelles ne génèrent que peu de trafic et doivent être le sujet d'une attention particulière pour être détectée.

Les attaques par saturation des équipements réseau (ou attaques sur la bande passante)

Avec l'augmentation des accès à Internet de haut débit chez les particuliers et grâce à l'emploi de technique de déni de service distribué ou réfléchi (voir ci-dessous), il est permis à un attaquant de générer assez de trafic pour saturer la capacité d'acheminement des équipements réseau. Ainsi, l'attaque par déni de service d'une machine consiste à saturer les équipements réseau permettant l'accès à cette machine en plus de s'attaquer à la machine elle-même.

Par exemple, les attaques « UDP Flood » ou « ICMP Flood » [21], très simples, ne consistent qu'à envoyer le plus grand nombre possible de paquets utilisant le protocole UDP, ou encore l'Internet Control Message Protocol (ICMP), qui est normalement destiné à l'administration des réseaux IP ou UDP. Ces paquets ont une taille la plus grande possible, afin de saturer la bande passante du lien d'accès à la cible. Il suffit ainsi que l'attaquant soit capable de générer du trafic d'un débit supérieur à la capacité du lien d'accès au réseau de sa cible pour que l'attaque soit un succès, car dans ce cas, les requêtes légitimes des utilisateurs ne pourront être acheminées jusqu'au serveur.

Les attaques par déni de service réfléchi

Le principe d'une attaque par déni de service réfléchi[22] (RDoS pour Reflected Denial of Service) est l'utilisation d'un intermédiaire pour l'envoi de trafic à la cible. Ce type d'attaque se base sur l'usurpation de l'adresse IP source par l'attaquant qui la remplace par l'adresse IP de la machine cible. Il suffit ensuite à l'attaquant d'envoyer un message à une machine intermédiaire provoquant une réponse vers la machine source. Puisque l'adresse source du message est l'adresse de la cible, la machine intermédiaire va émettre la réponse vers la machine cible. Ainsi, lors de l'envoi de nombreuses requêtes usurpées à la machine intermédiaire, une attaque par surcharge est réalisée contre la machine cible. L'efficacité de ce type d'attaque est de plus multipliée si les paquets envoyés par la machine intermédiaire à la machine cible sont de taille supérieure aux paquets initialement envoyés par l'attaquant à la machine intermédiaire. Un type d'attaque RDoS utilisant ce procédé est par exemple l'attaque « DNS Amplification » [102].

Les attaques par déni de service distribué

Le principe d'une attaque par déni de service distribué[45] (DDoS pour Distributed Denial of Service) est de provoquer l'émission de trafic destiné à surcharger une cible par un ensemble de machines intermédiaires. Pour cela, ces machines doivent être contrôlées par l'attaquant afin que ce dernier provoque l'envoi de trafic depuis toutes les machines intermédiaires vers la cible à un même moment déterminé. C'est pourquoi ces machines intermédiaires sont appelées zombies. La prise de

contrôle des machines intermédiaires par l'attaquant peut être obtenue par diverses méthodes qui ne seront pas abordées ici. Les messages envoyés par les zombies à la cible pour la surcharger sont de même nature que ceux envoyés pour réaliser une attaque par surcharge, le plus souvent une attaque sur la bande passante.

L'avantage principal de ce type d'attaque est la quantité de trafic généré qui est multipliée avec le nombre de zombies utilisés. De plus, il est difficile de se prémunir de ce type d'attaque. Puisque l'origine du trafic provoquant la surcharge est multiple, il est difficile de déterminer l'ensemble des machines zombies pour bloquer leurs communications. Enfin, une attaque par déni de service distribué à un impact sur l'ensemble du réseau. En effet, alors qu'une attaque classique par surcharge va congestionner un unique chemin (de l'attaquant jusqu'à la cible), l'attaque distribuée va affecter l'ensemble des chemins qui vont des zombies jusqu'à la cible. Ainsi, l'ensemble des équipements réseau permettant l'accès à la cible vont être perturbés, et plus ces équipements seront proches de la cible et plus ils seront affectés.

On peut noter que les symptômes d'une attaque par déni de service distribué peuvent être assez identiques à une surcharge non malfaisante d'une machine, qui se produit lorsque la machine n'a pas assez de ressources pour délivrer son service lors d'une forte demande des utilisateurs.

Prévention et atténuation des DoS

Nous allons nous intéresser aux méthodes existantes permettant de se protéger au moins partiellement des attaques par déni de service. Les attaques par déni de service exploitant une vulnérabilité sont en générales les plus simples à déjouer puisqu'il suffit d'appliquer un correctif logiciel pour ne plus être menacé. Il n'en va pas de même pour les attaques par surcharge, qui n'exploitent aucune faiblesse, mais se contentent d'envoyer des requêtes valides en grand nombre.

La mise place de règles de contrôle d'accès aux différents éléments réseau d'un opérateur est une solution élémentaire[24]. En effet, elles vont permettre le filtrage du trafic directement dans le réseau, et par conséquent :

- Pendant une attaque, de rejeter le trafic à destination de la cible. Même si cette méthode bloque aussi les requêtes légitimes des utilisateurs, elle aura pour bénéfice de diminuer la quantité de trafic dans le réseau et permettre l'accès aux autres équipements. Une fois l'attaque étudiée, il est ensuite possible de filtrer plus « finement » le trafic, en rejetant uniquement le trafic illégitime ou en rejetant les sources émettrices de l'attaque.
- Avant une attaque, la mise en place de règle de filtrage afin de rejeter le trafic invalide. On peut par exemple envisager de rejeter le trafic avec une adresse IP source qui n'appartient pas au réseau dont il est issu, afin de limiter les attaques avec usurpation de l'adresse source. On peut aussi bloquer le trafic contenant des requêtes qui ne concernent pas les machines pour lesquelles il est destiné.

D'autres solutions ont été proposées dans de nombreux travaux de recherche[57]. Par exemple, la redirection du trafic légitime dans des parties du réseau non affectée par l'attaque peut être envisagée.

Motivations des attaquants

La motivation des attaquants pour réaliser une attaque par déni de service est, suivant leurs profils, variée. Ainsi, on peut classer les attaquants[78] suivant trois catégories : les vandales, dont le but est la réalisation d'un DoS afin de satisfaire l'envie de nuire ou de réaliser un défi technique, les idéologues, dont le but est de nuire à une entité pour des raisons idéologiques et enfin les truands, qui vont tenter de tirer un profit économique de leur attaque.

Avec l'utilisation massive d'Internet pour les services de la vie courante et le développement d'une très importante économie autour d'Internet, les conséquences d'une attaque DoS sont devenues très sérieuses, parfois désastreuses. On a ainsi vu[78] l'apparition d'opérations de chantage visant de grands groupes présents sur Internet, mais aussi de plus petites compagnies (telles que les compagnies de paris en ligne) depuis l'année 2004. L'attaquant menaçait de lancer une attaque DoS si la cible refusait de lui verser de l'argent. Enfin, à la suite d'un différent politique, des attaques DoS ont été lancées contre les sites de l'administration publique et des grandes entreprises d'Estonie afin de paralyser Internet dans ce pays.

Cet incident a été qualifié de première cyberguerre de l'histoire. Depuis lors, ce type d'attaque de grande envergure a été observé lors d'autres conflits internationaux, par exemple lors du conflit en Ossetie du Sud ayant eu lieu en 2008[53].

2.3.3 Sureté de fonctionnement

On appelle sureté de fonctionnement la mesure de la capacité d'un système à délivrer un service à l'utilisateur.

Mesure de la sureté de fonctionnement

Nous allons aborder ici les notions utilisées pour évaluer la sureté de fonctionnement dans les réseaux.

Fiabilité

On appelle fiabilité[44] $R(t)$ d'un élément du réseau la probabilité que cet élément soit en état de fonctionnement pendant une durée t, sachant que cet élément fonctionne initialement. On a :

$$R(t) = prob(\text{L'élément fonctionne pendant l'intervalle de temps [0,t]})$$

Soit $\lambda(t)$, le taux de défaillance d'un élément. $\lambda(t)$ est la probabilité de non-fonctionnement d'un élément à l'instant t sachant que cet élément était jusque-là en fonctionnement, on peut[26] exprimer $\lambda(t)$ par :

$$\begin{aligned}\lambda(t) &= \lim_{dt\to 0} prob(\text{Défaillance à l'instant } t+dt/ \text{ Fonctionnement à l'instant t}) \\ &= -\frac{\frac{d}{dt}R(t)}{R(t)}\end{aligned}$$

On a ainsi :

$$R(t) = e^{-\int_0^t \lambda(u)du}$$

Notamment, si $\lambda(t)$ est constante, ce qui est parfois admis dans les réseaux de télécommunication lorsque l'élément étudié n'est pas en début ou en fin de vie. Avec :

$$\lambda(t) = \lambda_0,$$

On a :

$$R(t) = e^{-\lambda_0 . t}$$

Dans ce cas, $1 - R(t)$ est la fonction de répartition d'une distribution exponentielle de paramètre λ_0. Par conséquent, la probabilité qu'une panne affecte l'élément étudié au temps $t + dt$, alors que l'élément fonctionnait au temps t, est la même que la probabilité pour qu'une panne affecte l'élément au temps dt.

Nous faisons référence à la fiabilité des communications tout au long de ce document. Les formules exprimées dans cette section s'appliquent aussi à la fiabilité des communications des utilisateurs. La fiabilité d'une communication est ainsi la probabilité qu'elle soit toujours correctement délivrée après une certaine durée. Pour être correctement délivré, le trafic issu de cette communication doit être acheminé à l'utilisateur afin de satisfaire les besoins pour le bon fonctionnement de l'application, les attentes de l'utilisateur envers le service rendu, et d'éventuels SLA. Ces différentes contraintes ont été décrites dans la section 2.1. Ainsi, la livraison de la communication peut être interrompue par la présence d'un incident sans que la communication cesse d'être correctement délivrée. Cela dépend de la durée de cette interruption et des spécificités de la communication.

Disponibilité

La notion de disponibilité d'un système reflète la probabilité d'observer ce système en état de fonctionnement. Ainsi, on appelle disponibilité A la probabilité qu'un élément soit en état de fonctionnement à un instant donné. On a :

$$\begin{aligned} A &= prob(\text{À un instant quelconque, l'élément fonctionne}) \\ &= \frac{d}{dt} R(t) \end{aligned}$$

Nombre de neufs

Pour évaluer la disponibilité d'un système ou d'un élément, on utilise souvent l'expression : « nombre de neufs de disponibilité ». Ainsi un élément possède une disponibilité A_i de n neufs si :

$$A_i > \sum_{i=1}^{n} 9 \times 10^{-i}$$

La table 2.2 présente le temps moyen de non-fonctionnement par année en fonction du nombre de neuf de disponibilité d'un élément.

On considère que pour un élément réseau, 5 neufs sont un objectif à atteindre. Cet objectif est facilement atteint aujourd'hui par les éléments réseau simples (switch, câble, etc.) mais il est beaucoup plus difficile à atteindre pour les éléments complexes (serveurs Web, etc.) ou pour une communication de bout en bout, qui utilise de nombreux équipements réseau pour être acheminée.

TAB. 2.2: Nombre de neufs de disponibilité

Neufs de disponibilité	Temps moyen de non-fonctionnement par an
0,9 %	36,5 jours
0,99 %	3,6 jours
0,999 %	8,8 heures
0,9999 %	53 minutes
0,99999 %	5 minutes
0,999999 %	32 secondes

MTTF et MTTR

On appelle « temps moyen avant la panne ou l'incident » (Mean Time To Failures, MTTF) le temps moyen avant l'apparition d'un incident affectant un élément d'un réseau.

On appelle « temps moyen de réparation » (Mean Time To Repair, MTTR) le temps moyen nécessaire pour réparer un élément défectueux.

Ces paramètres nous permettent d'évaluer la disponibilité d'un élément en considérant l'aspect temporel de son comportement. On a en effet :

$$A = \frac{MTTF - MTTR}{MTTF}$$

De plus, lorsque le taux de panne $\lambda(t) = \lambda_0$ est constant, nous avons vu que $1 - R(t)$ est la fonction de répartition d'une distribution exponentielle de paramètre λ_0. Or, $1 - R(t)$ représente aussi la fonction de répartition des incidents affectant l'élément étudié. Par conséquent, le temps X avant l'incident suit une distribution exponentielle de paramètre λ_0. On a ainsi, avec $E(X)$ l'espérance de X :

$$\begin{aligned} MTTF &= E(X) \\ &= \frac{1}{\lambda_0} \end{aligned}$$

Stratégie en présence de fautes

Afin de permettre des communications fiables dans un réseau, différentes stratégies sont envisageables.

L'évitement des fautes

C'est l'ensemble des techniques de conception, de fabrication et d'implémentation qui permettent de produire un réseau fiable. Ces techniques comprennent la protection contre l'environnement extérieur, la résistance à l'usure, les preuves de programme, etc.

La tolérance aux fautes

C'est l'ensemble des techniques de conception qui permettent à un réseau de continuer à fonctionner même en présence d'un incident dans l'un de ses éléments. Par exemple, la redondance qui

consiste à multiplier certains éléments du réseau de manière à ce que si l'un d'eux tombe en panne, un autre élément soit disponible pour le remplacer.

La complexité et l'administration décentralisée des réseaux IP font qu'il est probablement impossible de réaliser un réseau capable d'anticiper et d'éviter toutes les fautes. Afin d'améliorer la fiabilité de la délivrance des communications à l'utilisateur, nous allons donc nous intéresser à un mécanisme de tolérance aux fautes : le rétablissement réseau.

2.4 Rétablissement réseau et routage

Nous allons étudier dans cette partie les mécanismes de rétablissement réseau. Ces mécanismes permettent de maintenir la délivrance d'une communication dans un réseau lorsque celui-ci est affecté par un incident. En particulier, nous nous intéresserons au routage, dont l'une des missions est de remplir cette fonction.

2.4.1 Les mécanismes de rétablissement

Nous allons tout d'abord introduire les concepts généraux liés au rétablissement réseau.

Définition

On appelle mécanisme de rétablissement réseau le mécanisme permettant de maintenir la connectivité entre deux noeuds d'un réseau lorsqu'un incident affecte les communications entre ceux-ci. En effet, les incidents peuvent affecter un ou plusieurs liens ou routeurs du réseau ce qui a pour conséquence l'impossibilité d'utiliser les liens ou routeurs affectés pour délivrer une communication.

Pour cela, le mécanisme, une fois l'incident détecté et localisé, redirige le trafic perturbé de manière à lui faire emprunter un chemin du réseau non affecté par l'incident. Le trafic peut ainsi atteindre sa destination.

Le mécanisme de rétablissement peut être géré de manière centralisée : les décisions sont alors prises par une entité centrale qui à une vision complète du réseau et des incidents l'affectant. Le contrôle peut aussi être décentralisé et dans ce cas, les décisions sont partagées par chacun des noeuds qui utilisent le mécanisme de rétablissement. Le mode de contrôle distribué, bien que plus complexe, ne nécessite pas de communications entre une entité centrale et les noeuds chargés de rediriger le trafic. Par conséquent, il est moins sensible aux pannes, plus réactif et nécessite moins de ressources. Il est ainsi le plus couramment utilisé dans les réseaux IP et nous étudierons ce type de mécanisme dans ce document.

Chemins de rétablissement

On appelle chemin principal ou primaire emprunté par une communication le chemin emprunté par son trafic lorsque le réseau est en situation de fonctionnement normal, en l'absence d'incidents. On appelle chemin alternatif ou de secours le chemin emprunté par ce trafic lorsque le mécanisme de rétablissement est enclenché pour contourner un incident qui affecte le chemin primaire.

Un chemin alternatif peut être dynamique si celui-ci est calculé à l'issue de la détection d'un incident sur le chemin principal. À l'inverse, il peut être précalculé lorsqu'il a été déterminé à l'avance de manière à anticiper un incident.

L'avantage de l'utilisation de chemins alternatifs précalculés est lors de la détection d'un incident, le chemin alternatif est déjà « prêt » et peut-être immédiatement utilisé. Cependant, ceci nécessite

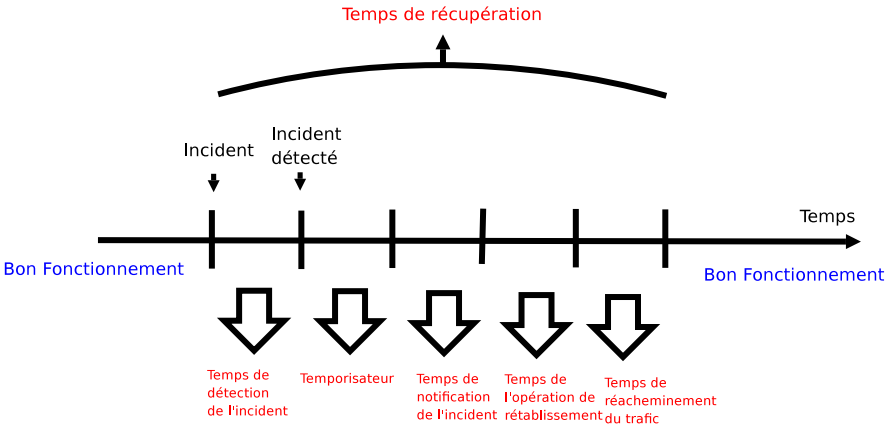


FIG. 2.3: Le cycle de rétablissement d'une communication

d'anticiper l'apparition des différents incidents pouvant affecter le réseau. De plus, l'utilisation de chemins dynamiquement calculés permet de proposer un chemin alternatif adapté à la localisation de l'incident, ce qui permet de le contourner « au plus près ».

Besoins topologiques

Pour fonctionner, le mécanisme de rétablissement réseau impose des contraintes sur la topologie du réseau. En effet, quel que soit l'élément du réseau affecté par un incident, le mécanisme de rétablissement doit être en mesure de proposer un chemin alternatif qui contourne cet incident. Pour cela, il faut que le chemin alternatif soit disjoint du chemin principal au moins au niveau du segment de chemin affecté par l'incident.

Un point unique de panne, est un élément du réseau, qui, s'il était affecté par un incident, ne pourrait être contourné et isolerait par conséquent une partie du réseau. Leur présence doit être évitée afin de permettre le fonctionnement du mécanisme de rétablissement.

C'est la topologie du réseau, c'est-à-dire à dire la position des différents noeuds et liens qui le compose, qui va déterminer s'il est possible de proposer un chemin alternatif lorsqu'un incident est présent, en fonction de la localisation de celui-ci. De même, la présence de points uniques de panne dépend de cette topologie.

D'autres besoins liés à la topologie du réseau peuvent être considérés, par exemple si des besoins de qualité de service doivent être satisfaits, il faut que le chemin alternatif soit capable de satisfaire ces besoins. Par exemple, si le chemin alternatif est trop long, le délai de livraison des communications utilisant ce chemin risque d'être beaucoup plus important qu'avec le chemin primaire.

Cycle de rétablissement

Les différentes étapes du rétablissement d'une communication par un mécanisme de rétablissement suivent un cycle qui est schématisé dans la figure 2.3.

Nous allons expliquer les différentes étapes de ce cycle. La présence d'une faute dans le réseau va se manifester par l'apparition d'un incident. Au bout d'une période appelée « temps de détection de l'incident », l'incident est détecté par un élément du réseau. L'élément qui a détecté l'incident peut attendre une période de temps appelée « temps de temporisation » avant de signaler cet incident de manière à ne pas déclencher le mécanisme de restauration en cas de fausse alerte ou de panne

transitoire, par exemple. Ensuite, la notification de l'incident va être envoyée aux différents éléments du réseau concernés par le mécanisme de rétablissement réseau. On appelle « temps de notification » le temps nécessaire à cette étape. Les éléments informés de l'incident vont ensuite déclencher le mécanisme de rétablissement, par exemple en calculant et en établissant un chemin alternatif. Le temps nécessaire à ce déclenchement est appelé « temps de l'opération de rétablissement ». Enfin, une fois le chemin alternatif établi, il faut que le trafic l'emprunte et arrive à destination avant que l'on puisse considérer que les communications sont rétablies. Cette période est appelée « temps de réacheminement du trafic ». La période couvrant l'ensemble du rétablissement, de l'apparition de l'incident à l'acheminement du trafic à sa destination par le chemin alternatif est appelé « temps de rétablissement ».

Une fois l'élément défaillant réparé, l'opération inverse doit être réalisée. En effet, l'utilisation des chemins alternatifs n'est généralement pas optimale une fois les chemins primaires rétablis. On appelle cycle de réversion le processus de rétablissement des chemins originels dans le réseau aux dépens des chemins alternatifs.

Critères de performance du mécanisme de rétablissement

Il existe de nombreux mécanismes de rétablissement qui ont chacun leurs avantages et leurs inconvénients. Nous allons présenter ici les critères qui permettent d'évaluer les performances des différents mécanismes de rétablissement réseau.

Portée du mécanisme

La portée du mécanisme de rétablissement est sa capacité à rétablir la connectivité en cas d'incident, en fonction des différents scénarios d'incident pouvant affecter le réseau. En effet, certains mécanismes de rétablissement réseau peuvent ne pas être utiles pour parer à certains scénarios d'incident. Par exemple, certains mécanismes vont permettre l'utilisation d'un chemin alternatif disjoint par les liens du chemin primaire, mais celui-ci pourrait être inefficace en cas d'incident affectant certains noeuds. Ou encore, certains mécanismes peuvent être incapables de rétablir la connectivité en cas de multiples pannes.

Certains mécanismes de rétablissement introduisent la notion de classe de trafic et le rétablissement est effectué en fonction de la classe de trafic concernée. Cette différenciation interne de la portée du mécanisme peut s'avérer pertinente dans les réseaux multiservices, où certaines communications sont plus sensibles que d'autres.

Temps de rétablissement

Le temps de rétablissement correspond au temps nécessaire à réaliser l'ensemble du cycle de rétablissement. C'est un paramètre essentiel de la performance d'un mécanisme de rétablissement. En effet, plus le temps de rétablissement est long et plus la période d'interruption des communications sera longue. Il est donc essentiel que le temps de rétablissement soit court de manière à ce que la délivrance des services ne soit pas trop perturbée en cas d'incident.

Qualité du chemin alternatif

La qualité du chemin alternatif peut varier en fonction du mécanisme de rétablissement utilisé. Ainsi, on considère que la qualité du chemin alternatif est un critère de performance du mécanisme de rétablissement. Parmi les critères de qualité du chemin alternatif, on peut citer la présence d'une bande passante disponible garantie, la pénalité de délai d'acheminement du chemin alternatif par rapport au chemin primaire, la gigue sur le chemin alternatif, etc.

De plus même si ce n'est pas directement en rapport avec la qualité du chemin alternatif, on peut considérer que la probabilité de déséquencement et la duplication des paquets lors de la mise en place et de la réversion du chemin alternatif sont un critère important pour l'évaluation des performances du mécanisme de rétablissement.

Consommation de ressource et passage à l'échelle

La consommation de ressources nécessaire au bon fonctionnement du mécanisme de rétablissement doit être suffisamment contenue pour rendre son utilisation réaliste. Parmi les ressources typiquement utilisées par les mécanismes de rétablissement, on peut citer la bande passante réseau consommée par les messages échangés entre les noeuds qui utilisent le mécanisme ou encore le temps de calcul nécessaire à ces noeuds pour réaliser les diverses opérations du processus de rétablissement.

De plus, les performances du mécanisme et les ressources qu'il consomme ne doivent pas trop dépendre de la taille du réseau. Dans le cas contraire, on dit que le mécanisme de rétablissement ne passe pas à l'échelle. Par exemple, la quantité d'information stockée dans les noeuds du réseau ne doit pas s'accroître trop fortement avec l'augmentation de la taille du réseau.

Stabilité

La stabilité du mécanisme de rétablissement mesure les conditions pour déclencher le processus de rétablissement. Un mécanisme de rétablissement se doit de ne pas enclencher le processus de rétablissement à la moindre variation de l'état du réseau, mais uniquement lorsque la présence d'un incident est avérée et le justifie. La stabilité du mécanisme de rétablissement sera bonne si ce dernier enclenche le processus de rétablissement de manière raisonnée, en concordance avec une vision temporelle déterminée de l'état dans lequel doit être le réseau.

2.4.2 Les protocoles de routage

Nous allons maintenant présenter les protocoles de routage, qui sont les principaux mécanismes permettant le rétablissement réseau utilisé dans les réseaux IP aujourd'hui.

Rôle du protocole de routage

Le but du routage dans les réseaux est d'assurer la connectivité entre l'ensemble des noeuds du réseau. Pour cela, les noeuds situés à l'intérieur du réseau, les routeurs, sont utilisés pour acheminer les communications qui transitent par eux de façon à ce qu'elles soient acheminées jusqu'à la destination souhaitée. Pour cela, chaque routeur maintient une table de routage dont le rôle est d'indiquer le noeud voisin du routeur auquel le trafic doit être transmis, pour chaque destination possible du trafic.

Le rôle du protocole de routage dynamique est de renseigner automatiquement les tables de routage. Il doit ainsi découvrir la topologie du réseau. De plus, lorsque cette topologie change, il doit être capable de modifier les tables de routage des routeurs en conséquence. C'est pourquoi le protocole de routage joue le rôle de mécanisme de rétablissement dans les réseaux. En effet, lorsqu'un incident

survient dans le réseau, le protocole de routage doit détecter la modification de la topologie entrainée par cet incident et modifier les tables de routage des routeurs de manière à « contourner » cet incident et de continuer à assurer la connectivité entre les noeuds, si toutefois la nouvelle topologie du réseau le permet (voir la section 2.4.1, ci-dessus).

Ainsi, le protocole de routage permet l'établissement de routes, ou chemins (primaires) entre chaque paire « noeud source, noeud destination » du réseau. Cette route est formée par la succession de liens et routeurs à emprunter pour relier ces deux noeuds. Parmi l'ensemble des routes possibles pour relier un noeud source à un noeud destination, le protocole de routage choisit généralement celle de moindre coût pour le réseau. Le coût d'une route est déterminé par une certaine métrique. Voici des exemples de métriques utilisées dans les protocoles de routage :

- Le nombre de sauts d'une route : c'est le nombre de noeuds traversés par la route pour relier le noeud source au noeud destination.
- Le temps d'acheminement de la route : c'est le temps requis pour acheminer le trafic du noeud source au noeud destination (bien que généralement c'est le temps aller-retour qui est considéré)
- La bande passante d'une route : c'est la bande passante (disponible ou maximale) pouvant être obtenue sur cette route. On l'obtient en prenant le minimum de la bande passante de chaque lien et routeur traversés par la route. Le coût d'une telle route est inversement proportionnel à la bande passante obtenue.

Routage par la source

Le principe de routage par la source[18] est de permettre au noeud qui émet une communication de choisir la route à emprunter pour joindre une destination. Pour cela, le noeud source ajoute aux paquets émis la succession des routeurs à emprunter pour arriver jusqu'à la destination. Ainsi, chaque routeur traversé lit à l'intérieur des paquets qu'il reçoit quel est le prochain routeur à qui envoyer ces paquets.

Il existe deux façons d'utiliser le routage par la source :

- Le mode strict, où tous les routeurs à traverser pour atteindre le noeud destination doivent être ajoutés par le noeud source aux paquets émis.
- Le mode libéral, où seuls certains routeurs sont indiqués dans les paquets émis par le noeud source.

Avec l'utilisation du mode strict, chaque paquet indique entièrement la succession de noeuds à emprunter pour joindre la destination, par conséquent, son principal avantage est que les différents routeurs traversés n'ont pas besoin d'utiliser leur table de routage (et n'ont par conséquent pas besoin d'utiliser un protocole de routage dynamique) puisqu'ils peuvent lire directement dans le paquet à transmettre à quel routeur ils doivent l'envoyer. L'inconvénient de cette méthode est l'accroissement du volume de trafic envoyé par un noeud puisque celui-ci doit ajouter à chacun des paquets envoyés les adresses de chacun des routeurs à traverser pour joindre la destination. Cet inconvénient est donc particulièrement fort lorsque la route à emprunter est longue.

L'utilisation du mode libéral permet au noeud émetteur de choisir certains des routeurs dits « intermédiaires » à emprunter pour joindre la destination. L'acheminement du trafic entre deux routeurs intermédiaires utilise le routage « normal » : il repose sur les tables de routage des différents routeurs formant la route entre les deux routeurs intermédiaires. L'avantage de ce mode par rapport au mode strict est qu'il permet au noeud source de ne pas indiquer l'ensemble des noeuds à traverser dans chacun de ses paquets, et ainsi de ne pas trop augmenter le volume de trafic. Toutefois, il peut garder un certain contrôle sur la route utilisée par le trafic en spécifiant certains des routeurs à emprunter.

Il faut remarquer que le routage par la source est parfois interdit dans les réseaux des opérateurs

constituants Internet : les paquets utilisant ce type de routage sont traités de manière normale ou même rejetés. Ceci est expliqué par diverses raisons : l'augmentation du volume du trafic engendré par son utilisation, le potentiel risque de sécurité lié au fait de laisser à l'utilisateur le choix de la route empruntée par son trafic, par exemple.

Le principe du routage par la source est intéressant pour nos travaux, car il donne à l'utilisateur la capacité de choisir la route utilisée par ses communications. En effet, certains travaux[27] ont mis en évidence qu'en cas de défaillance d'un élément du réseau, le routage par la source permet parfois de contourner cet élément et ainsi d'assurer la continuité des communications en un temps plus court que ne le permet un protocole de routage dynamique classique. Nous verrons dans la suite de nos travaux que dans certains cas, ce mécanisme constitue en effet une alternative viable au routage dynamique pour la récupération rapide après un incident dans le réseau.

Protocole de routage IP

Nous appelons protocole de routage IP les protocoles de routage dynamiques destinés à permettre l'acheminement des communications en se basant sur l'adresse de destination IP de leurs paquets. Il existe de nombreux protocoles de ce type et leur fonctionnement varie en fonction de la nature des réseaux dans lesquels ils sont utilisés.

Il existe deux catégories de protocole de routage IP utilisées dans les réseaux fixes : les Internal Gateway Protocol (IGP) et les External Gateway Protocol (EGP). Les EGP sont dédiés au routage entre différents domaines administratifs ou Autonomous System (AS). Un AS est un réseau administré par une même entité. Les IGP sont dédiés au routage à l'intérieur d'un même AS. Parmi les IGP, on peut distinguer deux modes de fonctionnement : les protocoles à vecteur de distance et les protocoles à état des liens

Les protocoles à vecteur de distance

Les protocoles à vecteur de distance reposent sur l'algorithme de fonctionnement suivant : chaque routeur transmet périodiquement sa table de routage à ses voisins (à l'initialisation du protocole, cette table est vide). Lorsqu'un routeur A reçoit la table d'un de ses voisins B, il peut compléter sa propre table avec les informations reçues. En effet pour une certaine destination, si le routeur A ne possède pas de route pour cette destination dans sa table d'acheminement ou si le cout de cette route est supérieur au cout de la route utilisée par le routeur B voisin additionné au cout du lien entre le routeur A et son voisin B, alors le routeur A peut mettre à jour sa table d'acheminement de manière à faire transiter le trafic pour cette destination par B.

Parmi les protocoles à vecteur de distance, on peut citer Routing Information Protocol[52] (RIP) qui a été utilisé par le passé, mais délaissé depuis. Interior Gateway Routing Protocol[9] (IGRP) a pour principale différence avec RIP l'utilisation de différentes métriques. Il est la propriété de Cisco et n'est utilisé que sur leur matériel. Enhanced Interior Gateway Routing Protocol [8] (EIGRP) est une évolution de IGRP qui pallie certains problèmes rencontrés par son prédécesseur.

Les protocoles à état de lien

Les protocoles à état de lien fonctionnent selon l'algorithme suivant : chaque routeur se renseigne sur le coût des liens entre lui et ses voisins et il transmet cette information à l'ensemble des routeurs du réseau. Chaque routeur connait ainsi l'ensemble de la topologie du réseau ainsi que le coût de chacun des liens. Il peut ainsi calculer, en utilisant l'algorithme de Dijkstra, la route de moindre coût pour

chacune des destinations possibles dans le réseau. Il sait ainsi par quel noeud voisin faire transiter le trafic en fonction de sa destination et met à jour sa table d'acheminement en conséquence.

Parmi les protocoles à état de lien, on peut citer Open Shortest Path First[62] (OSPF) qui a été développé par l'IETF et Intermediate System to Intermediate System[66] (IS-IS) qui est très proche d'OSPF, mais développé au sein de l'ISO.

Les protocoles EGP

Les protocoles de routage EGP ont pour rôle d'assurer la connectivité entre les différents AS du réseau. Il permet d'établir la connectivité entre un AS source et un AS destination en calculant la succession d'AS à emprunter, selon des accords conclus entre les opérateurs des différents AS. La façon dont le trafic est acheminé à l'intérieur des AS est déterminée par l'IGP utilisé à l'intérieur de chacun des AS. Par conséquent, le protocole EGP n'est utilisé que sur les routeurs « en bordure » d'AS, c'est-à-dire les routeurs reliés à des routeurs appartenant à un AS différent.

Border Gateway Protocol [83] (BGP) est le protocole EGP quasi exclusivement utilisé aujourd'hui dans le réseau Internet.

Les protocoles des MANET

Le routage dans les réseaux de type Mobile Ad Hoc Networks (MANET) est spécifique, car ce type de réseau, constitué de noeuds mobiles reliés entre eux par des liens sans fil, amène des contraintes de routage spécifiques. En particulier, dans les MANET, la mobilité des noeuds entraine un changement fréquent de la topologie du réseau, ce qui nécessite d'actualiser souvent et rapidement les tables de routage des noeuds.

Pour l'établissement des routes, deux principales catégories de protocole existent. La première concerne les protocoles dits réactifs qui établissent les routes « à la demande » : ce n'est que lorsqu'un noeud à besoin de joindre un certain noeud destination qu'il déclenche un mécanisme de découverte de la topologie du réseau de manière à calculer la route à emprunter pour joindre ce noeud. On peut citer parmi ces protocoles le protocole Ad Hoc On Demand Distance Vector [72] (AODV). Dans ce mécanisme, la recherche d'un noeud destination s'effectue en transmettant une requête à tous les noeuds du réseau. Cette requête est propagée jusqu'au noeud cherché ou jusqu'à un noeud possédant une route pour ce noeud. Un message de réponse est alors retourné à la source. Lors du transit des messages de requête et de réponse, chaque noeud traversé retient le noeud lui ayant transmis le message afin d'établir des routes grâce à l'algorithme à vecteur de distance vu plus haut. Un autre de ces protocoles est le protocole Dynamic Source Routing [34] (DSR). De la même manière que pour AODV, ce mécanisme transmet une requête à tous les noeuds du réseau pour découvrir une destination et un message de réponse est ensuite retourné à la source. Lors du transit des messages de réponse, chaque noeud traversé ajoute son adresse au message. Ainsi, lorsque le message de réponse atteint la source, celle-ci est informée de l'ensemble des adresses des noeuds à traverser pour atteindre la destination. Le noeud source utilise alors la technique de routage par la source vue à la section 2.4.2 pour acheminer les communications jusqu'à la destination.

La seconde regroupe les protocoles dits proactifs qui à la manière des protocoles de routage IGP classiques calculent à l'avance les routes pour l'ensemble des destinations possibles. À la différence des IGP classiques, la connaissance de la topologie par les noeuds doit être actualisée fréquemment et par conséquent, une attention particulière doit être apportée à la bande passante utilisée par les messages échangés entre les noeuds afin de remplir cette tâche, notamment lors du passage à l'échelle.

On peut citer, parmi ces protocoles, le protocole Optimized Link State Routing[11] (OLSR) qui utilise un algorithme de type état de lien, mais limite la transmission des informations topologiques dans le réseau à certains noeuds particuliers et le protocole Fisheye State Routing [71] (FSR) qui utilise aussi un algorithme de type état de lien, mais qui diminue la fréquence de la transmission des informations topologiques aux noeuds éloignés.

Certains protocoles, dits hybrides, utilisent ces deux technologies. Ils fonctionnent généralement de façon proactive pour établir des routes avec les noeuds proches et de manière réactive pour établir des routes avec des noeuds éloignés.

Les protocoles de routage des MANET sont intéressants pour nos travaux, car ils sont conçus de manière à réagir rapidement lors d'une modification de la topologie du réseau. Certains des mécanismes utilisés dans ces protocoles seront utilisés dans nos mécanismes destinés à rétablir rapidement la connectivité après un incident.

2.4.3 Rétablissement dans MPLS

Le protocole MPLS connait un grand succès, car il permet aux opérateurs de réseau de faire de l'ingénierie de trafic, c'est-à-dire de leur permettre un plus grand contrôle sur la façon dont est acheminé le trafic dans leurs réseaux. En particulier, des mécanismes de rétablissement réseau ont été proposés pour MPLS[100]. Nous allons présenter ces mécanismes.

Restauration globale

Avec le mécanisme de restauration globale, lorsqu'un incident affecte un LSP (les chemins calculés par les noeuds MPLS), c'est le noeud situé en entrée du LSP qui, lorsqu'il est informé de cet incident, calcule un nouveau LSP qui contourne cet incident. Avant de pouvoir être utilisé, le LSP doit être mis en place par un message propagé dans le réseau. L'avantage de cette technique est qu'elle est dynamique et ne requiert pas de configuration préalable. Par contre, le temps de rétablissement d'un LSP utilisable est long.

Protection globale

Avec le mécanisme de protection globale, un LSP alternatif est calculé à l'avance pour chaque LSP du réseau à protéger. Lorsqu'un incident affecte un LSP, c'est le noeud situé en entrée du LSP qui, lorsqu'il est informé de cet incident, cesse d'utiliser le LSP primaire et utilise le LSP alternatif pour acheminer ses communications. L'avantage de cette technique est que le temps de rétablissement est plus court qu'avec la restauration globale, puisque le LSP alternatif est immédiatement utilisable. Par contre, sa mise en place est plus complexe et coûteuse, en particulier si le nombre de LSP à protéger est important ou si des ressources doivent être réservées pour le LSP alternatif.

Protection locale

MPLS dispose de mécanismes de protection locale appelés MPLS Fast Reroute[67]. Dans ce mécanisme, ce sont les éléments réseau, liens ou routeurs, qui sont protégés. Pour chaque élément réseau à protéger et susceptible d'être affecté par un incident, un LSP « détour » est calculé à l'avance de manière à contourner cet élément. Ainsi, si un incident affecte un élément réseau protégé, les noeuds MPLS situés directement en amont de l'incident modifieront les LSP qui transitent par l'élément défaillant de manière à utiliser le LSP « détour » et contourner l'incident. L'avantage de ce mécanisme est que son temps de rétablissement est le plus court. En effet, lorsqu'un incident se produit, les LSP

alternatifs sont directement utilisables et le sont plus rapidement que pour les mécanismes globaux, puisque c'est le noeud situé directement en amont de l'incident qui déclenche l'utilisation des LSP alternatifs et que ce noeud est informé plus rapidement de l'incident, puisque plus proche de celui-ci. De plus, sa mise en place est moins coûteuse que celle de la protection globale, puisque le nombre de LSP alternatif à établir est fonction du nombre d'éléments réseau et non du nombre de LSP. Cependant, la protection locale ne permet pas d'adapter le rétablissement aux besoins spécifiques d'un LSP et sa mise en place reste complexe.

2.4.4 Autres mécanismes de rétablissement

Nous allons présenter dans cette partie certains travaux académiques présentant des mécanismes de rétablissement de niveau IP permettant une amélioration des performances par rapport aux mécanismes de routage classique dans ce domaine.

Routes de secours de bout en bout de niveau IP

Cette méthode consiste à calculer, pour chaque routeur, deux routes possibles pour atteindre chaque destination. Ceci permet, en cas d'apparition d'un incident sur un lien de la première route, d'emprunter la deuxième route pré calculée. Il existe néanmoins des contraintes à cette méthode. Il est en effet nécessaire que la route de secours soit valide, et par conséquent qu'elle ne soit pas touchée par l'incident. Il existe plusieurs algorithmes permettant le calcul d'une route de secours. On peut notamment citer Two Disjoint Shortest Path (TDSP)[58], une extension du protocole OSPF qui utilise un algorithme de Dijkstra modifié et permet de calculer, en plus du plus court chemin, un deuxième plus court chemin, totalement disjoint du premier, sous la condition qu'un tel chemin existe.

p-Cycles

À l'origine conçue pour les réseaux optiques WDM, la protection par « p-cycles » a été adaptée [96] pour la restauration rapide dans les réseaux IP conventionnels. Les « p-cycles » sont des boucles fermées, traversant un certain nombre de nœuds du réseau. Elles vont permettre la restauration en cas d'incident affectant un certain nombre de liens ou de routeurs, suivant la topologie de la boucle par rapport à celle du réseau.

Lors de l'apparition d'un incident affectant un lien, un routeur adjacent au lien affecté qui aurait dû acheminer des données via le lien défectueux encapsule ces données de façon à ce qu'elles soient acheminées par le cycle précalculé. Ensuite, les données voyagent dans le cycle jusqu'à ce qu'elles arrivent dans un routeur où le coût du chemin pour atteindre la destination initiale des données est moindre que le coût relevé lors de l'encapsulation. Ce routeur décapsule alors les données pour les acheminer jusqu'à la destination.

Les difficultés rencontrées pour la mise en place de cette protection sont la façon d'organiser les cycles dans un réseau donné de façon à minimiser l'augmentation du temps d'acheminement induit par le passage des données dans le cycle, tout en se protégeant de l'ensemble des incidents possibles. Les auteurs donnent les conditions pour parvenir à cet optimum, mais le calcul de la position des cycles est une optimisation combinatoire NP-difficile.

Arbres de secours

Le but de cette technique est de construire deux arbres recouvrant le graphe correspondant au réseau à protéger. La racine de ces arbres est la source d'émission des données. Les arbres doivent

être construits de telle façon que si l'on supprime n'importe quel nœud autre que la source, tous les autres nœuds du graphe doivent être reliés à au moins un des deux arbres. Ceci n'est réalisable que sous l'hypothèse que le graphe est redondant, c'est-à-dire que chaque nœud peut être relié à la source par deux chemins disjoints. Deux algorithmes[55] pour parvenir à ce but, en cas d'incident affectant un lien ou un nœud, ont été proposés. Malheureusement, l'algorithme ne calcule pas les arbres de coût minimum pour parvenir à une destination. Des améliorations de ces algorithmes ont cependant été développées pour construire des arbres qui amènent à une diminution notable du délai entre la source et la destination [106].

Resilient Routing Layers

Cette technique [41] consiste à calculer à l'avance des tables de routage dans un réseau où un ou plusieurs liens sont défaillants. Ainsi, lorsqu'une panne sur un lien est détectée, le routeur utilise la table de routage correspondant au sous-réseau dont le lien défectueux a été omis. L'efficacité de ce protocole dépend grandement du nombre de sous-réseaux précalculés, c'est-à-dire du nombre de liens omis dans chaque sous-réseau pré calculé. En effet, dans le cas où, par exemple, on précalcule deux sous-réseaux, et que dans chaque sous-réseau la moitié des liens ont été retirés, lors d'un incident, le sous-réseau utilisé pour le routage va être loin de l'optimal en terme de nombre de sauts. À l'inverse, si l'on calcule autant de sous réseau qu'il y a de liens, le sous-réseau utilisé sera optimal, mais le coût en calcul et en mémoire est trop important.

2.5 Routage et réseaux pair-à-pair

Dans cette partie, nous allons présenter les systèmes de type pair-à-pair, ayant connu un développement particulièrement important ces dernières années, dédiés à l'acheminement des communications. Nous allons en effet nous intéresser aux systèmes de ce type destinés à améliorer le routage dans les réseaux. Nous pensons en effet que ces systèmes peuvent être une solution pour améliorer la fiabilité des communications des utilisateurs des réseaux.

2.5.1 Définitions et caractéristiques

Tout d'abord, nous allons introduire les définitions et concepts liés aux systèmes présentés dans le reste de cette partie.

Les réseaux overlays

Un réseau overlay (ou réseau de recouvrement) est un réseau logiquement construit « au-dessus » d'un autre : les noeuds du réseau overlay sont un sous-ensemble des noeuds du réseau sous-jacent et les liens de ce réseau sont constitués d'un chemin (c'est-à-dire d'une succession de liens et de noeuds) dans le réseau sous-jacent. La figure 2.4 illustre ce principe.

Les réseaux overlays sont très présents dans les réseaux. Par exemple, on peut considérer que le réseau IP est un réseau overlay construit au-dessus du réseau physique ou encore qu'une technologie telle qu'un Réseau Privé Virtuel IPSec[37] construit un réseau overlay au-dessus du réseau IP.

On peut remarquer une certaine analogie entre les réseaux overlays et la représentation en couches de l'architecture des réseaux (voir la section 2.2.2). En effet, on peut considérer que pour chaque couche de l'architecture du réseau, un réseau overlay est formé au-dessus de la couche de niveau inférieur.

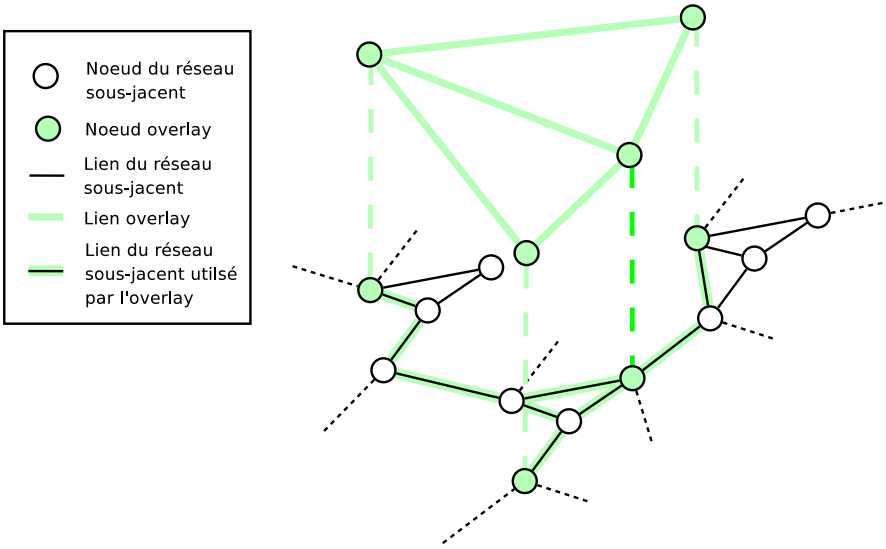


FIG. 2.4: Le réseau overlay et le réseau sous-jacent

Les réseaux pair-à-pair

Une architecture de réseau pair-à-pair (P2P) est une architecture de réseau informatique distribuée formée de plusieurs noeuds partageant certaines ressources afin de rendre un service. Contrairement à l'architecture clients/serveur, où les noeuds sont soit serveur, soit client, de la ressource, dans une architecture P2P, les noeuds sont à la fois client et serveur [87].

Ainsi, plusieurs notions distinctes définissent les réseaux P2P :

– L'architecture distribuée : les noeuds sont à la fois clients et serveurs. Le système peut ainsi fonctionner sans qu'aucune coordination centrale soit nécessaire. De plus, dans le cas des réseaux P2P « purs » [87] (qui est rarement implémenté en réalité), tous les noeuds du réseau sont équivalents et fonctionnent de manière identique dans le réseau.

– La mise en commun d'une ressource : les noeuds partagent entre eux une ressource matérielle (puissance de calcul, espace de stockage, bande passante réseau, etc.) de manière à ce que le réseau P2P délivre un service (partage de fichier, travail collaboratif, etc.). Cette notion implique que le plus souvent, les noeuds participant au réseau P2P sont « en bout » de réseau. En effet, ce ne sont pas les routeurs, qui disposent de peu de ressources matérielles, qui vont participer aux réseaux P2P, mais le plus souvent les machines des utilisateurs. Elles disposent d'importantes ressources matérielles et leurs utilisateurs sont prêts à les utiliser pour accéder à certains services réseau.

Un réseau P2P forme un réseau overlay. Bien que cela ne soit pas strictement obligatoire, nous considérerons dans la suite de ce document que les réseaux P2P sont des réseaux overlays construits au-dessus du réseau IP. Les noeuds du réseau overlay sont les noeuds participants au réseau P2P et sont un sous-ensemble des noeuds du réseau IP. Les liens du réseau overlay qui relient deux noeuds de ce réseau utilisent les routes qui relient ces noeuds dans le réseau IP.

Les réseaux P2P sont généralement utilisés pour permettre d'apporter un service aux utilisateurs que ne peuvent pas rendre les fournisseurs de services classiques (tels que les opérateurs de réseau ou les distributeurs de contenu), ou bien d'améliorer la délivrance de ce service. Par exemple, les services d'échanges P2P de fichiers se sont développés, car les fournisseurs de contenu ne pouvaient assurer

l'accès à ces fichiers (pour des raisons légales, mais aussi de coût). De même, certains services réseau tels que le multicast, n'étant pas implémentés dans les réseaux des opérateurs, ont été implémentés grâce aux réseaux P2P.

Routage applicatif

Nous allons maintenant introduire la notion de routage applicatif. Un système de routage applicatif réalise une opération de routage, c'est-à-dire qu'il est chargé d'assurer l'acheminement des communications jusqu'à leur destination en utilisant un chemin valide, en prenant en considération des informations de haut niveau, liées au service transporté par la communication. Le routage IP classique utilise uniquement les informations de niveau IP ou le plus souvent, l'adresse IP de destination seule, pour déterminer le chemin à utiliser. Le routage applicatif est parfois connu sous le terme « Layer 4-7 switching » [12].

En effet, avec le routage IP, c'est le plus souvent l'adresse IP de destination du paquet qui est utilisée pour réaliser les décisions de routage. Ceci implique que l'ensemble des paquets pour une même adresse IP de destination sera acheminé de la même façon, quelle que soit la nature des communications transportées. Cependant, il est parfois souhaitable que d'autres paramètres liés à la nature du service transporté ou aux besoins des utilisateurs soient considérés pour réaliser l'acheminement des communications. Le routage applicatif utilise donc d'autres informations, provenant par exemple des entêtes des protocoles de niveau transport ou application des paquets à transmettre, mais aussi d'informations externes, liées par exemple aux besoins d'un utilisateur, pour décider comment acheminer au mieux les communications.

Voici quelques exemples de situations où l'utilisation du routage applicatif peut s'avérer plus intéressante que le routage classique :

- Le fournisseur d'un service veut répartir les requêtes des utilisateurs entre ses différents serveurs délivrant ce service. Le routage applicatif permet d'acheminer une requête vers un de ces serveurs de manière transparente pour l'utilisateur : sa requête concerne le service à atteindre et c'est le système de routage applicatif qui détermine l'adresse du serveur à joindre.
- Dans Internet, les routes entre deux noeuds sont soumises aux accords passés entre les différents opérateurs des réseaux traversés pour joindre ces deux noeuds. Les performances réseau de ces routes peuvent ne pas être optimales : il est possible qu'une route transitant par un noeud tiers possède de meilleures performances que la route « directe » entre deux noeuds. Le routage applicatif permet de faire transiter le trafic par ces noeuds tiers et ainsi d'utiliser ces routes.
- Lorsqu'un incident affecte une communication, les mécanismes de rétablissement réseau déployés par un opérateur ne prennent pas en compte la criticité de celle-ci. L'utilisation d'un système de routage applicatif permet d'utiliser un mécanisme de rétablissement adapté aux spécificités d'une communication.

Après que la décision de routage est été effectué à partir des informations de haut niveau, le routage applicatif doit mettre en place un mécanisme pour la mise en oeuvre de l'acheminement désiré des communications dans le réseau IP. Pour cela, voici certains schémas de fonctionnement qu'il est possible d'utiliser pour implémenter ces mécanismes[70, 51] :

- La redirection DNS : ce mécanisme s'appuie sur la résolution de nom DNS : lorsqu'un utilisateur émet une requête afin de déterminer l'adresse IP associée à un nom de domaine, l'adresse IP qui lui est retournée est variable et correspond à une destination déterminée par des informations de haut niveau.
- La redirection transparente : les communications de l'utilisateur sont à destination d'une machine qui est chargée de les rediriger vers le serveur délivrant le service demandé.

– La redirection à postériori : l'utilisateur émet une requête vers une machine qui lui indique quel serveur contacter pour se voir délivrer le service demandé.
– L'acheminent par l'utilisateur : les communications sont directement modifiées par une application déployée chez l'utilisateur avant d'être émises dans le réseau.

Il faut noter qu'en fonction des informations utilisées pour prendre les décisions de routage et de l'implémentation du mécanisme d'acheminement des communications, un système de routage applicatif ne peut être fonctionnel que pour certains types de communication. Ainsi, si le système utilise des informations n'étant pas liées à une application particulière, et que son mécanisme d'acheminement peut être utilisé pour tout type de communication IP, il est probable qu'il puisse être utilisé pour tout type de communication. À l'inverse, si un système prend en compte des informations liées à la nature d'une application pour réaliser les décisions de routage ou met en oeuvre le mécanisme d'acheminement des communications, ce système ne fonctionnera que pour les communications liées à cette application. Ainsi, la plupart des systèmes de routage applicatif déployés aujourd'hui fonctionnent uniquement avec les communications Web qui utilisent le protocole HTTP.

Les systèmes utilisant le principe du routage applicatif ont des architectures variées. Le système peut être administré par une entité centrale ou être distribué, les différents éléments composant ce système peuvent avoir des rôles spécifiques ou équivalents, et peuvent être déployés à divers endroits du réseau. Nous allons présenter deux types de systèmes aux architectures distinctes : les Content Delivery Networks (CDN) et les systèmes de routage P2P (dans la section suivante).

Les CDN sont des architectures de système réseau qui permettent d'améliorer[28, 38] la délivrance de certains contenus à l'utilisateur. Pour cela, un système CDN réplique le contenu d'un serveur de contenu sur différents serveurs du réseau. Les CDN utilisent des techniques de routage applicatif de manière à rediriger les requêtes pour un certain contenu vers le serveur pouvant délivrer ce contenu de la façon la plus efficace possible : ce peut être le serveur le plus proche de l'utilisateur émettant la requête ou encore le serveur dont la charge est la plus faible, par exemple. Les CDN incluent aussi les mécanismes pour organiser la réplication du contenu en différents points du réseau, par exemple, mais ces techniques ne seront pas abordées ici.

Le routage P2P

Nous allons maintenant introduire la notion de routage P2P. Le routage P2P est une technique reprenant le principe du routage applicatif déployé dans un réseau overlay déployé au-dessus du réseau IP. Ce réseau peut ainsi être qualifié de P2P. Ainsi, un système de routage P2P est un réseau décentralisé de noeuds qui réalisent des décisions de routage et qui acheminent des communications, afin de proposer un acheminement des communications plus performant qu'il ne le serait avec le routage classique. On peut considérer que dans ce type de réseau, la ressource partagée entre les noeuds est la connectivité au réseau.

Ainsi, dans ce type de système, c'est la connectivité des noeuds au réseau qui va être utilisée pour améliorer l'acheminement des communications. En effet, entre chaque paire de noeuds du réseau de routage P2P, il existe une route IP. Puisque ces noeuds coopèrent entre eux, il va être possible qu'un utilisateur, pour joindre une destination quelconque du réseau, emprunte la route « classique » calculée par le routage IP, mais aussi une succession de routes entre chacun des noeuds du réseau P2P, jusqu'à la destination voulue. Le schéma 2.5 illustre ce concept.

Le choix de la route à utiliser entre les différentes routes possibles peut être ensuite fait en fonction de besoins variés, qui peuvent être adaptés aux attentes de l'utilisateur ou aux spécificités du service délivré.

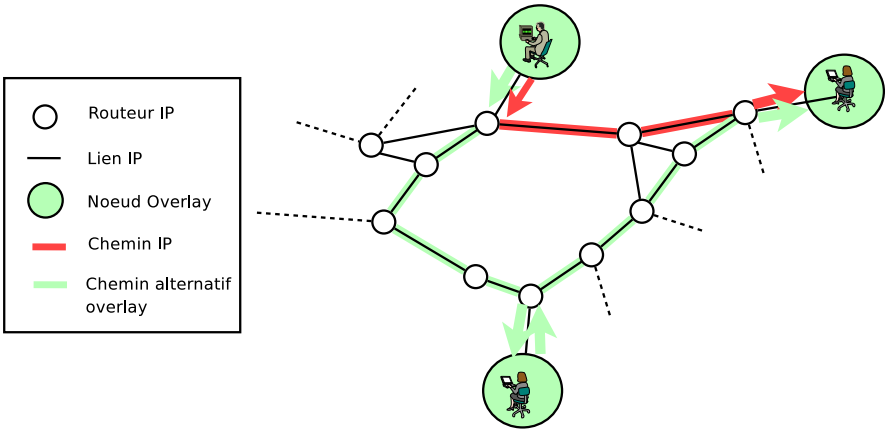


FIG. 2.5: Principe de fonctionnement du routage P2P

Motivation pour l'utilisation d'un système de routage P2P

Les motivations pour utiliser un système de routage P2P sont multiples.

Tout d'abord, comme illustré dans la figure 2.6, le routage P2P permet de proposer des routes permettant un acheminement de meilleure qualité des données[1, 50]. En effet, il arrive que la route calculée par le routage IP ne soit pas optimale en terme de performances réseau, comme le délai d'acheminement ou la bande passante disponible par exemple. C'est en particulier le cas pour les routes qui traversent plusieurs AS. En effet, ces routes, calculées par le protocole BGP, sont soumises aux accords commerciaux réalisés entre les opérateurs de réseaux et ne sont pas systématiquement les routes optimales en terme de performance réseau. Dans ce cas, l'utilisation du routage P2P va permettre de « court-circuiter » BGP en permettant d'emprunter des routes alternatives.

De plus, une dernière motivation à utiliser un système de routage P2P est d'accroitre la confidentialité des communications. En effet, lorsqu'une route issue du système de routage P2P est utilisée pour joindre une destination, les adresses IP des paquets transmis seront modifiées chaque fois que ceux-ci atteignent un des noeuds du réseau de routage P2P. Ainsi, il est impossible pour un tiers réalisant une écoute du trafic de connaitre l'origine et la destination réelles d'une communication en se basant uniquement sur l'entête IP des paquets. Cette technique est par exemple utilisée dans le système Freenet[10, 80], un réseau P2P destiné à assurer la liberté d'expression sur Internet.

Enfin, le routage P2P permet l'amélioration de la fiabilité des communications. En effet, comme illustré dans la figure 2.7, si un incident affecte la route du réseau IP utilisée pour l'acheminement d'une communication, celle-ci sera perturbée jusqu'à ce que le mécanisme de rétablissement de l'opérateur du réseau soit déclenché et permette l'utilisation d'une autre route afin de « contourner » l'incident ou que celui-ci soit réparé. Si la durée de ces opérations est trop longue, l'utilisation d'un système de routage P2P va éventuellement proposer d'emprunter une route alternative de manière à contourner l'incident le temps que celui-ci disparaisse. C'est cette utilisation du routage P2P qui nous intéresse en particulier.

2.5.2 Caractéristiques des systèmes de routage P2P

Nous allons voir dans cette partie les différents éléments permettant de caractériser un système de routage P2P. Ces différentes caractéristiques présentent des avantages et des inconvénients qui sont à prendre en considération lors de la conception d'un système de routage P2P, en fonction des objectifs

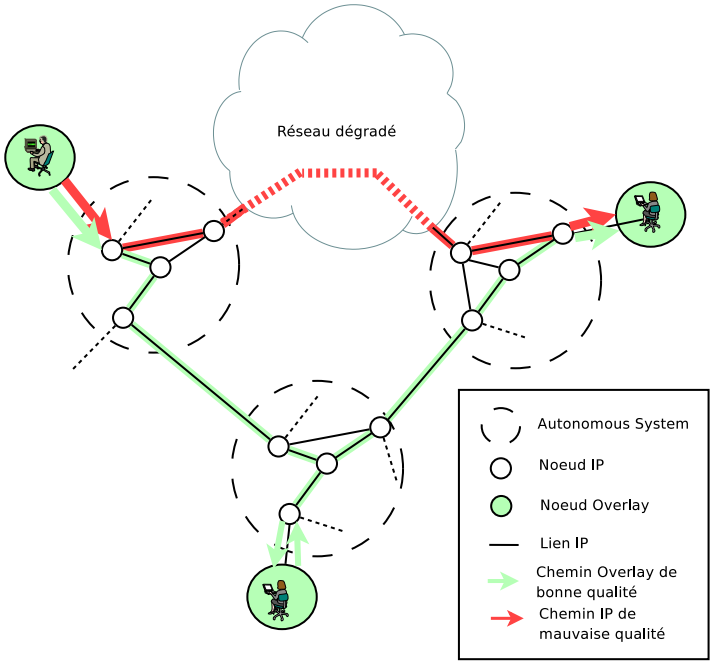


FIG. 2.6: Utilisation du routage P2P pour l'amélioration des performances

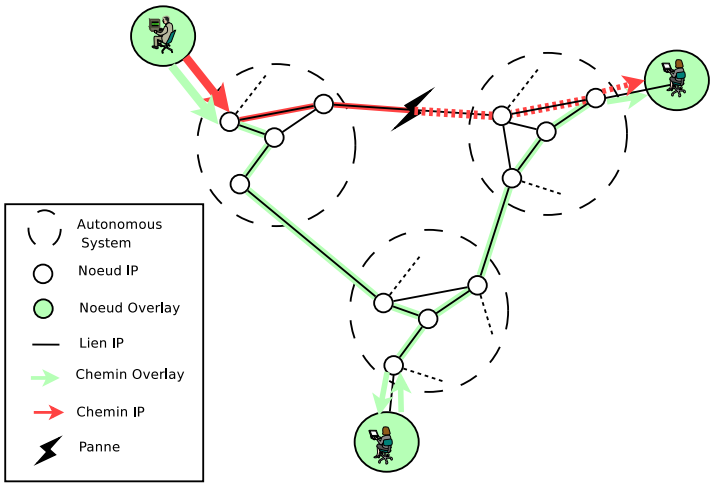


FIG. 2.7: Utilisation du routage P2P pour l'amélioration de la fiabilité

de celui-ci.

Mesure du réseau overlay

Comme pour un système de routage classique, les noeuds du système de routage P2P doivent mesurer le réseau overlay afin de connaitre l'état de celui-ci et ainsi prendre des décisions de routage. Pour cela, deux noeuds overlays mesurent le lien qui les relie en s'échangeant des messages qui sont acheminés normalement dans le réseau IP.

Le rôle premier de ces messages est de vérifier la connectivité entre deux noeuds. Ainsi, un noeud overlay sera informé que le lien qui le relie à un autre noeud peut être utilisé pour acheminer une communication. Cette mesure permet en particulier de détecter un incident qui affecterait les communications dans le réseau IP.

La mesure du réseau permet de plus de déterminer la qualité d'un lien en terme de performance réseau. Il est en effet possible de mesurer le délai d'acheminement aller-retour des communications lors de la mesure d'un lien overlay, mais aussi le taux de perte de paquet ainsi que d'estimer la bande passante disponible.

Le nombre de liens overlays mesurés ainsi que la fréquence de ces mesures peuvent être très variables en fonction des objectifs et des autres caractéristiques du système de routage P2P. En effet, en fonction du contexte d'utilisation du système de routage overlay :

- La mesure du réseau overlay peut être effectuée par chaque noeud du réseau overlay ou un nombre limité de noeuds.
- Le nombre de liens mesurés par chaque noeud peut être plus ou moins grand et peut même varier au cours du temps.
- La fréquence de ces mesures peut être plus ou moins grande, et peut aussi être variable en fonction de la situation.

En fonction de paramètres, le coût de la mesure en terme de consommation de bande passante réseau sera plus ou moins important. Ce coût ne doit pas être négligé, en particulier pour permettre le passage à l'échelle du système, lorsque le nombre de noeuds overlay est grand.

Utilisation systématique ou dédiée du routage P2P

Selon les objectifs du système de routage P2P, son utilisation peut être systématique ou alternative au routage classique. Ainsi, il est possible de ne prendre en charge une communication par le routage P2P que lorsque le routage classique ne satisfait plus à certains besoins.

Ainsi, en fonction de cette utilisation du système, le calcul de chemins overlays à emprunter peut être effectué de manière dynamique et systématique, en fonction des résultats de la mesure du réseau overlay. Dans ce cas, entre chaque noeud source et destination des communications prises en charge par le système de routage P2P, le chemin overlay utilisé pour l'acheminement est dynamiquement ajusté de manière à être le plus performant possible, selon les critères du système. À l'inverse, les chemins overlays peuvent être considérés comme des chemins alternatifs qui seront utilisés si le routage classique ne satisfait plus aux critères du système. Ces chemins peuvent être précalculés de manière à anticiper cette défaillance, mais ce n'est pas obligatoirement le cas.

Choix des chemins overlays

Les chemins overlays sont les chemins qui seront empruntés par les communications prises en charge par le système de routage P2P. Selon les objectifs du système de routage P2P, on peut choisir les chemins overlays en fonction de certaines propriétés.

Lorsque l'on souhaite améliorer les performances réseau des communications, le chemin overlay peut être choisi comme celui offrant les meilleures performances possible entre une source et une destination donnée. Pour cela, on utilise un algorithme de calcul du chemin de moindre coût selon une certaine métrique (le délai d'acheminement ou le taux de perte, par exemple) dans le graphe représentant le réseau overlay.

Lorsque l'on souhaite fiabiliser les communications en cas d'apparition d'un incident, une propriété pouvant être souhaitée pour les chemins overlays est d'être disjoints entre eux. Ainsi, lorsqu'un incident affecte une communication acheminée sur le chemin primaire, on cherche à disposer d'un chemin alternatif disjoint du chemin primaire de manière à ce que le risque que celui-ci soit aussi affecté par l'incident soit le plus faible possible.

Acheminement des communications par le routage P2P

Nous allons voir les différents mécanismes pouvant être employés pour permettre l'acheminement des communications dans le réseau overlay.

Tables de routage

L'utilisation de tables de routage, similaires à celles utilisées dans les routeurs IP, est une première possibilité. Les tables de routage, présentes dans chaque noeud participant au routage P2P, indiquent le noeud auquel transmettre une communication en fonction de la destination de celle-ci. Les communications sont ainsi acheminées de saut en saut au travers du réseau overlay. Les adresses utilisées dans les tables de routage peuvent être les adresses IP, bien que cela ne soit pas obligatoire : les noeuds peuvent être désignés par des adresses propres au réseau overlay.

Dans les systèmes de routage P2P qui utilisent des tables de routage, l'algorithme utilisé pour calculer les chemins du réseau est généralement l'algorithme à état de lien. Ceci implique que chaque noeud du réseau doit avoir une vision complète et similaire de la topologie de celui-ci. Pour cela, chaque noeud doit mesurer la connectivité avec chacun de ses voisins et transmettre ces informations aux autres noeuds du réseau, ce qui peut être couteux en terme d'utilisation des ressources réseau.

Routage par la source

Il est possible d'utiliser le routage par la source pour acheminer les communications dans le réseau overlay. Pour cela, il suffit que le noeud qui émet une communication dans le réseau overlay ajoute les adresses des noeuds overlays à successivement emprunter pour joindre la destination. Puisque ces adresses sont ajoutées à chaque paquet émis, ce procédé peut avoir un coût en ressource réseau important surtout si le nombre de noeuds overlay à traverser est important.

L'avantage de cette technique est que seul le noeud émetteur d'une communication a besoin de connaitre la topologie du réseau. De plus, cette connaissance peut-être incomplète, dans ce cas, les communications ne pourront être acheminées que par des noeuds overlays, dont le noeud source à la connaissance. Ainsi, la mesure du réseau overlay n'est effectuée que par le noeud source et n'est pas obligatoirement étendue à tous le réseau overlay.

Circuits virtuels

Pour acheminer les communications dans le réseau overlay, il est enfin possible d'utiliser des circuits virtuels, qui sont des chemins préétablis qui seront empruntés par les communications. Les circuits virtuels fonctionnent dans un réseau overlay de manière similaire à ce qui est fait dans MPLS.

L'utilisation de circuits virtuels permet de déterminer exactement la succession de noeuds overlays à traverser, comme avec le routage par la source, mais sans le besoin d'ajouter les adresses des noeuds à tous les paquets. Cependant, les circuits doivent être « ouverts » avant de pouvoir être utilisés pour acheminer une communication. Si une mesure du réseau overlay effectué par les noeuds, il est possible d'établir des circuits virtuels automatiquement, en fonction de ces mesures, mais l'établissement d'un circuit virtuel peut être effectuée de manière explicite par un noeud, en envoyant un message « d'ouverture » de circuit dans le réseau.

Topologie des réseaux overlays

Nous allons voir que la topologie du réseau overlay joue un rôle important dans les performances que l'on peut attendre de ce dernier. C'est en effet un paramètre essentiel à prendre en compte lors de la conception d'un système reposant sur un réseau overlay.

Définition

On appelle noeuds overlays l'ensemble des noeuds du réseau sous-jacent qui participent au réseau overlay. On appelle liens overlays l'ensemble des liens entre les différents noeuds overlays. La topologie du réseau overlay est l'ensemble des noeuds overlay et des liens overlays.

La topologie va donc définir quels sont les noeuds communiquant « directement » les uns avec les autres : ce sont ceux reliés par un lien. Pour permettre la communication entre deux noeuds n'étant pas reliés directement par un lien, des mécanismes de routage doivent être mis en place pour permettre l'acheminement des communications par des noeuds intermédiaires.

On peut remarquer que pour un système de routage P2P, la topologie désigne l'ensemble des liens et noeuds pouvant être empruntés pour acheminer les communications. Cependant, cette topologie n'est pas nécessairement respectée lors de la mesure de l'état du réseau ou de la transmission de messages liés au fonctionnement du système, par exemple.

Les différentes topologies

Nous allons présenter ici les différentes catégories de topologie rencontrées dans les réseaux overlays.

- La topologie « Full Mesh » : Dans cette topologie, il existe un lien entre chaque pair de noeud du réseau
- Les topologies en arbre : Il existe dans ces topologies un noeud racine d'où sont issus des liens vers des noeuds eux-mêmes racines d'autres noeuds. Cette topologie est couramment rencontrée dans les systèmes destinés à la diffusion de contenu[5], depuis un noeud unique vers plusieurs noeuds, ou mettant en place les communications multicast dans un réseau overlay[7].
- Les topologies structurées : Les topologies structurées désignent les topologies des réseaux overlays implémentant une table de hachage distribuée (DHT pour Distributed Hash Table). Les DHT servent à l'indexation de contenu réparti entre les différents noeuds d'un réseau P2P. Ce type de topologie permet de garantir que le nombre de sauts à effectuer dans le pire des cas pour acheminer un message de requête pour un contenu jusqu'au noeud responsable de son indexation soit de l'ordre de $O(ln(N))$, avec N le nombre de noeuds présents dans le réseau

overlay. Différentes topologies permettent ainsi d'implémenter ces systèmes. Parmi les plus connus, signalons les topologies « en anneau » utilisées par des systèmes tels que Chord[98], Pastry[85] ou Tapestry[110].

– Les topologies « Mesh » : Les topologies Mesh sont des topologies quelconques, où chaque noeud est relié par des liens overlays à un ensemble quelconque d'autres noeuds.

Incidence de la topologie du réseau overlay sur les performances

La topologie du réseau overlay a une influence sur les performances du système qui l'utilise[46]. En effet, si une topologie overlay est « peu connectée », lorsque la probabilité de présence d'un lien overlay entre deux noeuds choisis aléatoirement est faible, ou « fortement connectée », lorsque cette probabilité est élevée, les performances du système seront différentes.

Par exemple, le nombre de noeuds par lesquels transite une communication entre deux noeuds overlays quelconques sera plus important si la topologie est faiblement connectée que si elle l'est fortement. Ceci peut par exemple entraîner un délai de livraison des communications plus ou moins important.

À l'inverse, une topologie fortement connectée peut amener à une plus forte consommation de ressources. Par exemple, si un noeud a besoin d'interroger périodiquement ces voisins pour vérifier qu'aucun incident n'affecte le lien overlay qui les relie, la consommation globale de ressource pour effectuer cette tâche dépendra du nombre de liens overlay dans le réseau et par conséquent sera plus grande avec une topologie fortement connectée.

Pour le cas particulier des systèmes de routage P2P, la topologie du réseau overlay est déterminante dans la capacité du système à rétablir une communication affectée par un incident. En effet, pour que cela soit possible, il est nécessaire qu'un chemin alternatif contournant l'incident existe dans cette topologie.

Incidence de la topologie du réseau sous-jacent sur les performances

Le rapport entre la topologie d'un réseau overlay et celle du réseau sur lequel il est construit a une grande influence sur ses performances. En effet, en fonction de façon dont un réseau overlay est construit, sa topologie peut être très différente de la topologie du réseau sous-jacent.

Par exemple, deux noeuds « proches », distants de quelques sauts dans le réseau overlay peuvent être très éloignés dans le réseau sous-jacent. De même, il est possible que deux liens distincts dans le réseau overlay empruntent un ou plusieurs mêmes liens du réseau sous-jacent.

Ceci peut avoir un impact sur les performances, par exemple :

– Les efforts visant à limiter le délai de livraison des communications dans le réseau overlay en limitant le nombre de noeuds par qui les faire transiter peuvent être réduits à néant si la distance entre ces noeuds dans le réseau sous-jacent est trop importante, car dans ce cas le délai d'acheminement des communications entre ces noeuds sera trop grand.
– Les ressources du réseau sont gaspillées lorsque plusieurs liens overlays utilisent le même lien du réseau sous-jacent, car un même message qui sera diffusé sur des liens overlays distincts sera en réalité diffusé plusieurs fois sur le même lien physique et l'on peut considérer que ceci aurait pu être évité en utilisant une topologie de réseau plus adaptée.

Pour le cas particulier du routage P2P, la topologie du réseau sous-jacent va aussi influer sur la capacité du système à rétablir une communication. En effet, on peut être tenté d'utiliser pour ce type de système une topologie overlay fortement connectée, afin de garantir que si un lien overlay est

affecté par un incident, on puisse facilement trouver un chemin alternatif dans le réseau overlay qui le contourne. Cependant, plus le nombre de liens overlay est important et plus la probabilité que des liens overlays utilisent un même lien du réseau sous-jacent est importante. Ainsi, lorsqu'un incident affecte un lien du réseau sous-jacent, l'ensemble des liens overlay qui l'utilisent seront affectés et ne pourront pas être utilisés par le chemin alternatif pour contourner l'incident. Par conséquent, en fonction de la topologie du réseau sous-jacent, l'utilisation d'un nombre important de liens dans le réseau overlay ne garantit pas une plus grande capacité d'un système de routage P2P à rétablir une communication. Nous reviendrons dans le chapitre 5 sur les conditions nécessaires pour permettre le rétablissement d'une communication par un système de routage P2P.

Différents travaux ont proposé des solutions pour améliorer le placement des noeuds dans le réseau overlay lors de sa construction, afin qu'il corresponde mieux au placement des noeuds dans le réseau IP. Certains[82, 68] de ces travaux se basent sur le délai d'acheminement des communications pour réaliser une approximation de la distance entre deux noeuds dans le réseau IP. D'autres[105] découvrent la topologie du réseau sous-jacent en se basant sur la technique du « traceroute ».

2.5.3 Systèmes de routage P2P existants

Nous allons maintenant discuter des différents systèmes de routage P2P qui ont été présentés dans différents travaux académiques. Nous expliquerons leurs principales caractéristiques et nous discuterons de leurs avantages et de leurs inconvénients.

Le système Detour

Le système Detour[14] est probablement le premier système apparenté au routage P2P. Il est né après avoir constaté que les problèmes affectant les routes dans Internet étaient localisés, et par conséquent pouvaient être contournés. À l'aide d'une analyse des connexions entre 43 noeuds, il est en effet montré que dans environ 1 cas sur 3, il existe une route alternative, transitant par un ou plusieurs noeuds étudiés, dont le délai d'acheminement aller-retour des communications est significativement meilleur que celui obtenu en empruntant la route calculée par le routage classique.

Afin d'exploiter ces routes alternatives, il faut « détourner » les communications et les faire transiter par un ou plusieurs noeuds intermédiaires. Le système Detour propose donc de mettre en place des tunnels virtuels entre le noeud source et le premier noeud intermédiaire, éventuellement entre les noeuds intermédiaires successifs, et enfin entre le dernier noeud intermédiaire et le noeud destination. Les communications entre le noeud source et le noeud destination transitent ensuite par ces tunnels.

Pour mettre en place un tel système, les paquets créés par l'application exécutée sur le noeud source sont interceptés par le système Detour. L'entête IP ainsi que les données des protocoles des couches supérieures sont conservés, et encapsulés par un entête Detour. L'entête Detour contient les adresses de tous les noeuds intermédiaires à traverser pour joindre la destination. Ces données sont ensuite émises vers le premier noeud intermédiaire. Pour cela les données sont incluses dans un paquet IP classique dont l'adresse source est l'adresse du noeud source et l'adresse destination est l'adresse du premier noeud intermédiaire. Lorsque ce paquet est reçu par le noeud intermédiaire, il consulte l'entête Detour afin de connaitre le prochain noeud intermédiaire auquel envoyer les données initiales accompagnées de l'entête Détour. Les communications transitent ainsi de proche en proche jusqu'au dernier noeud destination. Sur celui-ci, le système Detour « décapsule » le paquet reçu afin de retrouver le paquet IP initial et que celui-ci soit traité comme n'importe quel autre paquet IP reçu. Ainsi, l'utilisation du système Detour est transparente pour l'application, aussi bien chez le noeud source que chez le noeud destination.

Le but des auteurs de Detour était d'intégrer leur système aux mécanismes de routage existants, afin d'améliorer la qualité des routes proposées par le routage IP. Ainsi, les auteurs de Detour ne font qu'évoquer la problématique d'auto organisation des noeuds participants, ainsi que le calcul et la mise en place des meilleurs tunnels à utiliser. Ces points seront adressés par le système RON.

Le système RON

Le système Resilient Overlay Network[1] (RON) a été proposé dans le but d'apporter une solution à certains problèmes du routage classique soulevés précédemment. Pour cela, des nœuds RON, placés dans le réseau, coopèrent afin d'acheminer au mieux les communications. Pour cela, elles empruntent un chemin composé de sauts successifs de nœuds RON.

Les nœuds RON s'échangent ainsi en permanence des informations sur la qualité des chemins les reliant et construisent une table de routage en fonction de plusieurs métriques telles que le délai, le taux de perte, la bande passante disponible. Ces informations sont recueillies par l'envoi de messages sondes sur les chemins entre les différents nœuds. L'utilisation d'un probing agressif permet de détecter rapidement un changement de topologie. Ceci se fait au détriment d'un surcoût d'utilisation de bande passante. De plus, l'utilisation de métriques variées permet de faire un choix plus fin des routes à emprunter, en fonction des besoins de l'application.

Le routage de RON fonctionne selon un algorithme à état de lien. Dans un réseau à N nœuds RON, chaque nœud a $N-1$ liens virtuels avec les autres noeuds. Chaque nœud effectue périodiquement une requête sur les $N-1$ liens virtuels afin d'évaluer les performances des liens. Ces informations sont ensuite propagées à l'ensemble du réseau. Chaque noeud est ensuite en mesure de calculer le chemin de moindre coût, selon les différentes métriques utilisées, vers chacun des noeuds RON de destination.

D'une manière similaire à Detour, pour permettre l'acheminement d'une communication dans le réseau RON, le premier noeud RON encapsule les données à acheminer. Quand un paquet arrive dans un nœud RON, s'il n'est pas le destinataire du paquet (auquel cas les données sont décapsulées afin de retrouver le paquet initial), il consulte sa table de routage pour connaitre le prochain nœud RON vers qui faire transiter le paquet afin de joindre la destination.

Évaluation des performances de RON

Les auteurs de RON ont réalisé une évaluation de ses performances. En utilisant les paramètres de détection par défaut, un noeud RON est capable de détecter un incident et, si possible, de rétablir une communication, en un temps moyen de 19 secondes. Avec de tels paramètres, la bande passante réseau consommée par chaque noeud pour faire fonctionner RON est de 30 kbit/s, pour un réseau de 50 noeuds RON.

L'utilisation de RON permet une amélioration des performances des routes utilisées par les communications. C'est particulièrement le cas lorsqu'une route calculée par le routage classique est de mauvaise qualité. En effet, avec un réseau de 12 noeuds RON, 5 % des routes calculées par RON et utilisées à la place de la route IP classique ont permis une diminution de 5 % du taux de perte des paquets, 11 % ont permis une diminution d'au moins 40 ms du délai de livraison aller-retour des paquets, et 5 % ont permis de doubler la bande passante disponible.

Critiques à l'encontre de RON

Le système RON a été critiqué car il ne passe pas à l'échelle : son coût de fonctionnement est très important, en particulier lorsqu'un trop grand nombre de noeuds RON sont présents dans le réseau.

En effet, la topologie utilisée par RON est de type Full Mesh, car il existe un lien overlay entre chaque pair de noeuds RON. Chaque noeud RON effectue périodiquement une mesure active, par envoi de messages sondes, de chacun des liens overlay auxquels il est reliés et propage ensuite ces informations à l'ensemble des noeuds du réseau. Si N est le nombre de noeuds RON dans le réseau, il y a ainsi $N.(N-1)$ liens à mesurer. La consommation des ressources du réseau est alors très importante si N est grand. Afin d'améliorer le passage à l'échelle de RON, il a été proposé[94] de limiter la propagation des messages d'information sur l'état des liens overlay à un sous ensemble de noeuds. Cette solution n'altère pas la capacité de RON à trouver le chemin de moindre coût dans le réseau overlay.

Une autre critique pouvant être faite à un système tel que RON est sa complexité par rapport aux objectifs à atteindre. En effet, l'utilisation de tables de routage permet d'établir des routes traversant un nombre arbitraire de noeuds RON, en fonction de la route de moindre coût calculée par l'algorithme à état de lien. Cependant, il est démontré (voir la section 2.5.3 plus bas) que dans la plupart des cas, cette route de moindre coût transite par un seul noeud RON. Par conséquent, l'utilisation d'un algorithme à état de lien parait peu adaptée, car elle amène un important coût de fonctionnement pour le calcul de ces routes « simples ».

Le système NATRON

Le système NATRON[60] est une amélioration de RON destinée à permettre l'utilisation de ce système pour joindre des hôtes n'étant pas eux-mêmes membres du réseau P2P. Pour cela, NATRON introduit un mécanisme de translation d'adresse.

Un des problèmes des systèmes de routage P2P est la mesure de la performance des routes entre le noeud source, qui initie une communication, et le noeud destination, qui n'est pas membre du réseau P2P. En effet, le système de routage P2P ne sait pas à priori quel sera le noeud destination avant que la communication ne soit initiée par le noeud source. Par conséquent, le système n'est pas informé de la qualité des différentes routes utilisables au moment où la communication débute, et ne peut pas immédiatement utiliser la route de meilleure qualité.

La méthode utilisée dans NATRON pour sélectionner la meilleure route est basée sur le délai d'acheminement des premiers paquets échangés à l'établissement d'une connexion TCP (le paquet SYN et sa réponse SYN ACK). Lors de l'initialisation de la connexion, le premier paquet est envoyé sur la route proposée par le routage classique. Si au bout d'un temps T, la réponse n'est pas parvenue, le noeud source détermine les 3 « meilleurs » noeuds du réseau P2P et réémet le paquet sur les 3 routes transitant par ces 3 noeuds du réseau avant de joindre la destination. Les 3 meilleurs noeuds sont les 3 noeuds parmi ceux du réseau P2P dont le lien overlay avec le noeud source est de meilleure qualité. La qualité de ce lien est connue en permanence puisque ces noeuds font tous parties du réseau P2P et échangent en permanence des informations sur la qualité des liens qui les relient, selon une certaine métrique, d'une manière semblable à ce qui a été décrit pour RON dans la section 2.5.3.

La conception du système NATRON est similaire à celle de RON à l'exception de spécificités chez le noeud source et le noeud « intermédiaire », par lequel transite une communication si la route reliant la source à la destination via ce noeud est utilisée. Chez le noeud source, un mécanisme de surveillance de l'émission d'un paquet d'initiation d'une connexion TCP doit être mis en place afin d'éventuellement réémettre ce paquet vers une route proposée par le réseau overlay si la réponse par la route classique n'est pas reçue dans le temps imparti. Chez le noeud intermédiaire, un mécanisme de translation d'adresse[95] (NAT) classique doit être mis en place afin de permettre la communication avec la destination de manière transparente pour celle-ci : en effet de son point de vue, la communication à lieu avec le noeud intermédiaire. C'est ce dernier qui modifie les paquets qui transitent par lui

afin de rendre l'utilisation d'une route overlay invisible pour la destination.

L'évaluation de NATRON montre que ce système permet une diminution des temps de téléchargement sur un serveur lorsqu'il est utilisé, en particulier dans les cas « problématiques », où la route classique est de mauvaise qualité. Cependant, il semble que la politique de sélection de la meilleure route de NATRON soit perfectible : en effet, dans 30 % des situations, l'utilisation d'une route overlay améliorerait les performances d'accès au serveur. Cependant, lorsque cette situation se produit, c'est seulement 1 fois sur 5 que NATRON sélectionne effectivement cette route.

Internet Indirection Infrastructure

Internet Indirection Infrastructure[97] (I3) est un système basé sur le routage dans un overlay à l'aide de points de rendez-vous. Dans ce système, les noeuds fournisseurs d'un service proposent à l'utilisateur de « poser un déclencheur » pour accéder à un service. Pour cela, l'utilisateur envoie un message particulier qui va établir un chemin et permettre l'acheminement de ses communications à travers le réseau overlay jusqu'au fournisseur de service. Ce procédé permet par exemple à I3 d'automatiser :

- La composition des services : L'utilisateur peut en effet poser une succession de déclencheurs de façon à ce que les communications soient acheminées chez différents fournisseurs de service, afin que ceux-ci délivrent un service composé.
- La mobilité des utilisateurs lors de l'accès au service
- Le multicast

Ce système, bien que n'étant pas adapté au rétablissement rapide des connexions, est intéressant, car il propose une alternative à la topologie Full Mesh utilisée dans le réseau RON en utilisant une topologie structurée.

Routage P2P sur réseaux structurés

L'utilisation du réseau structuré Tapestry comme réseau overlay utilisé par le routage P2P des communications a été étudiée [109]. Tapestry dispose en effet de plusieurs avantages qui justifient son utilisation comme base d'un système de routage P2P :

- Il dispose de mécanismes de détection des incidents affectant ses liens overlays.
- Si un incident affecte un lien overlay, un noeud de Tapestry dispose pour chacun d'entre eux d'un lien « de secours » vers un autre noeud du réseau overlay, de manière à préserver la connectivité des noeuds dans la topologie.
- Tapestry mesure le délai d'acheminement entre les noeuds de son réseau et adapte sa topologie en fonction du résultat de ces mesures, de manière à ce qu'elle soit cohérente avec la topologie du réseau IP.

Les auteurs de cette étude proposent d'utiliser Tapestry pour transporter les communications, de manière à bénéficier de ces atouts pour augmenter la fiabilité des communications en cas d'incident dans le réseau. Ce système s'avère être un mécanisme de rétablissement réseau efficace : les auteurs montrent ainsi qu'il est possible de rétablir une communication affectée par un incident en 700 ms, et que la bande passante consommée pour le fonctionnement du système reste faible. De plus, les propriétés d'autoconfiguration du réseau Tapestry permettent de maintenir une disponibilité élevée du système, même en cas d'incidents multiples. Enfin, le délai supplémentaire de délivrance des communications entraîné par leur acheminement par le réseau Pastry reste limité à moins de 10 %.

Nous pensons cependant que ce système n'est pas adapté au rétablissement rapide des communications. En effet, le temps de rétablissement indiqué ne tient pas compte du délai de réacheminement

des données, délai important dans Tapestry puisqu'un nombre de sauts importants doit parfois être effectué pour joindre un noeud destination. De plus, ce système ne prend pas en compte les besoins particuliers en fiabilité des services et des utilisateurs.

Routage Source à Un Saut

Afin d'améliorer le rétablissement des communications en cas d'incident dans le réseau, la technique du Routage Source à Un Saut (One Hop Source Routing) est examinée dans une étude[27]. Le principe du Routage Source à Un Saut, est, à la manière de ce qui est proposé dans le système Detour, de faire transiter une communication par un noeud tiers pour joindre une destination lorsque la communication est interrompue par un incident affectant la route utilisée avec le routage classique.

Les auteurs constatent que si l'on dispose de 40 noeuds pouvant servir d'intermédiaires, le Routage Source à Un Saut permet de contourner un incident et de maintenir une communication en la faisant transiter par un de ces noeuds dans 56 % des cas observés. Les auteurs émettent l'hypothèse que dans les autres cas, l'incident est situé trop près de la source ou de la destination pour pouvoir être contourné. Ils constatent de plus qu'en moyenne, lorsqu'un incident se produit, la moitié des noeuds intermédiaires peuvent être utilisés pour maintenir la communication. Ainsi, les auteurs montrent que se limiter 4 noeuds intermédiaires par qui faire transiter une communication affectée par incident suffit à maintenir une probabilité de rétablissement satisfaisante.

Path Probing Relay Routing

Un système basé sur la technique du Path Probing Relay Routing (PPRR, ou Routage Relayé par Mesure de Chemin) a été proposé[6]. Cette technique propose un algorithme de mesure des chemins alternatifs à emprunter en cas de défaillance de la route principale utilisée pour acheminer les communications entre un noeud source et un noeud destination. Les auteurs de ce système déduisent des études précédemment effectuées qu'il n'est pas utile d'utiliser un système de routage complexe dans un réseau overlay et puisqu'il suffit généralement de faire transiter une communication par un seul noeud tiers pour contourner un incident affectant le chemin « direct », calculé par le routage classique.

Les auteurs proposent donc un algorithme pour déterminer un ensemble de « meilleurs » noeuds intermédiaires parmi l'ensemble des noeuds participants au système de routage P2P. C'est parmi cet ensemble que le noeud relai par qui faire transiter les communications si un incident affecte le chemin principal sera choisi. Seuls les chemins transitant par ces noeuds sélectionnés sont mesurés, afin d'économiser les ressources du réseau en ne mesurant que les meilleurs chemins. La métrique utilisée pour évaluer la qualité d'un chemin est le délai d'acheminement aller-retour. Pour sélectionner les noeuds membres de l'ensemble des meilleurs noeuds, $2.k$ noeuds sont sélectionnés aléatoirement parmi l'ensemble des noeuds overlay et la qualité du chemin vers la destination est mesurée. Les noeuds via lesquels les k meilleurs chemins ont été mesurés sont inclus dans l'ensemble des meilleurs noeuds. Ensuite et à chaque étape, k nouveaux noeuds sont choisis aléatoirement et les $2.k$ chemins transitant par ces noeuds et par les noeuds de l'ensemble des meilleurs noeuds sont mesurés. Les noeuds des k meilleurs chemins mesurés forment le nouvel ensemble des meilleurs noeuds.

Les résultats de simulations du comportement de PPRR montrent que la probabilité de trouver un chemin alternatif viable lors de l'apparition d'un incident affectant le chemin principal est, avec PPRR, très proche de celle mesurée pour RON. Les auteurs mesurent que la plus haute probabilité est obtenue lorsque k, le cardinal de l'ensemble des meilleurs noeuds, vaut 16. Cependant, aucune mesure de la vitesse de rétablissement d'une communication n'est effectuée.

PeerWise

PeerWise[50] est un système de routage P2P dont le but est de diminuer le délai d'acheminement des communications. Le routage dans PeerWise utilise lui aussi un unique noeud tiers par lequel faire transiter les communications lorsque le chemin Internet n'est pas satisfaisant. En effet, il existe parfois dans Internet des violations de l'inégalité triangulaire[111]. Dans ce cas, le délai d'acheminement entre deux noeuds A et B soit plus important quel la somme des délais entre les noeuds A et C et C et B. Peerwise propose alors de faire transiter les communications entre A et B par le noeud C. Afin de favoriser le partage équitable des ressources réseaux entre les participants au système, Peerwise utilise le principe « d'avantage mutuel » : un noeud A ne peut faire transiter ses communications par un noeud C pour une destination B que si pour ce dernier il existe une destination D pour laquelle le délai d'acheminement d'une communication sera plus faible en la faisant transiter par le noeud A.

Pour permettre le passage à l'échelle, la mesure active des délais d'acheminement entre l'ensemble des noeuds, telle qu'effectuée dans RON, n'est pas envisageable. PeerWise ne réalise les mesures qu'entre les noeuds qui sont le plus susceptibles de faire parti d'une violation de l'inégalité triangulaire. Pour les déterminer, chaque noeud du système calcule des coordonnées réseau, à l'aide de l'algorithme Vivaldi[15]. Ce système utilise la mesure des délais d'acheminement entre un sous ensemble de noeuds et grâce à la triangulation, détermine les coordonnées de chaque noeud dans un plan. Par conséquent, dans ce système, les coordonnées des noeuds respectent toutes l'inégalité triangulaire. Chaque noeud PeerWise mesure ensuite les délais d'acheminement vers un sous-ensemble d'autres noeuds. Si la différence entre le résultat d'une mesure et le délai prédit par les coordonnées calculées est grande, Peerwise considère que la probabilité que ces noeuds soient impliqués dans une violation d'inégalité triangulaire est importante. Chaque noeud maintient ainsi une liste de noeuds « candidats », susceptibles de diminuer le délai d'acheminement de leurs communications vers une destination en les faisant transiter par l'un d'entre eux. La vérification de la présence d'une inégalité triangulaire est ensuite vérifiée en mesurant le délai d'acheminement entre ces noeuds candidats et vers un noeud destination donné.

Cette architecture permet à Peerwise de passer à l'échelle même en présence d'un important nombre de noeuds. De plus, il est montré que l'utilisation du principe d'avantage mutuel n'est que peu nuisible au fonctionnement du système, car la plupart des noeuds peuvent aider à améliorer le délai d'acheminement des communications des autres noeuds et très peu n'ont pas besoin d'autres noeuds pour améliorer ce délai. Peerwise permet une diminution du délai d'acheminement moyen dans le réseau d'au moins 10% et en moyenne d'environ 25%. Cependant, ce système ne permet pas de rétablir rapidement une communication (cela ne fait pas partie de ces objectifs). Par exemple, l'établissement d'un chemin overlay s'effectue par négociation entre les noeuds, afin de vérifier l'avantage mutuel, ce qui n'est pas approprié lorsque l'acheminement doit être rétabli au plus vite.

Chapitre 3

La détection d’incidents par envoi de messages sondes

3.1 Introduction

Comme nous l'avons décrit dans la section 2.3, les communications dans les réseaux IP sont en permanence susceptibles d'être affectées par des incidents qui vont entraîner leurs interruptions. Ceci pose problème lors de certaines utilisations sensibles de ces réseaux nécessitant de garantir une grande fiabilité lors de la délivrance des communications. Cependant, il n'est en général pas possible de prévoir ou de se prémunir totalement contre l'apparition d'incident durant une session de communication.

Lorsqu'un incident apparaît et affecte un des éléments nécessaires au bon fonctionnement d'une communication, il est nécessaire qu'il soit détecté, pour permettre le déclenchement d'une réaction appropriée. En particulier, dans le cas des mécanismes de rétablissement réseau, si un incident affecte un équipement tel qu'un routeur, utilisé pour l'acheminement d'une communication, il est nécessaire de détecter cette défaillance afin de permettre la redirection du trafic d'une communication vers un chemin du réseau non affecté par l'incident. De plus, dans le contexte de fiabilité des communications sensibles de ce document, il est particulièrement important de détecter les incidents de manière précise, en un temps maximum déterminé, de manière à alerter le système de rétablissement réseau au plus vite et ainsi permettre le rétablissement des communications en un temps le plus faible possible.

C'est pourquoi dans ce chapitre, nous allons nous intéresser à ces mécanismes de détection d'un incident. Plus particulièrement, nous nous intéresserons aux mécanismes permettant la détection des incidents entrainant l'impossibilité pour deux correspondants de communiquer entre eux. Nous étudierons ainsi dans quelles conditions ces mécanismes sont en mesure de détecter un incident en un temps donné. Nous nous intéresserons aux mécanismes qui effectuent une interrogation active du correspondant distant, grâce à l'envoi de messages sondes, afin de vérifier qu'il n'y a pas d'incident sur le chemin emprunté dans le réseau par le trafic de la communication. Nous étudierons différents modes de fonctionnement pour ces mécanismes.

La méthode de détection d'incident par envoi de messages sondes nous parait être la plus appropriée, car c'est la seule à pouvoir être appliquée à tout type d'infrastructure : elle ne dépend pas du matériel utilisé et elle aussi indépendante de la communication, car elle ne repose pas sur le trafic échangé pour permettre la détection de l'incident. Ce type de mécanisme est largement utilisé dans divers systèmes déployés dans les réseaux IP. C'est le cas par exemple des protocoles de routage tel que OSPF, qui utilisent l'échange de messages HELLO pour s'assurer de la connectivité entre routeurs voisins. Dans d'autres cas, tels que de nombreux systèmes P2P, la vérification de la connectivité est effectuée par interrogation du correspondant.

Dans cette étude, nous tenterons de résoudre une des problématiques lors de l'utilisation de ce type de mécanisme : la configuration de leurs paramètres. Il s'agit d'adapter le fonctionnement des mécanismes, et en particulier la fréquence d'envoi des messages sondes en fonction du temps de détection désiré, de manière à obtenir les meilleures performances, en terme d'utilisation de la bande passante réseau ou du risque d'apparition de faux positifs, par exemple. De plus, nous présenterons les performances des différents mécanismes de détection d'incident par envoi de messages sondes étudiés dans ce chapitre. Nous verrons ainsi dans quelles conditions ces mécanismes peuvent détecter un incident en un temps maximum, exprimé par leurs utilisateurs en fonction de leurs besoins en fiabilité. Les résultats de cette étude seront obtenus par simulation, puis validés par des expérimentations dans un vrai réseau.

Les contributions de ces travaux sont les suivantes : nous présenterons les différents modèles de fonctionnement des mécanismes de détection d'incident par envoi de messages sondes et nous introduirons un nouveau modèle de fonctionnement. Nous proposerons ensuite une modélisation d'un réseau et des différents mécanismes afin de simuler leurs comportements. Ces simulations seront validées par des mesures expérimentales. Nous déterminerons ainsi quels sont les meilleurs paramètres

à utiliser pour chacun des mécanismes. Enfin, nous étudierons et comparons les performances des mécanismes en fonction des besoins des utilisateurs exprimés par un temps maximum de présence d'un incident non détecté toléré.

Les résultats de cette étude pourront être utilisés par une grande variété de systèmes. De nombreux systèmes utilisent en effet l'envoi de messages pour vérifier la connectivité entre deux noeuds d'un réseau. C'est le cas par exemple des systèmes de supervision réseau, ou encore, des systèmes de routage P2P. Nous présenterons notamment un système de ce type dans le chapitre suivant.

Les sections composant ce chapitre sont les suivantes : dans un premier temps, nous allons nous intéresser au contexte de ces travaux, ainsi qu'aux travaux de recherche relatifs à ce sujet. Nous présenterons ensuite la modélisation des mécanismes de détection et du réseau utilisés pour obtenir nos résultats. Nous étudierons ensuite quels sont les paramètres de configuration à utiliser pour chaque mécanisme. Au chapitre suivant, nous présenterons les résultats des performances des différents mécanismes, et nous les comparerons entre eux. Nous conclurons ce chapitre dans une dernière section.

3.2 Contexte et travaux apparentés

Nous allons maintenant introduire le contexte nécessaire à la bonne compréhension des travaux présentés ici. Nous présenterons tout d'abord les différents mécanismes de détection d'incident, puis nous décrierons les différents modèles de fonctionnement utilisés par les mécanismes de détection par envoi de messages sondes, que nous étudions ici. Enfin, nous présenterons les différents travaux académiques en relation avec notre sujet.

3.2.1 Les mécanismes de détection d'incident

Nous allons voir qu'il existe plusieurs approches pour détecter un incident.

La première approche est la notification externe d'un incident. Dans cette approche, le mécanisme de détection d'incident est externe au mécanisme de rétablissement réseau et utilise un mécanisme spécifique pour détecter l'incident. Lorsqu'un incident est détecté, le mécanisme de détection informe le mécanisme de rétablissement qu'un incident s'est produit. Nous considérons que les mécanismes de « notification d'un incident par un protocole de couche inférieure » [86, 89] font partis de cette approche. Dans ce type de mécanisme, si un protocole de couche inférieure à la couche sur laquelle est déployé le mécanisme de rétablissement protocole détecte un incident réseau, il peut « remonter » cette information au mécanisme de rétablissement pour que ce dernier soit directement informé de la présence d'un incident. Par exemple, cette approche est utilisée lorsqu'un le protocole de niveau physique détecte la rupture d'un lien et qu'il informe le protocole de routage de niveau IP qu'un incident est survenu sur ce lien.

L'autre approche généralement utilisée est l'envoi de messages sondes. Il s'agit d'effectuer une interrogation active du correspondant distant, en lui envoyant un message spécifique, afin de vérifier qu'il n'y a pas d'incident sur le chemin emprunté par le trafic dans le réseau. Contrairement à l'approche précédente, cette méthode de détection d'incident est incluse dans le mécanisme de détection d'incident et ne dépend de l'infrastructure sur lequel le mécanisme est déployé. Nous utiliserons cette méthode dans nos travaux et nous allons donc détailler son fonctionnement et les travaux de recherche qui la concernent dans la suite de ce chapitre.

Nous allons nous concentrer sur les mécanismes de détection de type point à point. Ce type de mécanisme est déployé entre deux noeuds du réseau afin de permettre la détection par un des noeuds de l'apparition d'un incident affectant les communications vers l'autre noeud. Nous n'aborderons pas

les mécanismes de détection d'incident dits distribués, qui sont déployés sur un ensemble de noeuds et qui permettent par exemple qu'un noeud A soit informé par un noeud O qu'un incident affectant les communications vers un noeud B est survenu. Ce choix est motivé par le fait que nous nous focalisons sur la rapidité de la détection des incidents affectant une communication donnée et pas la détection d'incident au sein d'un système de plusieurs noeuds.

Ce type de mécanisme est largement utilisé dans divers systèmes déployés dans les réseaux IP. C'est le cas, par exemple, des protocoles de routage, tel que OSPF[62], où un routeur transmet à chacun de ses routeurs voisins (auxquels il est directement relié) un message HELLO afin que ces derniers sachent que les communications avec ce routeur sont possibles. Un routeur qui n'aurait pas reçu de messages HELLO de son voisin pendant un temps déterminé, considère qu'un incident est apparu entre lui et son voisin. Une variante de cette méthode de détection consiste en l'interrogation explicite de son correspondant. Dans ce cas, un noeud du réseau envoie un message d'interrogation à son correspondant qui lui retourne immédiatement un message de réponse. En cas d'absences consécutives d'un certain nombre de messages de réponse aux messages d'interrogation envoyés par un noeud, celui-ci considère qu'un incident est survenu. Cette méthode est par exemple employée par l'outil « ping », qui utilise les messages du protocole Internet Control Message Protocol (ICMP) de type ECHO REQUEST et ECHO RESPONSE pour vérifier la connectivité entre deux machines d'un réseau[75].

3.2.2 Travaux apparentés à la détection d'incident

Plusieurs travaux[47, 36] proposent un cadre pour les mécanismes de détection des incidents par envoi de messages sondes. En particulier, dans le cadre du développement de MPLS, un effort de standardisation de ces méthodes de détection d'incident a été effectué, au sein du protocole Bidirectional Forwarding Detection[36] (BFD), afin de fournir un cadre générique pour de tels mécanismes de détection d'incident. Cependant, ce protocole est spécifiquement destiné à être utilisé par les protocoles de routage, afin de détecter les pannes affectant les liens ou les routeurs. De plus, il n'est qu'un cadre pour la méthode de détection, mais ne contient pas de recommandations quant au paramétrage de ceux-ci.

En effet, une des problématiques lors de l'utilisation de ce type de mécanisme est la configuration de leurs paramètres. Il s'agit essentiellement de la fréquence d'envoi des messages sondes ainsi que du nombre de messages maximum pouvant être perdus avant que le mécanisme considère qu'un incident est apparu. Ainsi, si l'on souhaite détecter un incident en un temps court, il convient d'utiliser une fréquence d'envoi de messages et un nombre de messages pouvant être perdus faible. Cependant, cela a pour incidence une utilisation de bande passante réseau élevée à cause du nombre important de messages sondes utilisés. De plus, le risque d'apparition de faux positifs, lorsque des messages sondes sont perdus sans qu'un incident ne soit effectivement présent dans le réseau, est plus important. Ceci peut se produire lors d'une congestion ponctuelle du réseau par exemple.

Plusieurs études sont consacrées au paramétrage des protocoles HELLO utilisés dans les protocoles de routage pour détecter une perte de connectivité entre deux routeurs voisins. Le but de ces travaux est de diminuer le temps de détection d'une panne d'un routeur ou d'un lien par son routeur adjacent. Ceci permet de déclencher le mécanisme de re-routage pour permettre de rétablir la connectivité au plus vite[49]. Pour cela, il est nécessaire de diminuer la fréquence d'envoi des messages HELLO ainsi que le temps maximum où, en l'absence de réception d'un message HELLO, un routeur considère qu'un incident est apparu. Il a été montré[25] qu'en cas de congestion du réseau, une fréquence trop faible entraîne l'apparition de nombreux faux positifs, ce qui nuit à la stabilité du routage. Il est montré que le temps de détection optimal que l'on peut obtenir avec le protocole

HELLO d'OSPF est d'environ 4 secondes. Il a été proposé[17] un mécanisme de détection des incidents hybride : lorsque du trafic est envoyé sur un lien d'un routeur, le mécanisme considère qu'il n'est pas nécessaire d'envoyer un message HELLO puisque le routeur adjacent qui réceptionnera le trafic constatera qu'aucun incident n'est présent. Si le trafic cesse, l'algorithme reprend l'envoi de messages HELLO afin d'informer le routeur adjacent qu'aucun incident n'est apparu.

Certains travaux se consacrent à l'étude de détection d'un incident affectant une communication de bout en bout. Dans ce cas, le mécanisme de détection des incidents est déployé sur les noeuds participant effectivement à la communication. Ces mécanismes sont par exemple utilisés dans les réseaux overlays, où les noeuds voisins dans ces réseaux ne sont pas nécessairement voisins dans le réseau IP. Le mécanisme de détection utilisé dans RON a été étudié[112] : les temps de détection d'un incident et le nombre de messages sondes nécessaires sont exprimés en fonction de l'intervalle d'envoi des messages grâce à une modélisation mathématique du comportement du mécanisme de détection de RON. D'autres travaux[61] proposent un mécanisme de détection de type question/réponse, où le temps d'attente de la réponse par le noeud ayant envoyé la question est fonction du délai d'acheminement aller-retour (ou RTT pour Round Trip Time) mesuré pour les messages question/réponse précédemment envoyés. Cette méthode est similaire au calcul du Retransmission Time Out (RTO) par TCP, qui est le temps au bout duquel un paquet non acquitté est considéré comme perdu.

D'autres travaux expriment le temps de détection maximal d'un incident et la probabilité de faux positifs pour les mécanismes de détection distribués. Certains[16] proposent un mécanisme de détection des incidents basé sur le modèle Gossip, d'autres[23], qui propose un mécanisme de détection hiérarchique. Enfin, les performances de différents modèles (point à point ou distribués) de mécanisme de détection d'un incident affectant la communication vers un noeud au sein d'un réseau overlay de type Chord[98] ont été comparées [113].

3.3 Modélisation du comportement des mécanismes

Dans cette section, nous allons présenter les différentes modélisations utilisées pour étudier le fonctionnement des mécanismes de détection d'incident. L'ensemble des notations utilisées sont récapitulées dans la table 3.1.

3.3.1 Les différents modèles de fonctionnement

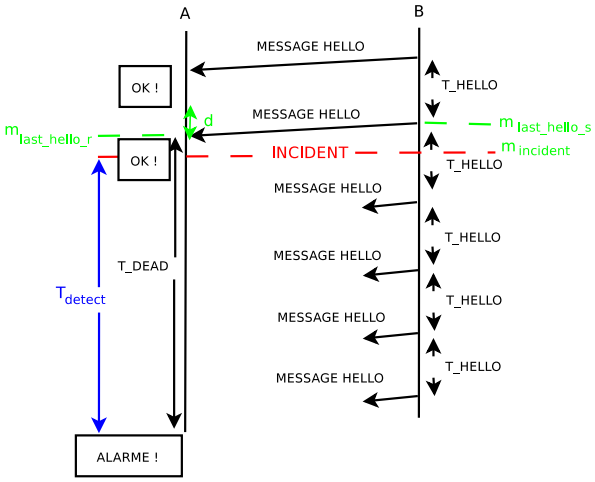
Nous l'avons souligné, il existe différents modèles de fonctionnement pour les mécanismes de détection d'incident par envoi de messages sondes. Nous allons maintenant présenter en détail le principe de fonctionnement de ces différents modèles. De plus, nous présenterons un nouveau modèle de fonctionnement appelé « Pull Adaptatif ».

Le modèle de détection Push

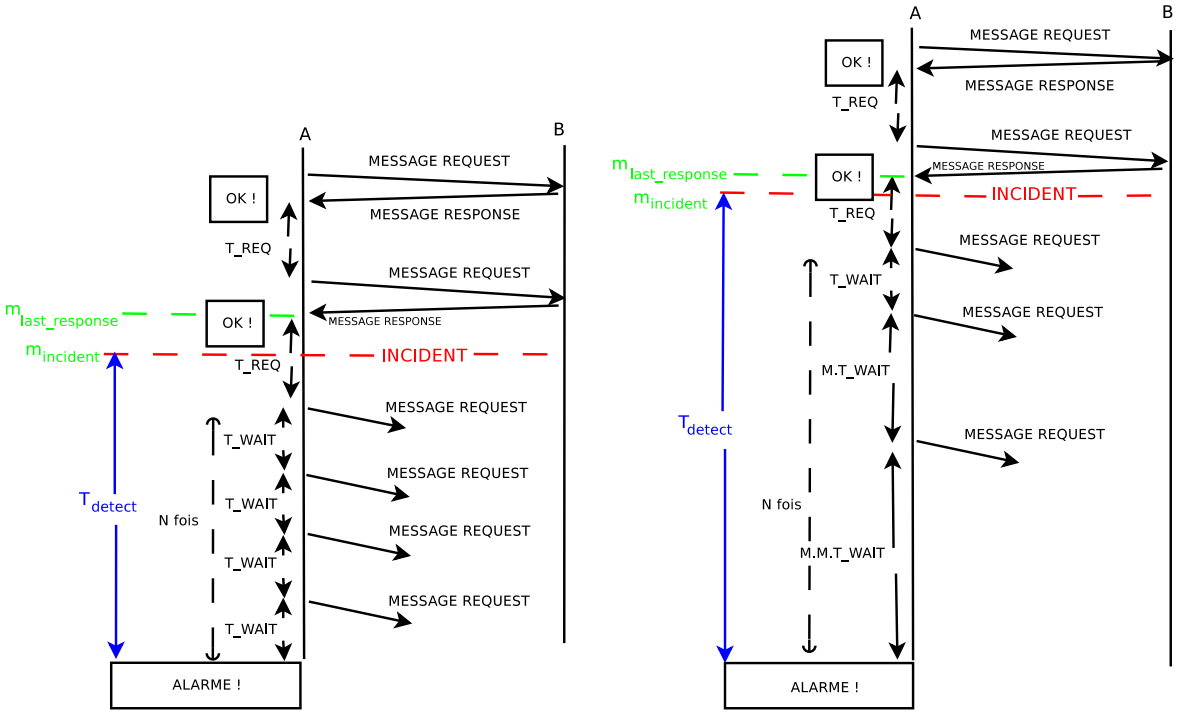
Deux modèles existent en effet : le premier est le modèle Push, appelé « HELLO protocol » lorsqu'il est employé par les protocoles de routage. Comme indiqué dans la figure 3.1a, dans le modèle Push, un noeud B envoie périodiquement un message HELLO à un noeud A. Si, au bout d'un temps donné, le noeud A n'a pas reçu de messages HELLO du noeud B, le noeud A considère qu'un incident affectant les communications vers le noeud B est survenu. Dans ce document, nous utiliserons la notation T_HELLO pour exprimer l'intervalle entre deux envois d'un message HELLO par le noeud B et T_DEAD pour exprimer le temps au bout duquel le noeud A, en l'absence de réception de messages HELLO provenant du noeud B, considère qu'un incident est survenu entre lui et le noeud B.

TAB. 3.1: Notations utilisées pour la modélisation des mécanismes de détection d'incident

Paramètres du modèle Push	
T_HELLO	Temps entre deux envois successifs d'un message HELLO
T_DEAD	Temps avant de déclarer un incident lorsqu'aucun message HELLO n'est reçu
Paramètres des modèles Pull et APull	
T_REQ	Temps avant l'envoi d'un message REQUEST après la réception du message RESPONSE
T_WAIT	Temps d'attente pour la réception d'un message RESPONSE après l'envoi d'un message REQUEST
N	Nombre maximum de messages RESPONSE successivement non reçus avant de déclarer un incident
M	Facteur multiplicatif du temps T_WAIT pour le mécanisme APull
Paramètres du réseau	
D	Délai d'acheminement minimal
V	Délai d'acheminement ajouté moyen
A	Disponibilité du réseau
Critères de performance	
TMD	Durée maximale de présence d'un incident non détecté tolérée
T_{detect}	Temps de détection d'un incident
R_{late}	Taux de détection tardive d'un incident
Q_{false}	Nombre de faux positifs par minute
Q_{msg}	Nombre de messages par minute
N_{msg}	Bande passante consommée par le mécanisme en kb/s



(a) Le type Push



(b) Le type Pull (c) Le type APull

FIG. 3.1: Fonctionnement des différents type de mécanisme de détection

Le modèle de détection Pull

Le deuxième modèle est le modèle Pull et est représenté dans la figure 3.1b. Ce modèle est un mécanisme de type question/réponse : un noeud A envoie un message REQUEST au noeud B. Lorsque le noeud B reçoit un message REQUEST, il retourne immédiatement un message RESPONSE au noeud A. Si le noeud A n'a pas reçu de messages RESPONSE provenant du noeud B à l'issu de l'envoi d'un certain nombre de messages REQUEST, il considère qu'un incident affectant les communications vers le noeud B est apparu. Dans ce document, nous utiliserons la notation T_REQ pour indiquer le temps pour réémettre un message REQUEST une fois qu'un message RESPONSE attendu a été reçu. Nous utiliserons la notation T_WAIT pour exprimer le temps maximum d'attente du message RESPONSE par le noeud A après qu'il ait envoyé un message REQUEST au noeud B. Si le temps T_WAIT est écoulé sans que le noeud A ait reçu de messages RESPONSE, il réémet immédiatement un message REQUEST. Enfin, nous utiliserons la notation N pour exprimer le nombre maximum de messages REQUEST consécutifs qui peuvent rester sans réponse avant de considérer qu'un incident est survenu.

Le modèle de détection Pull Adaptatif

Le dernier modèle que nous présenterons est le modèle Pull Adaptatif ou APull. C'est un modèle original, qui s'inspire des mécanismes utilisés pour le calcul du temps limite de retransmission dans TCP, le Retransmission Time Out[76], pour améliorer le mécanisme de type Pull. Son fonctionnement est représenté dans la figure 3.1c. Comme pour le modèle Pull, ce modèle est un mécanisme de question/réponse. La différence avec le mécanisme Pull est que le temps d'attente T_WAIT est calculé dynamiquement de manière à refléter les paramètres du réseau mesurés lors des précédents échanges de messages sondes. Ainsi, le temps d'attente T_WAIT pour l'arrivée d'un message RESPONSE en réponse à l'envoi d'un message REQUEST dépendra du Round Trip Time (RTT), le délai d'acheminement aller-retour, ainsi que de la gigue, ou variation du RTT, mesurés lors des échanges de ces messages. Ainsi, nous exprimons le temps T_WAIT par :

$$T\_WAIT = rtt_l + K \cdot vrtt_l$$

Les variables rtt_l et $vrtt_l$ sont des moyennes pondérées des RTT et de la gigue, mises à jour à chaque réception d'un message RESPONSE. Lors de la réception d'un message RESPONSE en un temps rtt_{new} après l'envoi d'un message REQUEST, les variables rtt_l et $vrtt_l$ sont mises à jour de la manière suivante :

$$vrtt_l \leftarrow (1-B) \cdot vrtt_l + B \cdot |rtt_l - rtt_{new}|$$
$$rtt_l \leftarrow (1-A) \cdot rtt_l + A \cdot rtt_{new}$$

En cas d'expiration du temps T_WAIT, c'est-à-dire lorsque le message RESPONSE n'est pas parvenu au bout du temps T_WAIT après l'envoi du message REQUEST, la valeur du temps d'attente T_WAIT est mise à jour de la manière suivante :

$$T\_WAIT \leftarrow T\_WAIT \cdot M$$

Différentes constantes sont utilisées dans cet algorithme. A et B sont des constantes qui déterminent le poids à accorder, respectivement, au temps rtt_{new} et à la variance $rtt_l - rtt_{new}$ mesurés lors

de la réception d'un nouveau message. K est une constante qui définit le poids à accorder à la variation du RTT pour le calcul du temps T_WAIT. Enfin, M est une constante qui définit le facteur d'augmentation de la durée T_WAIT d'attente d'un message RESPONSE à la suite d'une expiration de ce temps d'attente. De précédentes études [33] ont montré que les valeurs optimales, dans le cadre de l'utilisation de TCP, sont respectivement 1/8 et 1/4 pour A et B, 4 pour K. Nous utiliserons ces valeurs par la suite avec ce mécanisme.

3.3.2 Critères de performance étudiés

Nous allons tout d'abord exprimer quels sont les critères de performances des mécanismes, c'est-à-dire les différents éléments à étudier afin d'évaluer la performance d'un mécanisme de détection.

Nous considérons qu'en fonction des situations d'utilisation, un mécanisme de détection doit être configuré de manière à détecter un incident plus ou moins rapidement. Ainsi, le fonctionnement d'un tel mécanisme doit dépendre d'une constante, notée TMD, qui représente la durée maximale tolérée pendant laquelle un incident peut être présent dans le réseau sans qu'il soit détecté. Les critères de performance qui sont étudiés ici expriment la possibilité et les contraintes de réalisation de cet objectif par le mécanisme de détection d'incident.

Il faut remarquer que le temps TMD n'est pas simplement le temps de détection d'un incident maximum souhaité. En effet, à cela s'ajoute une contrainte sur la durée de l'incident : il n'est pas nécessaire que le mécanisme détecte un incident dont la durée est inférieure au temps TMD. Nous considérons donc que lorsque la durée d'un incident est inférieure à une certaine durée, dépendante du contexte d'utilisation du mécanisme de détection d'incident, l'incident n'a pas besoin et ne devrait pas être reporté. En effet, dans certaines situations, la détection d'un incident va entraîner une réaction importante dans un système et il faut par conséquent s'assurer que l'incident persiste suffisamment longtemps dans le système pour justifier une telle réaction.

Par conséquent, le temps TMD indique à la fois la durée minimum d'un incident à partir de laquelle celui-ci doit être reporté et le temps maximum souhaité pour la détection de cet incident.

Les critères de performances de ces mécanismes qui seront étudiés ici sont :

- La quantité de faux positifs, notée Q_{false}, qui représente le nombre moyen d'incidents signalés par le mécanisme de détection, par minute, alors qu'aucun incident n'était présent dans le réseau, ou bien qu'un incident était présent, mais que sa durée était inférieure au TMD.
- Le taux de détection tardive, noté R_{late}, qui est le rapport entre le nombre de détection par le mécanisme effectuées en un temps supérieur au TMD et le nombre total d'incident de durée supérieure au TMD dans le réseau.
- La quantité de messages nécessaire au fonctionnement du mécanisme, notée Q_{msg}. C'est le nombre de messages échangés dans le réseau, par minute, pour le fonctionnement du mécanisme. Nous étudierons aussi la quantité de bande passante consommée par les messages échangés, notée N_{msg}, en kilo bits par seconde. Nous fixerons alors la taille d'un message à 64 octets, ce qui correspond à la taille d'un paquet ICMP de type ECHO REQUEST[75].

3.3.3 Comportement du réseau

Avant de présenter les modèles des différents mécanismes de détection, nous allons introduire les notations décrivant le comportement du réseau dans lequel ils seront utilisés. En effet, la fréquence avec laquelle un incident affecte les communications du réseau, la distance dans le réseau entre les noeuds ou encore la variation du délai d'acheminement des messages entre ceux-ci sont des facteurs

qui affectent le fonctionnement d'un mécanisme de détection d'incident et qui sont par conséquent à prendre en considération.

Nous allons donc introduire des paramètres qui caractériseront la « configuration » du réseau reliant les deux noeuds sur lesquels est déployé le mécanisme de détection d'incidents. Voici ces paramètres :

- La distance entre les noeuds, notée D, qui sera exprimée en millisecondes (ms) et qui représente la durée d'acheminement aller-retour minimale d'un message dans le réseau entre ces deux noeuds.
- La variation moyenne du délai d'acheminement entre les noeuds, notée V, qui sera exprimée en millisecondes et qui représente le temps ajouté à D pour obtenir le délai d'acheminement aller-retour moyen d'un message entre les noeuds dans le réseau.
- La disponibilité moyenne d'une communication dans le réseau, notée A, qui représente la portion de temps durant lequel aucun incident affectant la communication n'est présent dans le réseau.

Lors de nos simulations, nous avons modélisé le temps d'achemienemt d'un message dans le réseau par une variable aléatoire, noté r, qui suit une distribution de Pareto ayant pour paramètre d'échelle la constante D et pour paramètre de forme la constante $\frac{D+V}{V}$. Ceci permet d'obtenir une espérance de $D+V$ et une valeur minimale de D pour la variable r. Cette modélisation du délai d'acheminement d'un réseau a déjà été utilisée et validée dans de précédents travaux[108].

Lorsque le délai d'acheminement pour « l'aller simple » d'un message a besoin d'être modélisé, nous avons utilisé une variable aléatoire, notée d, qui suit une distribution de Pareto de paramètres d'échelle $D/2$ et de forme $\frac{\frac{V}{2}+\frac{D}{2}}{\frac{V}{2}}$. Ceci permet d'obtenir une espérance de $\frac{D+V}{2}$ et une valeur minimale de $\frac{D}{2}$ pour la variable d.

Afin de modéliser l'apparition d'un incident affectant les communications du réseau, nous utilisons les temps moyens avant la panne, noté MTTF et le temps moyen de réparation, noté MTTR. Ces paramètres ont déjà été décrits dans la section 2.3.3.

Nous considérons que le taux de défaillance et de rétablissement d'une communication sont constants dans le temps. Par conséquent, dans notre modèle, le temps avant l'apparition d'un incident et la durée de celui-ci suivent une distribution exponentielle de paramètre $\frac{1}{MTTR}$ et $\frac{1}{MTTF}$, respectivement.

Dans nos simulations, nous avons choisi de fixer la valeur MTTR à 1 seconde. Ce choix, en concordance avec les mesures effectuées sur les communications de bout en bout et présentées dans la section 2.3.1, implique par exemple les propriétés suivantes :

- Lorsqu'un message est perdu, la probabilité qu'un message envoyé 500 ms plus tard soit aussi perdu est de 61 %.
- 50 % des incidents ont une durée inférieure à 694 ms.

Ainsi, le temps MTTF peut être calculé en fonction de A, qui est un des paramètres de la configuration du réseau présenté plus haut et du MTTR par la formule déjà présentée dans la section 2.3.3 :

$$MTTF = \frac{A.MTTR}{1-A}$$

La figure 3.2 présente le nombre d'incidents par jour affectant une communication dans notre réseau modélisé, en fonction de la durée de ces incidents, pour différentes valeurs de A (et MTTR=1s). On constate que lorsque $A = 0,999$, la moyenne du nombre d'incidents de durée supérieure à 5000 ms est inférieure à 1, alors qu'elle est environ de 6 pour $A = 0,99$. De plus, on constate que le nombre

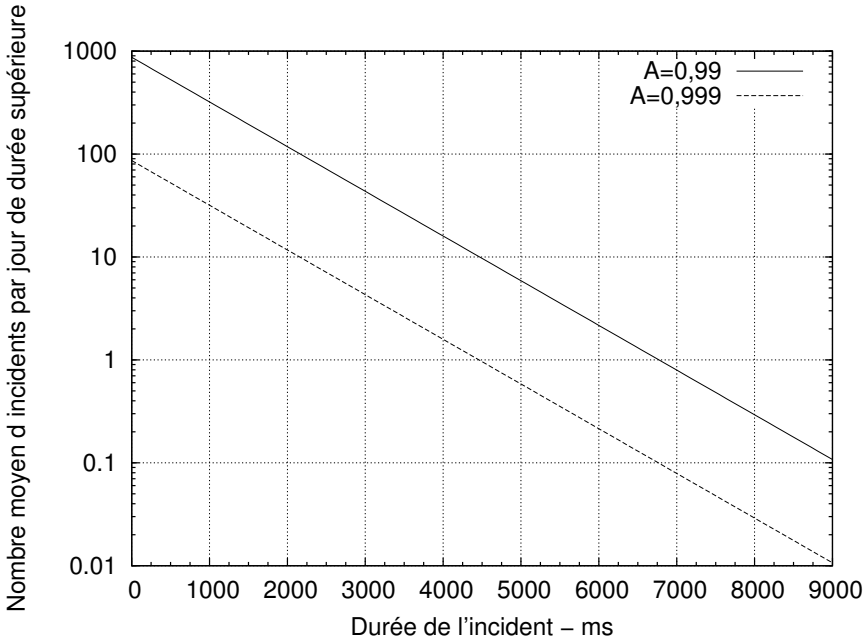


FIG. 3.2: Distribution des pannes et de leurs durées dans le modèle

d'incidents courts, d'une durée inférieure à 1 seconde est de plusieurs centaines lorsque $A = 0,99$, alors qu'il est de quelques dizaines lorsque $A = 0,999$. Le nombre d'incidents et leurs durées peuvent paraitre élevés. Ceci est voulu, car nous étudions les communications entre noeuds situés en bout de réseau, et leurs communications sont plus fréquemment interrompues que les communications entre deux routeurs adjacents, par exemple. De plus, nous souhaitons analyser le comportement des mécanismes de détection d'incident, et il est donc nécessaire d'étudier ces mécanismes dans des conditions où les incidents sont susceptibles d'apparaitre.

3.4 Choix des paramètres des différents mécanismes

Dans cette partie, nous allons étudier chacun des mécanismes présentés dans la section 3.3.1 séparément. L'objectif de cette étude est, pour chaque mécanisme, d'identifier les meilleurs paramètres à utiliser, en fonction du contexte d'utilisation de ce mécanisme, c'est-à-dire le temps *TMD* désiré pour le mécanisme et l'état du réseau.

3.4.1 Mécanisme de type Push

Nous allons tout d'abord nous intéresser au mécanisme de type Push.

Évaluation du temps de détection

Nous allons dans un premier temps exprimer le temps de détection d'un incident par le mécanisme Push. Nous considérons ici un incident devant être détecté, donc de durée supérieure au temps de détection du mécanisme.

Soit $m_{incident}$, l'instant d'apparition de l'incident dans le réseau, $m_{last\_hello\_r}$, l'instant où le noeud A a reçu un message HELLO pour la dernière fois, $m_{last\_hello\_s}$, l'instant où le noeud B a envoyé ce

message et d, le temps d'acheminement de ce message entre B et A. On note x mod y le reste de la division entière de x par y. On considère que le premier message HELLO est envoyé par le noeud B à l'instant 0.

Notons T_{detect}, le temps de détection d'un incident affectant le réseau. On a, comme illustrer par la figure 3.1a :

$$\begin{aligned} T_{detect} &= T\_DEAD - (m_{incident} - m_{last\_hello\_r}) \\ &= T\_DEAD - (m_{incident} - (m_{last\_hello\_s} + d)) \\ &= T\_DEAD - (m_{incident} - ((m_{incident} \\ &\quad - (m_{incident} \bmod T\_HELLO)) + d)) \\ &= T\_DEAD + d - (m_{incident} \bmod T\_HELLO) \end{aligned} \tag{3.1}$$

Le paramètre T_DEAD

Pour le protocole de type Push, le temps de détection est fortement lié à la valeur T_DEAD comme le montre l'évaluation du temps de détection maximum d'un incident par le mécanisme effectuée précédemment : Nous considérons le pire des cas où un incident affecte un noeud juste après que celui-ci est envoyé un message HELLO. Le temps de détection sera dans ce cas égal au temps T_DEAD additionné au temps d'acheminement du dernier message HELLO reçu. Par conséquent, lorsque le temps TMD est défini, le temps T_DEAD doit être défini comme légèrement inférieur, lorsque cela est possible, afin de permettre une détection en un temps inférieur à TMD, même dans le pire des cas.

Ainsi, nous proposons de choisir le paramètre T_DEAD de la manière suivante :

$$T\_DEAD = \begin{cases} TMD - 500ms & \text{, si TMD} > 1000\text{ms} \\ T\_DEAD = 500ms & \text{, si 500ms} < \text{TMD} \leq 1000\text{ms} \\ T\_DEAD = TMD & \text{, si TMD} \leq 500\text{ms} \end{cases}$$

Le paramètre T_HELLO

Nous allons maintenant étudier quel est le meilleur intervalle de temps T_HELLO à utiliser. Bien que cette valeur n'ait finalement que peu d'incidence sur le temps de détection final, le nombre de messages HELLO échangés va avoir une importance sur la qualité de la détection, en particulier sur le taux d'apparition de faux positifs Q_{false}. En effet, lorsque de nombreux messages HELLO sont envoyés, le risque qu'ils n'arrivent pas à destination lors d'un incident ponctuel dans le réseau est diminué. Bien sûr, le nombre de messages transitant dans le réseau sera directement lié à la fréquence T_HELLO d'envoi des messages HELLO.

Le graphique 3.3 montre le taux Q_false de faux positifs en fonction du nombre de messages HELLO envoyés pendant l'intervalle de temps T_DEAD, ce qui correspond à la valeur T_DEAD divisée par T_HELLO. Ces mesures ont été effectuées pour des valeurs de T_HELLO variants de 100 ms à 5000 ms. On observe que lorsque la disponibilité du réseau vaut $A = 0.999$, le nombre de faux positifs par minute Q_{false} est quasi nul dès que le nombre de messages HELLO envoyé par période T_DEAD est supérieur à 2. Par contre, lorsque la disponibilité du réseau est plus faible ($A = 0.99$), il est nécessaire d'utiliser au moins 3 messages HELLO par temps T_DEAD pour obtenir un nombre Q_{false} inférieur à 0.2 faux positif par minute.

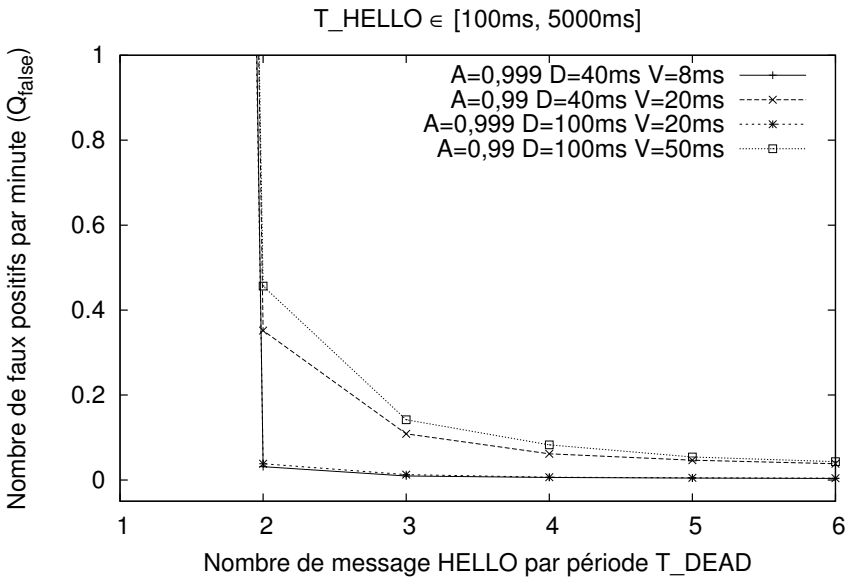


FIG. 3.3: Nombre de faux positifs du mécanisme Push en fonction du nombre de messages HELLO envoyé par période de temps T_DEAD

Lorsque la disponibilité du réseau est dégradée, il est donc nécessaire d'utiliser un nombre de messages HELLO par période T_DEAD supérieure à 2. Cependant, l'utilisation d'une valeur trop faible pour T_HELLO entraînera un coût significatif en bande passante. Nous considérerons donc que l'envoi de trois messages HELLO par période T_DEAD est un bon compromis, et nous utiliserons une valeur de $T\_HELLO = \frac{T\_DEAD}{3}$ dans la suite de nos mesures.

3.4.2 Mécanisme de type Pull

Nous allons maintenant étudier le mécanisme de type Pull.

Évaluation du temps de détection

Nous allons dans un premier temps exprimer le temps de détection d'un incident par le mécanisme Pull. Nous considérons ici un incident devant être détecté par le mécanisme, donc de durée supérieure au temps de détection du mécanisme.

Soit $m_{incident}$, l'instant d'apparition de l'incident dans le réseau et $m_{last\_response}$, l'instant où le noeud A a reçu un message RESPONSE pour la dernière fois. On considère que le premier message REQUEST est envoyé par le noeud A à l'instant 0.

Notons T_{detect}, le temps de détection d'un incident affectant le réseau. On a, comme illustré par la figure 3.1b :

$$\begin{aligned} T_{detect} &= m_{last\_response} + T\_REQ + N.T\_WAIT - m_{incident} \\ &\text{, avec : } m_{last\_response} \leq m_{incident} < m_{last\_response} + T\_WAIT \end{aligned} \tag{3.2}$$

On constate qu'avec le mécanisme de type Pull, le temps de détection est lié aux valeurs T_REQ, T_WAIT et N. Nous considérons le pire des cas où un incident affecte une communication juste après

que le noeud A est envoyé un message REQUEST. Le temps de détection sera dans ce cas égal au temps $T\_REQ + N.T\_WAIT$. Par conséquent, afin de permettre au mécanisme Pull de détecter un incident en un temps inférieur au TMD, nous proposons de choisir les paramètres du mécanisme Pull en fonction du TMD suivant la formule :

$$TMD = T\_REQ + N.T\_WAIT \tag{3.3}$$

Le paramètre T_WAIT

Nous allons étudier les performances du mécanisme en fonction du choix de la valeur de T_WAIT. Cette valeur a une incidence sur le nombre de faux positifs Q_{false}. En effet, si le mécanisme n'attend pas suffisamment longtemps un message RESPONSE et considère celui-ci comme perdu alors que ce n'est pas le cas, un faux positif peut finir par être déclenché. Cependant, choisir un temps T_WAIT trop grand va entraîner une augmentation du temps de détection total. Il est clair que la valeur optimale de T_WAIT dépendra de la distance dans le réseau entre les noeuds sur lesquels est déployé le mécanisme et qu'elle doit par conséquent être choisie suffisamment grande pour garantir un taux de faux positif faible, même lorsque le délai d'acheminement entre deux noeuds dans le réseau est important.

Le graphique 3.4 présente le taux Q_{false} en fonction de la valeur de T_WAIT, pour les différentes configurations de réseau. Ces résultats ont été obtenu en utilisant une valeur variant de 100 à 5000 ms pour T_REQ et de 3 pour N. On observe que lorsque la disponibilité du réseau est de $A = 0.999$, le nombre de faux positifs par minute Q_{false} est presque nul dès que le temps T_WAIT est supérieur à 50 ms lorsque le délai d'acheminement minimal $D = 40ms$ et à 175 ms lorsque $D = 100ms$. Lorsque la disponibilité du réseau est $A = 0.99$, Q_{false} décroit plus lentement vers zéro : lorsque $T\_WAIT = 250ms$, le nombre de faux positifs par minute est légèrement inférieur à 0.2, lorsque $T\_WAIT = 1000ms$, il est de 0.04.

Ce graphique confirme la nécessité de choisir le temps T_WAIT en fonction du délai d'acheminement du réseau. En effet, si le temps d'attente d'un message n'est pas supérieur au temps d'acheminement de ce message dans le réseau, il est normal d'observer un grand nombre de faux positifs, puisque le message est déclaré perdu avant qu'il n'ait eu le temps d'être acheminé. C'est ce que l'on observe dans le graphique lorsque $T\_WAIT < D + V$. Puisque le mécanisme Pull ne connait pas à priori le temps d'acheminement des communications entre les noeuds sur lesquels il est déployé, il est nécessaire de choisir une valeur suffisamment grande pour T_WAIT. Nous constatons de plus que lorsque la partie variable du délai d'acheminement est importante (A=0.99), le nombre de faux positifs reste élevé si le temps T_WAIT n'est pas assez grand. Il est donc nécessaire de choisir un temps T_WAIT grand, bien que cela augmente le temps de détection global d'un incident. Nous proposons ainsi d'utiliser la valeur $T\_WAIT = 1000ms$, lorsque le temps TMD souhaité le permet.

Le paramètre N

Nous allons maintenant étudier les performances du mécanisme en fonction du choix de la valeur de N. Comme précédemment avec le choix de T_HELLO pour le protocole Push, le choix de N va avoir une incidence sur le nombre de faux positifs. En effet, envoyer plusieurs messages REQUEST avant de considérer qu'un incident est apparu permet de limiter l'apparition d'un faux positif lorsque plusieurs messages REQUEST sont perdus suite à un incident de courte durée.

Le graphique 3.5 présente le taux Q_{false} en fonction du choix de N et de la configuration du réseau. Les valeurs présentées ont été obtenues pour $T\_WAIT = 1000ms$ et T_REQ variant de 100 à 5000 ms.

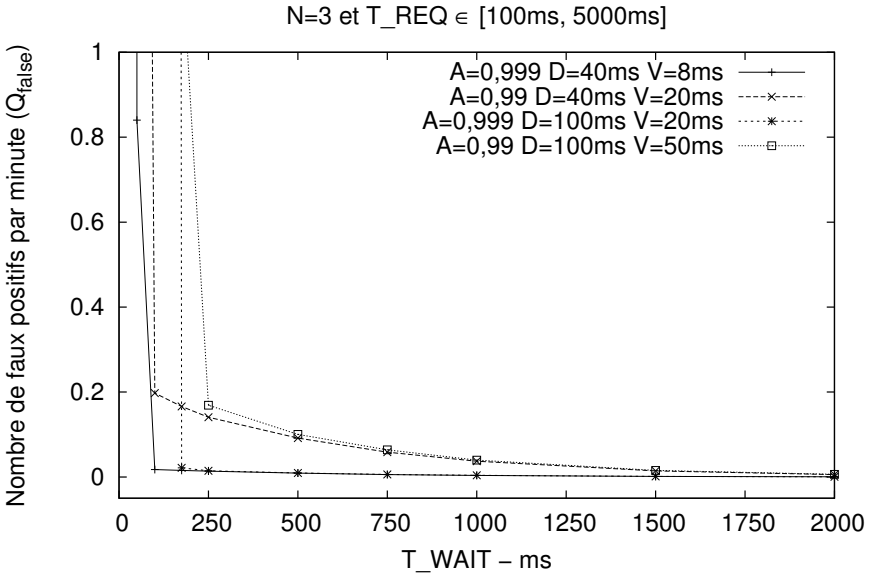


FIG. 3.4: Influence du paramètre T_WAIT sur le nombre de faux positifs du mécanisme Pull

On observe que lorque la disponibilité du réseau est importante ($A = 0.999$), le nombre Q_{false} de faux positifs par minute est très faible, même si un unique message REQUEST peut être perdu ($N = 1$). Lorsque la disponibilité du réseau est dégradé ($A = 0.99$), le nombre Q_{false} décroit plus lentement avec N : il est supérieur à 0,1 lorsque $N = 2$ et environ égal à 0.04 lorsque $N = 3$.

Il est donc nécessaire d'utiliser plusieurs messages REQUEST pouvant être perdus successivement pour ne pas observer trop de faux positifs lorsque le réseau est dégradé. En effet dans ce cas, il est plus courant qu'un incident entraîne la perte d'un ou deux messages REQUEST (ou RESPONSE) alors que sa durée est faible. Cependant, comme pour le paramètre T_WAIT, le choix d'une grande valeur pour N augmente le temps de détection global d'un incident. Par conséquent, nous proposons d'utiliser une valeur de N égale à 3 lorsque le temps TMD désiré le permet.

Le paramètre T_REQ

Nous allons maintenant déterminer les valeurs à choisir pour le paramètre T_REQ. Puisque les autres valeurs ont été déterminées, nous pouvons déduire la valeur de T_REQ à utiliser en fonction du TMD à l'aide de la formule 3.3 présentée plus haut. On a en effet :

$$T\_REQ = TMD - N.T\_WAIT$$

Par conséquent, puisque nous avons proposé d'utiliser une valeur de 3 pour N et de 1000 pour T_WAIT, le paramètre T_REQ peut être choisi de la manière suivante :

$$T\_REQ = TMD - 3000ms$$

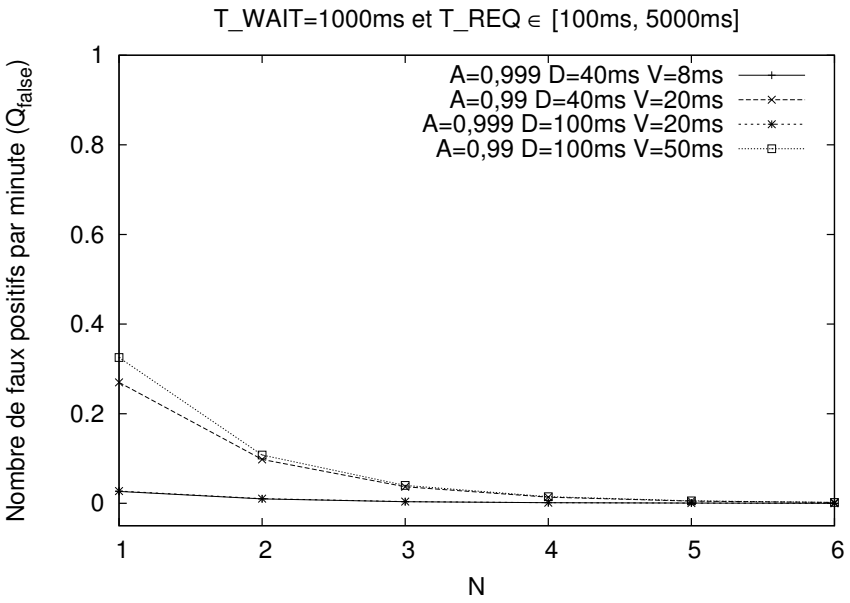


FIG. 3.5: Influence du nombre N de messages pouvant être perdus sur le nombre de faux positifs du mécanisme Pull

TAB. 3.2: Choix des paramètres du mécanisme de type Pull

	TMD - ms		
Paramètre	$TMD > 3500$	$1750 < TMD \leq 3500$	$TMD \leq 1750$
T_WAIT (ms)	1000	500	500
N	3	3	2
T_REQ (ms)	$TMD - 3000$	$TMD - 1500$	$max(0, TMD - 1000)$

Choix des paramètres pour les faibles TMD

Il apparait que le choix du paramètre T_REQ présenté ci-dessus n'est possible que dans le cas où le TMD est supérieur à 3000 ms. Sinon, il faut choisir une valeur de T_REQ nulle. Ceci aurait deux conséquences :

- Une importante consommation de ressource réseau, puisqu'un message RESQUEST est immédiatement émis à la suite de la réception d'un message RESPONSE.
- L'impossibilité d'assurer la détection en un temps inférieur au TMD, puisque l'égalité $TMD = T\_REQ + N.T\_WAIT$ ne serait plus respectée.

Par conséquent, nous proposons de diminuer les valeurs des paramètres T_WAIT et N lorsque le temps TMD est trop faible. La diminution de ces paramètres va cependant entrainer un nombre de faux positifs et un taux de détection tardive plus importants. Malgré tout, cela est nécessaire pour permettre la détection d'un incident en un temps inférieur au TMD. Le tableau 3.2 présente nos propositions pour le choix des valeurs des différents paramètres du mécanisme Pull, en fonction du TMD. Nous avons choisis ces différentes valeurs de manière à diminuer au maximum les paramètres T_WAIT et N, sans que cela n'entraine un trop grand nombre de faux positifs, en fonction des observations précedemment effectuées.

3.4.3 Mécanisme de type Pull adaptatif

Nous allons maintenant étudier le mécanisme de type Pull adaptatif (APull).

Évaluation du temps de détection

Nous allons dans un premier temps exprimer le temps de détection d'un incident par le mécanisme APull. Nous considérons un incident devant être détecté par le mécanisme, donc dont la durée est supérieure au TMD.

Le temps d'attente T_WAIT du mécanisme APull est variable, en fonction du délai d'acheminement des messages REQUEST et RESPONSE précédemment échangés : Comme indiqué dans la présentation de ce mécanisme dans la section 3.3.1, en cas de non-réception d'un message RESPONSE après un temps d'attente T_WAIT, un nouveau message est envoyé et le temps T_WAIT est multiplié par la constante M. Par conséquent, lors d'un incident, le temps T_WAIT va évoluer ainsi :

$T\_WAIT \leftarrow T\_WAIT_0$,
où le temps T_WAIT_0 est calculé après la réception
du dernier message RESPONSE
$T\_WAIT \leftarrow M.T\_WAIT_0$,
à l'issue de l'absence de réception du message RESPONSE
au bout d'un temps T_WAIT après l'envoi du message REQUEST
$T\_WAIT \leftarrow M.M.T\_WAIT_0$,
de même
...

Ceci est répété N fois, après quoi l'alarme est déclenchée. Ainsi, le temps total d'attente après l'envoi d'un message REQUEST en présence d'un incident est :

$$\sum_{k=0}^{N-1} (T\_WAIT_0.M^k) = T\_WAIT_0.\frac{1-M^N}{1-M}$$

Soit $m_{incident}$, l'instant d'apparition de l'incident dans le réseau et $m_{last\_response}$, l'instant où le noeud A a reçu un message RESPONSE pour la dernière fois. On considère que le premier message REQUEST est envoyé par le noeud A à l'instant 0.

Notons T_{detect}, le temps de détection d'un incident affectant le réseau. On a, comme illustré par la figure 3.1c :

$$T_{detect} = m_{last\_response} + T\_REQ + T\_WAIT_0.\frac{1-M^N}{1-M} - m_{incident} \tag{3.4}$$

Nous considérons le pire des cas où un incident affecte une communication juste après que le noeud A est envoyé un message REQUEST. Le temps de détection sera dans ce cas égal au temps $T\_REQ + T\_WAIT_0.\frac{1-M^N}{1-M}$. Par conséquent, afin de permettre au mécanisme APull de détecter un incident en un temps inférieur au TMD, nous proposons de choisir les paramètres du mécanisme APull en fonction du TMD suivant la formule :

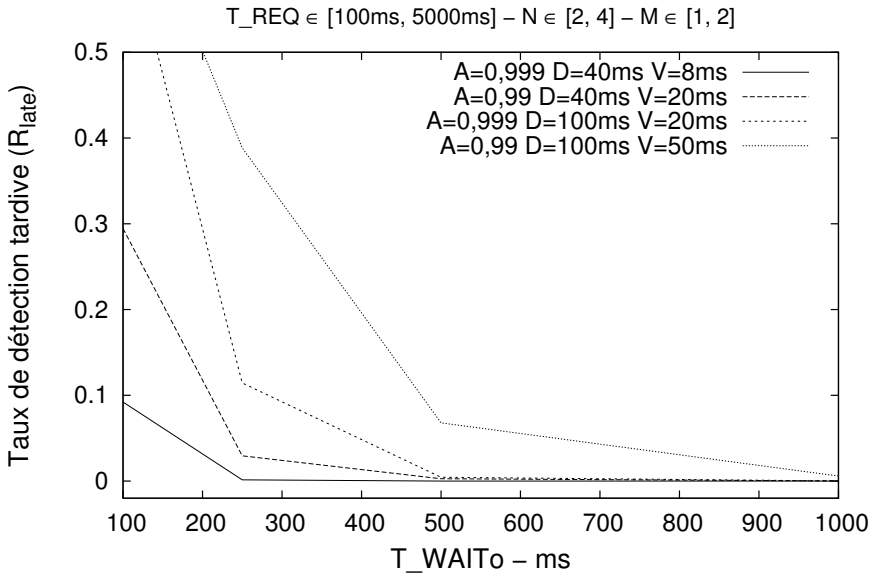


FIG. 3.6: Influence de la valeur choisie pour T_WAIT_0 sur le taux de détection tardive du mécanisme APull

$$TMD = T\_REQ + T\_WAIT_0.\frac{1-M^N}{1-M} \tag{3.5}$$

Choix de la valeur T_WAIT_0

Nous allons étudier les performances du mécanisme en fonction du choix de la valeur de T_WAIT_0. Comme montré par la formule ci-dessus, cette valeur est utilisée lors du calcul du TMD et représente le dernier temps T_WAIT calculé par le mécanisme avant qu'un message ne soit perdu. Il faut souligner que cette valeur n'est pas un paramètre du mécanisme APull mais intervient uniquement pour le calcul du temps TMD en fonction des paramètres du mécanisme.

Il est nécessaire de choisir une valeur de T_WAIT_0 qui soit proche de la valeur de T_WAIT calculée par le mécanisme lors de la réception du dernier message RESPONSE avant l'apparition de l'incident. En effet, si la valeur de T_WAIT_0 est trop inférieure à la valeur calculée de T_WAIT, le temps de détection d'un incident pourra être plus grand que le temps TMD requis. Cependant, il convient de choisir une valeur faible pour T_WAIT_0 afin de minimiser le temps TMD atteignable avec le mécanisme APull.

Le graphique 3.6 présente le taux R_{late} en fonction de la valeur de T_WAIT_0, pour les différents configurations de réseau. Ces résultats ont été obtenu en utilisant une valeur variant de 100 à 5000 ms pour T_REQ, de 2, 3 ou 4 pour N et de 1 ; 1,25 ; 1,50 ; 1,75 ou 2 pour M.

On observe qu'en fonction des configurations réseau observées, le taux de détection tardive est proche de 0 à partir d'une certaine valeur de T_WAIT_0. Lorsque le délai d'acheminement moyen entre les noeuds est faible ($D = 40ms$), le taux R_late est inférieur à 0,03 dès que T_WAIT_0 est supérieur à 250 ms. Lorsque le délai d'acheminement entre les noeuds est plus grand ($D = 100ms$), le taux R_late devient inférieur à 0,1 lorsque T_WAIT_0 est supérieur à 500 ms.

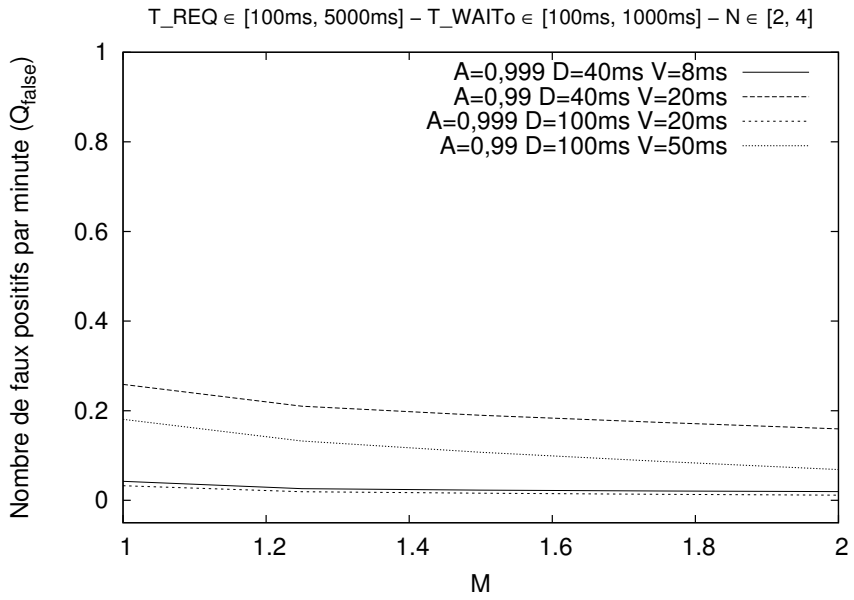


FIG. 3.7: Influence du paramètre M sur le nombre de faux positifs du mécanisme APull

Il apparait que le choix minimal du temps T_WAIT_0 dépend du délai d'acheminement dans le réseau. En effet, le temps T_WAIT est calculé par le mécanisme APull en fonction du délai d'acheminement des messages. Il est par conséquent normal, pour une même valeur T_WAIT_0, d'observer différents taux R_{late} en fonction des différentes configurations de réseau. Puisqu'en général, il n'est pas possible d'estimer à priori le délai d'acheminement des communications entre les noeuds sur lesquels le mécanisme APull est déployé, il est nécessaire de choisir une valeur suffisamment grande pour T_WAIT_0. Nous proposons ainsi de choisir une valeur $T\_WAIT_0 = 500ms$.

Le paramètre M

Nous allons étudier les performances du mécanisme en fonction du choix de la valeur de M. Nous l'avons vu, M est le facteur par lequel est multiplié le temps d'attente d'un message T_WAIT à la suite de la perte d'un message. Une grande valeur de M permet ainsi de limiter l'apparition de faux positifs consécutifs à la livraison tardive de messages RESPONSE, mais aura pour conséquence un temps de détection des incidents plus importants. Il faut ainsi choisir la plus petite valeur pour M qui n'introduise pas un trop grand nombre de faux positifs.

Le graphique 3.7 présente le taux Q_{false} en fonction de la valeur de M, pour les différentes configurations de réseau. Ces résultats ont été obtenu en utilisant une valeur variant de 100 ms à 5000 ms pour T_REQ, de 100 ms à 1000 ms pour T_WAIT_0 et de 2, 3 ou 4 pour N. On observe que lorsque la variation du délai d'acheminement est faible et que la disponibilité du réseau est importante ($A = 0.999$), le nombre de faux positifs par minute Q_{false} est toujours inférieur à 0,05 et le choix de M a peu d'influence sur cette valeur. Ce n'est pas le cas avec une configuration de réseau « dégradée » ($A = 0.99$) : La valeur Q_{false} est alors plus importante et le choix de M influe sur cette valeur. On observe que le nombre de faux positifs par minute varie de 0,25 pour $M = 1$ à 0,16 pour $M = 2$ lorsque $D = 40ms$ et de 0,18 pour $M = 1$ à 0,07 pour $M = 2$ lorsque $D = 100ms$

Il apparait ainsi que lorsque le réseau est « dégradé », c'est-à-dire lorsque sa disponibilité est faible

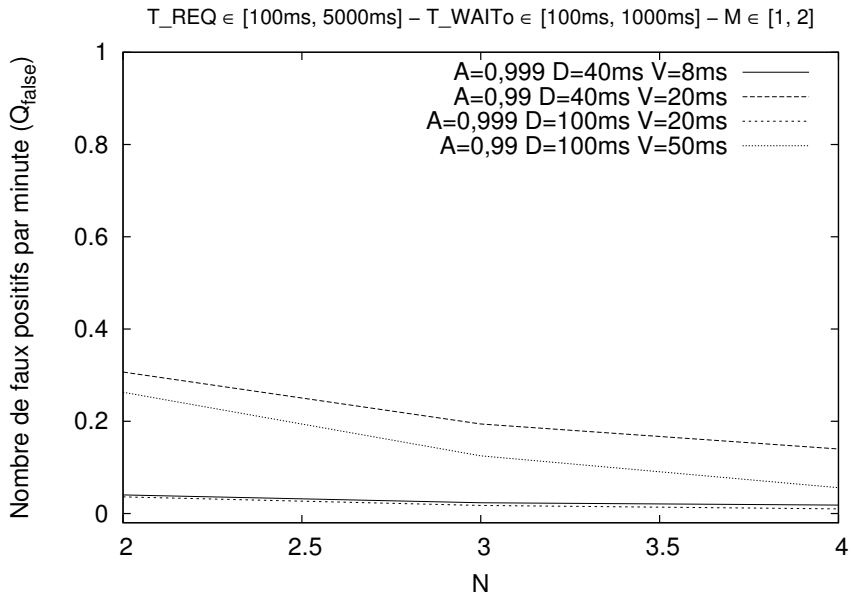


FIG. 3.8: Influence du nombre N de messages pouvant être perdus sur le nombre de faux positifs du mécanisme APull

($A = 0.99$) et que la variation du délai d'acheminement est importante par rapport au délai d'acheminement moyen ($V = 20ms$ ou $V = 50ms$), il est important de choisir une valeur de M supérieure à 1. Cependant, un tel choix ne diminue que faiblement le nombre de faux positifs en particulier à partir de $M \geq 1,25$. Puisque M a une forte influence sur le temps de détection de l'incident et que nous cherchons à minimiser ce temps, nous proposons d'utiliser une valeur de $M = 1,5$.

Le paramètre N

Nous allons maintenant étudier les performances du mécanisme APull en fonction du choix du paramètre N. De la même manière qu'avec le mécanisme Pull, N va avoir une incidence sur le nombre de faux positifs.

Le graphique 3.8 présente le taux Q_{false} en fonction du choix de N et de la configuration du réseau. Les valeurs présentées ont été obtenues pour T_REQ variant de 100 à 5000 ms, T_WAIT_0 de 100 à 1000 et M de 1 à 2.

On observe que lorque la disponibilité du réseau est importante ($A = 0.999$), le nombre Q_{false} de faux positifs par minute est très faible, même si un unique message REQUEST peut être perdu ($N = 1$). Lorsque la disponibilité du réseau est dégradé ($A = 0.99$), le nombre Q_{false} est plus important : il est supérieur à 0,2 si $N \leq 2$ et et décroit lentement lorsque N augmente.

De même qu'avec le mécanisme Pull, il est nécessaire d'utiliser un nombre suffisant de messages REQUEST avant de déclencher une alarme pour ne pas observer trop de faux positifs lorsque le réseau est dégradé. Cependant, comme précédemment, le choix d'une grande valeur pour N augmente le temps de détection global d'un incident. Par conséquent, nous proposons d'utiliser une valeur de N égale à 3 lorsque le temps TMD désiré le permet.

TAB. 3.3: Choix des paramètres du mécanisme de type APull

	TMD - ms		
Paramètre	$TMD > 2600$	$1600 < TMD \leq 2600$	$TMD \leq 1600$
T_WAIT_0	500	300	300
M	1,5	1,5	1,5
N	3	3	2
T_REQ - (ms)	$TMD - 2375$	$TMD - 1425$	$max(0, TMD - 750)$

Le paramètre T_REQ

Nous allons maintenant déterminer les valeurs à choisir pour le paramètre T_REQ. Puisque les autres valeurs ont été déterminées, nous pouvons déduire la valeur de T_REQ à utiliser en fonction du TMD à l'aide de la formule 3.5 présentée plus haut. On a en effet :

$$T\_REQ = TMD - T\_WAIT_0.\frac{1-M^N}{1-M}$$

Par conséquent, puisque nous avons proposé d'utiliser une valeur de 3 pour N, de 500 ms pour T_WAIT_0 et de 1,5 pour M, le paramètre T_REQ peut être choisi de la manière suivante :

$$\begin{aligned} T\_REQ &= TMD - 500ms.\frac{1-1,5^3}{1-1,5} \\ &= TMD - 2375ms \end{aligned}$$

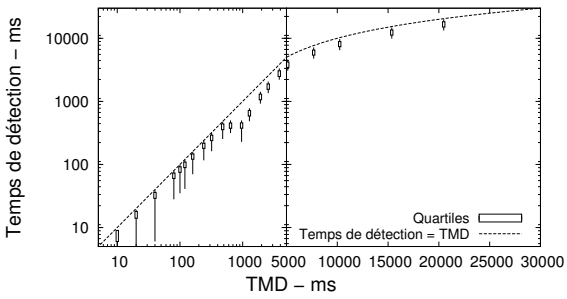
Choix des paramètres pour les faibles TMD

Le choix du paramètre T_REQ présenté ci-dessus n'est possible que dans le cas où le TMD est supérieur à 2375 ms. Il faudrait dans le cas contraire choisir une valeur de T_REQ nulle. De même que pour le mécanisme Pull, il est nécessaire d'adapter les paramètres T_WAIT_0, M et N lorsque le TMD demandé est trop faible afin de permettre la détection des incidents en un temps inférieur au TMD, bien que cela entraînera une dégradation des performances du mécanisme.

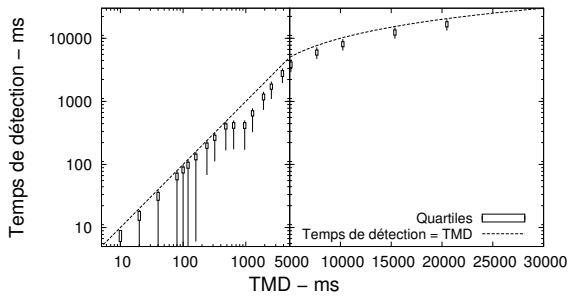
Le tableau 3.3 présente nos propositions pour le choix des valeurs des différents paramètres du mécanisme APull, en fonction du TMD. Ces différentes valeurs ont été choisies de manière à diminuer au maximum les paramètres T_WAIT_0, M et N, sans entrainer un trop grand nombre de faux positifs, en fonction des observations précedemment effectuées.

3.5 Évaluation et discussion des performances des mécanismes

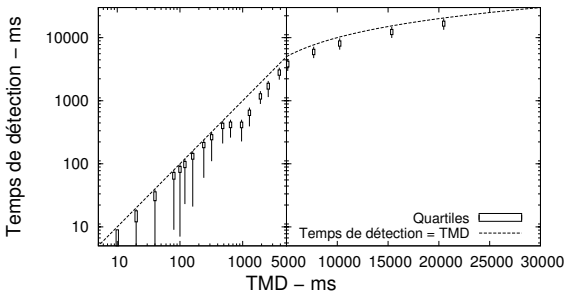
Maintenant que les paramètres optimaux des différents mécanismes ont été étudiés, il est possible de comparer leurs performances. Dans cette partie, nous allons étudier, pour chacun des critères introduits plus haut, les performances des différents mécanismes, en fonction d'un temps TMD déterminé. Ainsi, les résultats présentés ici expriment un des critères de performance des mécanismes de détection d'incident évoqué dans la section 4.4.1 en fonction du temps TMD requis, qui est la durée maximale de présence d'un incident non détecté dans le réseau. Par conséquent, le mécanisme est



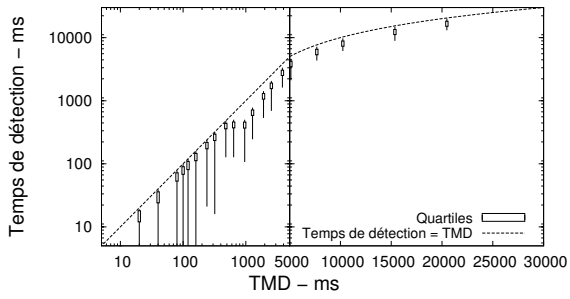
(a) Configuration réseau : A=0,999 D=40ms V=8ms



(b) Configuration réseau : A=0,99 D=40ms V=20ms



(c) Configuration réseau : A=0,999 D=100ms V=20ms



(d) Configuration réseau : A=0,99 D=100ms V=50ms

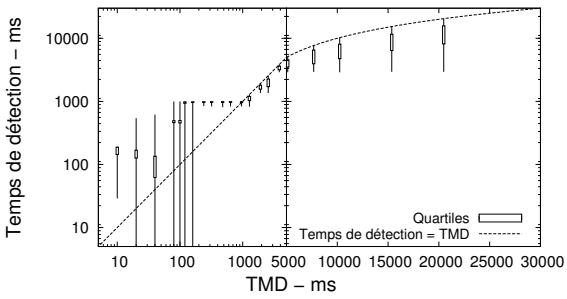
FIG. 3.9: Répartition des temps de détection mesurés avec le mécanisme Push, en fonction du temps TMD demandé, pour les différentes configurations de réseau

configuré en fonction du TMD demandé en accord avec ce qui a été présenté dans la section précédente. Comme indiqué dans cette section, lorsque le TMD est faible, des paramètres « dégradés » sont utilisés.

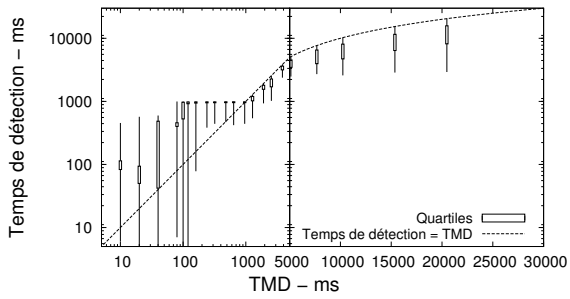
3.5.1 Temps de détection d'un incident

Nous allons étudier les temps de détection d'un incident observés avec les différents mécanismes, en fonction du temps TMD demandé et pour différentes configurations de réseau. Les graphiques 3.9a, 3.9b, 3.9c, 3.9d, 3.10a, 3.10b, 3.10c, 3.10d, 3.11a, 3.11b, 3.11c et 3.11d présentent, pour différentes configurations réseaux, la répartition des temps de détection des différents mécanismes en fonction des temps TMD demandés. La répartition des temps de détection est exprimée sous forme de quartiles : pour chaque valeur de TMD étudiée, le trait supérieur indique les valeurs des 25 % plus grands temps de détection observés, le trait inférieur indique les valeurs des 25 % plus faibles temps de détection observés et la « boite » indique les valeurs des 50 % temps de détection observés restant. De plus, sur chaque graphique, la courbe $T_{detect} = TMD$ est tracée. Ainsi, les temps de détection indiqués par les quartiles se trouvant au-dessus de cette courbe indiquent une détection tardive, puisque ces détections ont été réalisées en un temps supérieur au TMD.

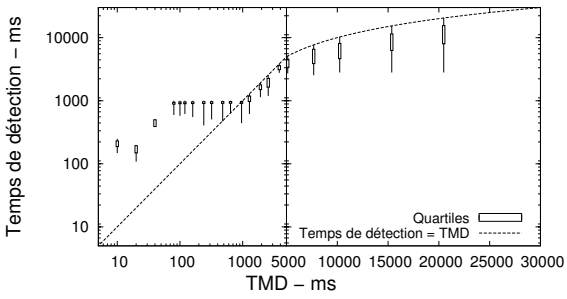
Nous allons discuter des temps de détection du mécanisme de type Push observés dans les gra-



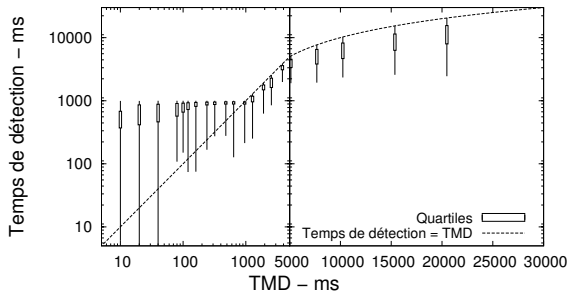
(a) Configuration réseau : A=0,999 D=40ms V=8ms



(b) Configuration réseau : A=0,99 D=40ms V=20ms



(c) Configuration réseau : A=0,999 D=100ms V=20ms



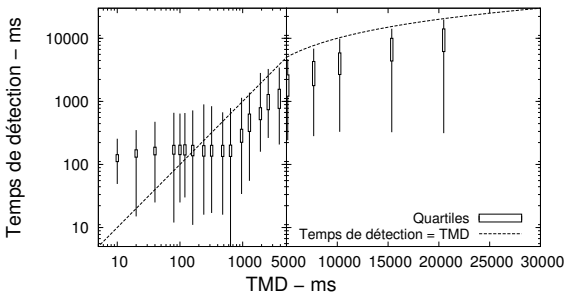
(d) Configuration réseau : A=0,99 D=100ms V=50ms

FIG. 3.10: Répartition des temps de détection mesurés avec le mécanisme Pull, en fonction du temps *TMD* demandé, pour les différentes configurations de réseau

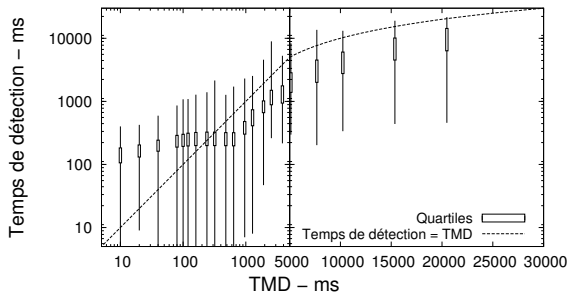
phiques 3.9a,3.9b,3.9c et 3.9d. Nous constatons qu'avec ce mécanisme, les temps de détection sont systématiquement inférieurs au *TMD*, quel que soit ce dernier et quelle que soit la configuration de réseau observée. De plus, la répartition des temps de détection est faible, c'est-à-dire que les écarts observés entre les temps de détection pour une même valeur de *TMD* sont peu important. Enfin, pour un *TMD* donné, les temps de détection sont généralement proches de ce temps.

Nous pouvons expliquer ces résultats par la façon de fonctionner du mécanisme Push : avec ce mécanisme, des messages HELLO sont envoyés à intervalles constants dans le réseau. Par conséquent, il est normal que les résultats obtenus dépendent peu de la configuration du réseau sur lequel le mécanisme est déployé. De plus, le paramètre *T_DEAD* du mécanisme Push, qui détermine le temps avant de signaler un incident, est toujours inférieur ou égal au *TMD*. Il est donc normal que les temps de détection observés soient inférieurs au *TMD*. Enfin, la faible répartition des temps de détection est la conséquence qu'ils sont tous approximativement compris entre $T\_DEAD - T\_HELLO$ et *T_DEAD*, comme expliqué dans la formule 3.1.

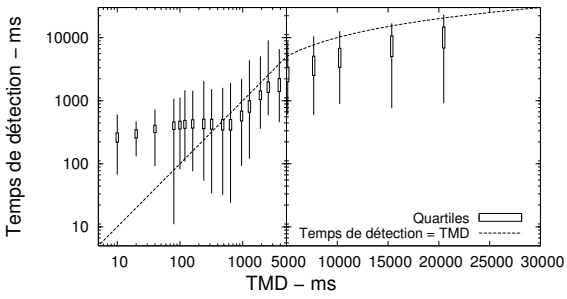
Nous allons discuter des temps de détection du mécanisme de type Pull observés dans les graphiques 3.10a,3.10b,3.10c et 3.10d. Avec ce mécanisme, nous constatons que les temps de détection ne sont pas, dans la plupart des cas, supérieurs au *TMD* lorsque celui-ci est inférieur à 1 seconde. Les temps de détection minimum généralement observés avec le mécanisme Pull sont de l'ordre de la



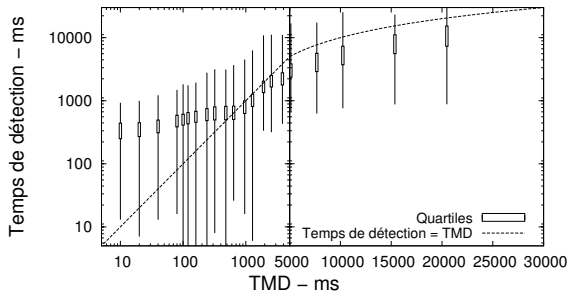
(a) Configuration réseau : A=0,999 D=40ms V=8ms



(b) Configuration réseau : A=0,99 D=40ms V=20ms



(c) Configuration réseau : A=0,999 D=100ms V=20ms



(d) Configuration réseau : A=0,99 D=100ms V=50ms

FIG. 3.11: Répartition des temps de détection mesurés avec le mécanisme APull, en fonction du temps TMD demandé, pour les différentes configurations de réseau

centaine de millisecondes, et il semble que plus le délai d'acheminement entre les noeuds du réseau est important et plus le temps minimal moyen de détection est élevé. Pour que les temps de détection soient inférieurs au TMD, il faut que ce dernier soit supérieur ou égal à une seconde. La répartition des temps de détection est assez large pour un temps TMD donné. Même si la plupart des détections sont réalisées en un temps légèrement inférieur au TMD, une minorité est effectuée en un temps largement plus court. Ce phénomène semble plus marqué lorsque le réseau est « dégradé » ($A = 0.99$).

Comme expliqué dans la section 3.4.3, lorsque le TMD est faible, nous avons choisi de dégrader les paramètres du mécanisme Pull afin de permettre une détection plus rapide. Cependant, celle-ci ne peut être arbitrairement basse en raison du fonctionnement du mécanisme Pull ainsi que des valeurs minimales à conserver pour les paramètres de ce mécanisme afin de ne pas voir apparaître un trop grand nombre de faux positifs. Ainsi, il n'est pas étonnant d'observer l'impossibilité pour le mécanisme Pull de détecter de manière fiable un incident en un temps inférieur à 1000 ms. Comme montré par la formule 3.2, les temps de détection de ce mécanisme sont compris approximativement entre $N.T\_WAIT$ et $T\_REQ + N.T\_WAIT$. Par conséquent, il est normal d'observer une plus grande répartition des temps de détection mesurés pour un même temps TMD qu'avec le mécanisme Push.

Nous allons maintenant étudier les temps de détection observés avec le mécanisme de type APull dans les graphiques 3.11a,3.11b,3.11c et 3.11d.

Avec ce mécanisme, nous observons qu'il est possible dans une majorité de cas de détecter un incident en un temps inférieur au TMD lorsque celui-ci est de l'ordre de quelques centaines de millisecondes, en fonction de la configuration du réseau sur lequel est déployé le mécanisme. Il apparaît que plus la distance entre les noeuds ($D+V$) est importante et plus le temps minimum de détection sera grand. Cependant, nous voyons que même lorsque le TMD est grand, certaines détections seront tardives, car effectuées en un temps supérieur à ce TMD. Ce phénomène est plus marqué lorsque le réseau est dégradé ($A = 0,99$). La répartition des temps de détection est importante pour un même TMD et, lorsque le TMD est suffisamment grand, la majorité des temps de détection sont dont de durée deux fois moindre que le TMD.

De même qu'avec le mécanisme Pull, bien que les paramètres du mécanisme APull aient été dégradés lorsque le TMD demandé est faible, les temps de détection de ce mécanisme ne peuvent être arbitrairement bas et sont au minimum de quelques centaines de millisecondes, en fonction de la configuration du réseau. Avec le mécanisme APull, les temps de détection sont très variables pour un même TMD. En effet, comme montré par la formule 3.4, le temps pour détecter un incident est très lié à la valeur de T_WAIT calculé par le mécanisme lors de la réception du dernier message RESPONSE. Puisque cette valeur dépend de l'ensemble des échanges de messages effectué avant l'apparition de l'incident, il n'est pas étonnant que le temps de détection final soit très variable.

Dans cette section, nous avons étudié les temps de détection d'un incident pouvant être atteint par les différents mécanismes en fonction d'un TMD demandé. Il s'est avéré que le mécanisme de type Push est le seul à pouvoir détecter un incident en un temps inférieur à 100 millisecondes. De plus, quel que soit le TMD, nous avons observé que l'ensemble des détections était effectué en un temps inférieur au TMD. Enfin, le mécanisme Push n'est pas sensible aux conditions réseau de délai, de gigue et de perte. Par conséquent, il apparaît sur ces points plus performants que les mécanismes Pull et APull. En effet, le mécanisme Pull, bien que peu sensible aux conditions réseaux, ne permet les détections non tardives d'un incident que lorsque le TMD est supérieur à 1 seconde. Le mécanisme APull permet des détections plus rapides. Celles-ci peuvent être effectuées en un temps inférieur au TMD dès lors que celui-ci est inférieur à quelques centaines de millisecondes, en fonction des conditions du réseau. Les performances du mécanisme APull sont en effet sensibles aux conditions réseau et se dégradent si celles-ci sont mauvaises. De plus, même lorsque le TMD est grand, il arrive que le temps de détection d'un incident soit supérieur à ce temps.

Nombre de faux positifs

Nous allons maintenant nous intéresser au nombre de faux positifs observés lors de l'utilisation des différents mécanismes, en fonction des TMD demandés et des configurations de réseau. Les graphiques 3.12,3.13 et 3.14 présentent, pour différentes configurations réseaux, le nombre de faux positifs observés par minute, en fonction du TMD demandé au mécanisme. De plus, sur chaque courbe décrivant le nombre de faux positifs est indiquée la limite à partir de laquelle le taux de détection tardive est inférieur à 50 % et à 5 %. Ceci permet de renseigner les valeurs de TMD à partir desquelles le mécanisme peut détecter un incident en un temps inférieur aux objectifs fixés.

Le nombre de faux positifs observés pour le mécanisme Push en fonction du TMD est indiqué dans le graphique 3.12. 4 configurations réseau sont étudiées. Nous pouvons observer que le nombre de faux positifs mesurés par minute diffère selon la configuration réseau. Lorsque les pertes dans le réseau sont peu importantes ($A = 0,999$) le nombre de faux positifs est important lorsque le TMD est inférieur à environ 100 ms, puis il reste stable à environ un faux positif toutes les 20 minutes (5.10^{-2} faux positifs par minutes) lorsque le TMD est compris entre 100 ms et 1000 ms et il diminue plus fortement lorsque le TMD est supérieur à 1 seconde. Un comportement similaire est observé lorsque

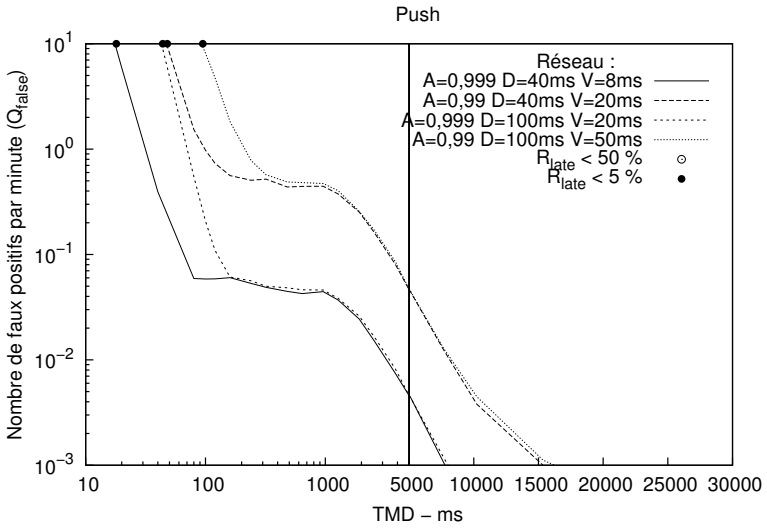


FIG. 3.12: Évaluation du nombre de faux positifs du mécanisme Push, en fonction du temps TMD demandé

le réseau est dégradé ($A = 0,99$) : le nombre de faux positifs est élevé si le TMD est inférieur à 100 ms puis se stabilise à un faux positif toutes les deux minutes si le TMD est compris entre quelques centaines de millisecondes et 1 seconde. Il diminue ensuite lorsque le TMD devient plus grand. Il faut un TMD d'environ 5 secondes pour observer un nombre de faux positifs comparable à celui observé dans le réseau non dégradé. On observe de plus que nombre de faux positif est légèrement inférieur lorsque le délai d'acheminement entre les noeuds est faible ($D = 40ms$). Enfin, le taux de détection tardive, indiqué dans le graphe à l'extrémité des courbes, est négligeable quel que soit le TMD (celui-ci est en fait toujours nul, comme indiqué précédemment dans les graphiques 3.9a,3.9b,3.9c et 3.9d).

On peut déduire de ce graphique que le mécanisme de type Push peut être utilisé pour détecter un incident de manière fiable à partir d'un TMD d'une centaine de millisecondes, lorsque les conditions réseau ne sont pas trop dégradées. En effet, le nombre de faux positifs alors observé est très bas et les taux de détections tardives sont nuls. Lorsque le réseau est dégradé, les taux de faux positifs sont plus élevés, et il faut que le TMD soit d'au moins quelques secondes pour ne pas voir apparaître de faux positifs avant 10 minutes en moyenne.

Le graphique 3.13 représente le nombre de faux positifs observés pour le mécanisme Pull, en fonction du TMD et pour 4 configurations réseaux. Nous observons que le nombre de faux positifs diffère selon la configuration réseau. Lorsque les pertes dans le réseau sont peu importantes ($A = 0,999$), le nombre de faux positifs est d'environ 10^-1 par minute si le TMD est inférieur à 400 ms, puis il se stabilise autour de 4.10^{-2} (un faux positif toutes les 25 minutes) lorsque le TMD est compris en 400 ms et 1500 ms et enfin il diminue rapidement pour les plus grands TMD. Lorsque le réseau est dégradé ($A = 0,99$), le nombre de faux positifs par minute est plus important : il d'environ $0,5$ lorsque le TMD est inférieur à 1 seconde et décroît rapidement ensuite, mais reste supérieur à 10^{-3} même lorsque le TMD est de quelques secondes. Il faut un TMD d'environ 5 secondes pour observer un nombre de faux positifs comparable à celui observé dans le réseau non dégradé. On observe enfin que le nombre de faux positifs ne dépend pas du délai d'acheminement (D=40ms ou D=100ms) entre

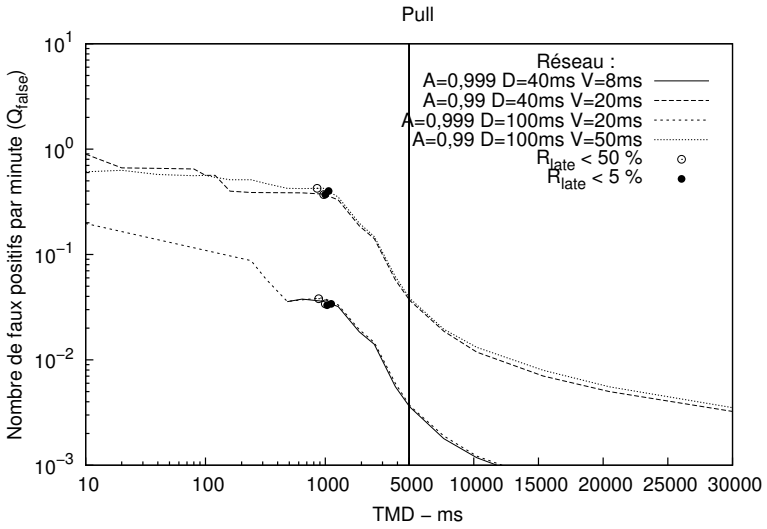


FIG. 3.13: Évaluation du nombre de faux positifs du mécanisme Pull, en fonction du temps TMD demandé

les noeuds sur lesquels le mécanisme Pull est déployé. Le taux de détection tardive, comme il a été précédemment observé dans les graphiques 3.9a,3.9b,3.9c et 3.9d, est supérieur à 50% si le TMD est inférieur à environ 1 seconde et inférieur à 5% lorsque le TMD est supérieur à 1 seconde.

On peut déduire de ces observations que le mécanisme de type Pull ne peut être utilisé pour détecter un incident de manière fiable seulement avec un TMD supérieur ou égal à une seconde. Bien que pour des valeurs de TMD inférieures, la quantité de faux positifs observés soit assez faible lorsque le réseau n'est pas dégradé, le mécanisme de type Pull ne permet pas de détecter un incident en un temps aussi court. Cependant, à partir d'un TMD d'une seconde, le mécanisme de type Pull permet de détecter un incident de manière fiable.

Le nombre de faux positifs en fonction du TMD, pour le mécanisme APull, est représenté dans le graphique 3.14. Ici encore, selon la configuration du réseau, le nombre de faux positifs observé est différent. Pour l'ensemble des configurations, le nombre de faux positifs est constant lorsque le TMD est inférieur à 650 ms environ. Dans ce cas, le nombre de faux positifs est d'environ 2 par minute si le réseau est dégradé et que le délai d'acheminement entre les noeuds est faible ($A = 0,99$ et $D = 40ms$), de $1,1$ par minute si le réseau est dégradé et le délai d'acheminement plus important ($A = 0,99$ et $D = 100ms$), de $0,4$ par minute si le réseau est non dégradé et le délai faible ($A = 0,999$ et $D = 40ms$) et de $0,2$ par minute si le réseau est non dégradé et le délai plus important ($A = 0,999$ et $D = 100ms$). Pour des valeurs de TMD supérieures, le nombre de faux positifs décroit : Pour un TMD égal à une seconde, il est inférieur à 1 par minute lorsque le réseau est dégradé et à 0,1 par minute lorsqu'il ne l'est pas. Pour obtenir un nombre de faux positifs inférieur à 1 par heure, il faut utiliser un TMD d'au moins 5 secondes lorsque le réseau est non dégradé. Si le réseau est dégradé, de grandes valeurs de TMD, de l'ordre de la demi-minute doivent être utilisées. Le taux de détection tardive est plus dépendant des conditions du réseau qu'avec les autres mécanismes. De plus, le taux de détection tardive est plus faible lorsque le délai d'acheminement entre les noeuds du réseau est faible ($D = 40ms$). Dans ce cas, le TMD minimal à utiliser pour obtenir un taux R_{late} inférieur à 5%

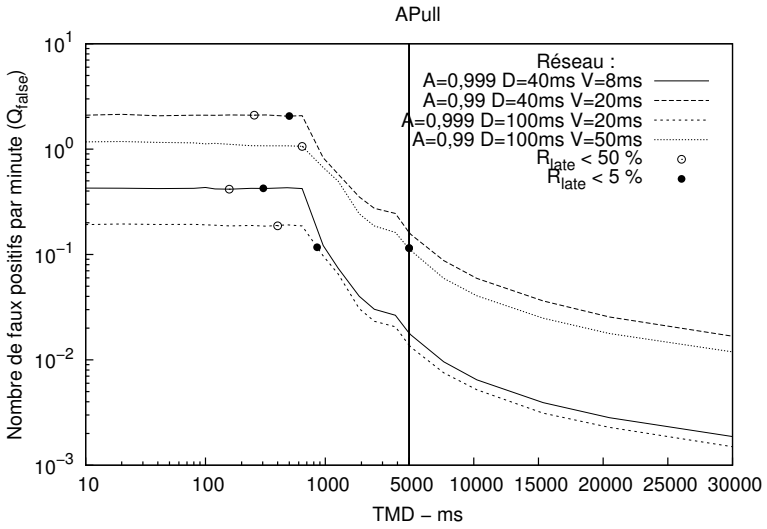


FIG. 3.14: Évaluation du nombre de faux positifs du mécanisme APull, en fonction du temps TMD demandé

est de quelques centaine de milli-secondes, tandis qu'il est d'au moins une seconde lorsque le délai d'acheminement entre les noeuds est plus important ($D = 100ms$).

Nous constatons que le mécanisme de type APull peut être utilisé pour détecter un incident de manière fiable pour un TMD de quelques centaines de millisecondes à une seconde. La valeur minimale de TMD à utiliser pour obtenir des performances acceptables avec ce mécanisme varie en fonction des conditions du réseau. Cependant, même lorsqu'un TMD important est utilisé, le nombre de faux positifs observés reste non négligeable, en particulier si le réseau est dégradé.

Nous avons pour l'instant étudié le nombre de faux positifs que l'on risque d'observer, pour chacun des mécanismes configurés pour un TMD donné. Nous avons défini dans la section 4.4.1 ce qu'était un faux positif pour un certain TMD : nous considérons qu'un faux positif apparaît lorsque le mécanisme alerte de la présence d'un incident alors qu'aucun incident n'est présent ou bien qu'un incident est présent, mais que sa durée de persistance est inférieure au TMD. Cependant, ces situations ne sont pas identiques en terme de « gravité » du faux positif : il est souvent plus gênant de déclarer un incident alors qu'aucun incident n'est présent dans le réseau que lorsqu'un incident est présent, mais que sa durée est inférieure au TMD, en particulier si cette durée est proche du TMD. En effet, l'action consécutive au déclenchement de l'alerte, bien que non nécessaire (c'est un faux positif qui l'a entraînée), ne sera peut-être pas inutile puisqu'un incident est tout de même présent dans le réseau. Par conséquent, nous avons décidé de mesurer la répartition des durées d'incidents qui déclenchent les faux positifs.

Le graphique 3.15 présente, pour chaque mécanisme, la répartition des rapports entre la durée d'un incident et le TMD, pour chaque faux positif observé. Un rapport de 0 signifie qu'aucun incident n'était présent dans le réseau lorsque le faux positif a été déclaré, tandis qu'un rapport appartenant à l'intervalle $]0;100[$ indique qu'un incident était présent dans le réseau, mais que sa durée était inférieure au TMD. Nous observons que pour les mécanismes Push et APull, plus du tiers des faux positifs sont déclarés alors qu'aucun incident n'était présent dans le réseau. Ce n'est pas le cas pour le

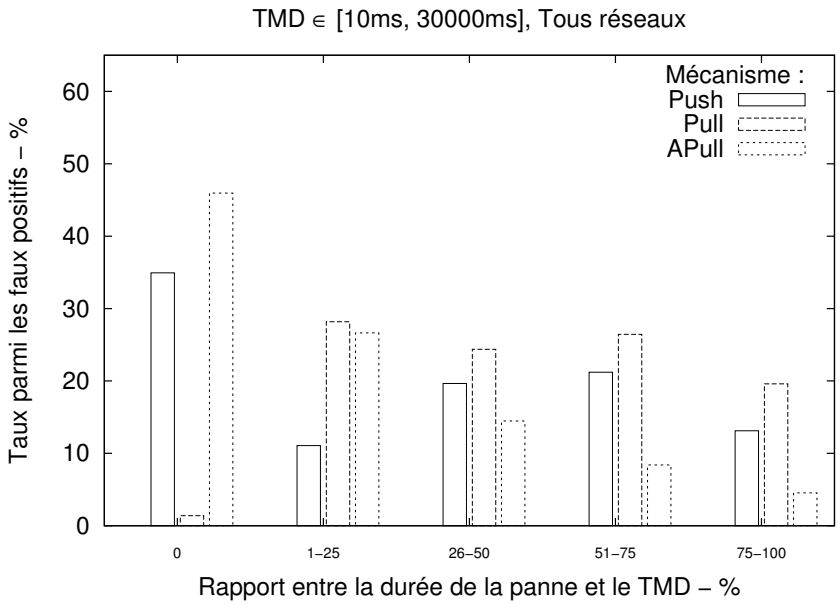


FIG. 3.15: Répartition de la durée d'un incident, en cas de faux positif, pour les différents mécanismes

mécanisme Pull, où le nombre de faux positifs déclarés alors qu'aucun incident n'est présent dans le réseau est quasi nul. Lorsqu'un incident existe, sa durée par rapport au TMD est plus variable pour les mécanismes de type Push et Pull alors qu'elle semble plus souvent courte pour le mécanisme APull.

Ces observations mettent en évidence une qualité du mécanisme Pull. En effet, nous pouvons constater que lorsque ce mécanisme alerte de la présence d'un incident, dans la grande majorité des cas, un incident est présent dans le réseau. À l'inverse, le mécanisme APull déclare souvent des faux positifs alors qu'aucun incident, ou un incident de courte durée, ne sont présents dans le réseau.

Dans cette partie, nous avons étudié la fréquence d'apparition des faux positifs, pour les différents mécanismes. Nous avons observé que le mécanisme de type Push est le seul qui n'entraîne pas un grand nombre de faux positifs lorsque le TMD est de quelques millisecondes. Pour un TMD de l'ordre d'une demi-seconde, le mécanisme APull peut aussi être utilisé, si les conditions du réseau ne sont pas trop dégradées. Le mécanisme Pull est le mécanisme ayant montré les meilleures performances en ce qui concerne le nombre de faux positifs : leurs fréquences d'apparition est la plus faible parmi les mécanismes étudiés et il ne déclenche que rarement d'alertes si aucun incident n'est présent dans le réseau. Cependant, ceci n'est possible que pour des TMD supérieurs à 1 seconde.

Consommation en bande passante

Nous allons enfin étudier le nombre de messages nécessaires aux mécanismes pour fonctionner, en fonction du TMD et pour différentes configurations de réseau. Les graphiques 3.16, 3.17 et 3.18 présentent, pour différentes configurations réseaux, le nombre de messages envoyés dans le réseau par le mécanisme en fonction du TMD demandé. Sur l'axe de droite est représentée la bande passante consommée, en kbit/s : nous avons pour cela fixé la valeur de 28 octets par message envoyé dans le réseau, ce qui correspond à la taille d'un paquet IPv4 contenant un message ICMP de type « Echo Request », sans données ajoutées.

Pour le mécanisme de type Push, on observe dans le graphique 3.16 que le nombre de messages

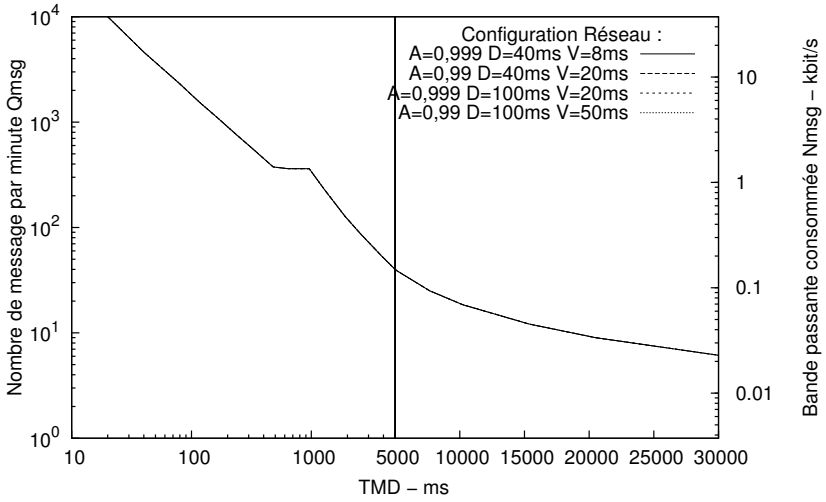


FIG. 3.16: Consommation en bande passante du mécanisme Push, en fonction du temps TMD demandé

utilisés par le mécanisme est proportionnel au TMD demandé (sauf lorsque le TMD est compris entre 500 ms et 1000 ms, voir la section 3.4.1. De plus, cette quantité ne dépend pas des conditions réseau. Ceci est normal, puisque le mécanisme Push envoie des messages à intervalle de temps constant. Par exemple, on observe que le mécanisme Push utilise environ 940 messages par minute (3,5 kbit/s) lorsque le TMD est de 200 ms, 345 messages par minute (1,3 kbit/s) lorsque le TMD vaut 1 seconde.

Pour le mécanisme de type Pull, le graphique 3.17 nous indique que le nombre de messages utilisé par le mécanisme dépend des conditions réseau. Il apparaît que plus le délai d'acheminement moyen est faible ($D+V$) et plus le nombre de messages envoyés est grand. En effet, lorsque le TMD vaut 1 seconde, le nombre de messages par minute observés varie de 308 (1,1 kbit/s) à 413 (1,5 kbit/s). Cet écart devient moins significatif lorsque le TMD augmente. De plus, à partir d'un TMD de 1 seconde, le nombre de messages utilisés décroît rapidement.

Le graphique 3.18 nous indique que, de même que pour le mécanisme Pull, le nombre de messages utilisés par le mécanisme APull dépend des conditions du réseau : plus le délai d'acheminement moyen est faible ($D+V$) et plus le nombre de messages envoyés est grand. À partir d'un TMD d'environ 650 ms, le nombre de messages utilisés décroît rapidement. Par exemple, pour la configuration réseau $A = 0,999, D = 100ms, V = 20ms$, le nombre de messages par minute passe de 800 (3,0 kbit/s) pour un TMD de 650 ms à 318 (1,2 kbit/s) pour un TMD de 1000 ms.

Les observations effectuées dans cette section montrent que le nombre de messages utilisés par les mécanismes de détection d'incident dépend du TMD demandé. En effet, plus le TMD est grand et plus le nombre de messages nécessaires est faible. De plus, ce nombre dépend aussi des conditions réseau pour les mécanismes de type Pull et APull : plus le délai d'acheminement des messages entre les noeuds est faible et plus le nombre de messages envoyés sera important. Cette observation n'est pas surprenante puisque ces mécanismes attendent la réception d'un message RESONSE, avant d'en envoyer un nouveau après une durée T_REQ. Par conséquent plus le temps d'acheminement du message réponse est court et plus la fréquence d'envoi des messages sera importante. Nous avons observé

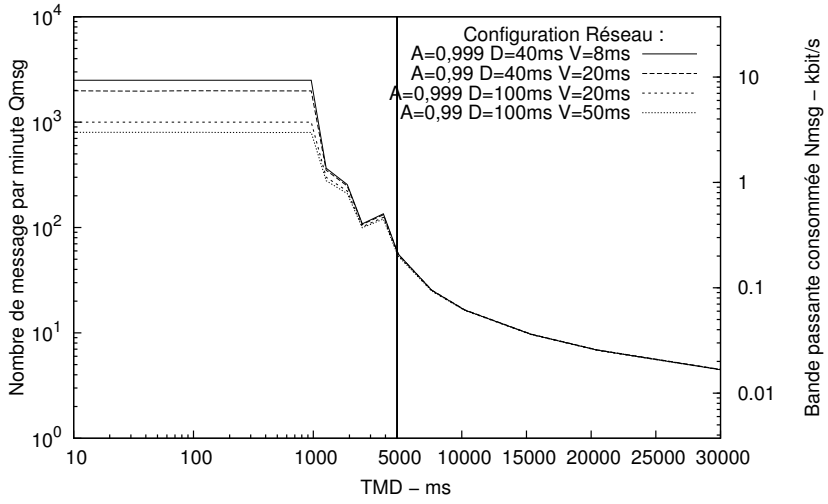


FIG. 3.17: Consommation en bande passante du mécanisme Pull, en fonction du temps TMD demandé

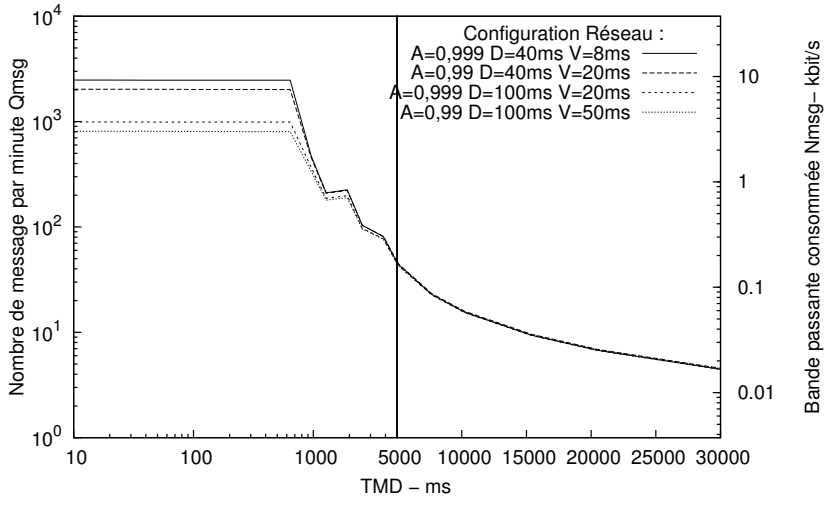


FIG. 3.18: Consommation en bande passante du mécanisme APull, en fonction du temps TMD demandé

TAB. 3.4: Récapitulatif des performances des mécanismes de détection

		TMD (ms)				
Critère	**Mécanisme**	100	500	1000	2500	5000
Q_{false} (faux/min)	Push	0,059	0,045	0,042	0,015	0,0046
	Pull	x**	0,36	0,036	0,014	0,0037
	APull	0,43	0,43	0,12	0,030	0,018
R_{late} (%)	Push	0,00	0,00	0,00	0,00	0,00
	Pull	99,3	100	71*	0,00	0,00
	APull	97,1	0,33	0,02	0,05	0,00
Bande passante (kbit/s)	Push	6,8	1,4	1,3	0,32	0,15
	Pull	9,3	9,3	8,3*	0,40	0,24
	APull	9,2	9,2	1,6	0,38	0,18

* : Avec la configuration du mécanisme Pull utilisée, de 1000 ms constitue une valeur limite pour le mécanisme. Par exemple, le taux de non-détection chute à 0 dès que l'on utilise un TMD supérieur de quelques dizaines de millisecondes, en fonction de la configuration du réseau.

** : Aucune mesure possible pour cette configuration

que le nombre de messages utilisés par le mécanisme de type APull est légèrement inférieur à celui utilisé par le mécanisme de type Pull, à TMD équivalent. Le mécanisme de type APull est donc plus performant en terme de consommation de bande passante que le mécanisme Pull. Le mécanisme de type Push est quant à lui moins consommateur de message que les autres mécanismes lorsque le TMD est inférieur à environ 7,5 secondes, et légèrement plus consommateur sinon.

3.5.2 Discussion des performances des mécanismes

Nous allons maintenant discuter de façon générale des performances des mécanismes de détection d'incident que nous avons étudiés dans ce chapitre. Nous allons tout d'abord rappeler les performances mesurées dans cette partie de manière synthétique. Nous comparerons ensuite les mécanismes entre eux en discutant de leurs avantages et inconvénients.

Récapitulatif des performances

Dans cette section, nous allons tout d'abord synthétiser les performances mesurées pour l'ensemble des mécanismes, et nous les comparerons. Le tableau 3.4 récapitule les différents critères de performances étudiés pour chaque mécanisme, en fonction de différentes valeurs de TMD. Pour les résultats présentés, nous avons utilisé la configuration de réseau suivante : $A = 0,999; D = 40ms; V = 8ms$.

Avantages et inconvénients

Nous allons maintenant discuter des avantages et des inconvénients des différents mécanismes.

Mécanisme Push

Nous avons vu que le mécanisme de type Push est le seul capable de détecter un incident avec un TMD de l'ordre de la centaine de millisecondes. En effet, le taux de détection tardive est nul avec ce mécanisme, et ce quel que soit le TMD et le nombre de faux positifs est faible, environ un toutes les 20 minutes, pour une détection en un temps si court. Enfin, la bande passante utilisée, 6.8 kbit/s pour un TMD de 100 ms, bien qu'élevée, est, elle aussi, acceptable.

Pour les TMD plus élevés, le choix du mécanisme Push est tout à fait acceptable. En effet, la consommation de bande passante est la plus basse mesurée parmi les différents mécanismes lorsque le TMD est inférieur à 7,5 secondes. Les faux positifs sont peu fréquents, y compris lorsque le réseau est dégradé : pour un TMD supérieur à 1 seconde, on observe généralement moins d'un faux positif toutes les dix minutes en moyenne.

Un autre avantage pour le mécanisme Push est qu'il est peu sensible aux conditions réseau, sauf pour le risque d'apparition de faux positifs qui croît si le réseau est dégradé. Par conséquent, son comportement est prédictible et il est inutile de connaître les conditions de réseau qui relie les noeuds sur lesquels il est déployé.

Cependant, le plus gros inconvénient du mécanisme de type Push est issu de son mode de fonctionnement : les messages utilisés pour la détection réseau sont unidirectionnels. Ceci entraîne deux inconvénients :

– Une session doit être établie pour mettre en place le mécanisme ;
– Les incidents n'affectant qu'un certain sens d'une communication ne sont pas détectés.

Le premier inconvénient est dû au fait qu'avec le mécanisme de type Push, le noeud qui détecte l'incident n'est pas celui qui envoie les messages. Par conséquent, le noeud voulant mettre en place le mécanisme doit signaler à son correspondant qu'il faut qu'il démarre l'envoi de message HELLO et l'intervalle T_HELLO entre deux envois successifs. Ainsi, le correspondant doit être prêt à recevoir et retenir de telles requêtes. Ce n'est pas le cas avec les mécanismes Pull et APull, car les correspondants de ces mécanismes se contentent de répondre immédiatement lors de la réception d'un message. Ils n'ont donc pas besoin de maintenir de session.

Le second inconvénient est la conséquence du trajet unidirectionnel des messages HELLO. En effet, les messages HELLO parcourent le réseau en provenance du noeud « correspondant » et à destination du noeud qui détecte l'incident. Par conséquent, si incident affecte uniquement les communications dans le sens allant du noeud qui détecte l'incident vers le noeud correspondant, celui-ci ne sera pas détecté. Ceci peut par exemple se produire lorsque les chemins dans le réseau IP ne sont pas les mêmes pour les deux sens de communication, c'est-à-dire qu'ils empruntent des routes différentes dans le sens « aller » et « retour ». Pour permettre la détection « bidirectionnelle », il suffirait cependant de déployer le mécanisme Push sur les deux à la fois, afin qu'ils soit émetteurs et récepteurs de message HELLO. Ceci aurait pour conséquence de doubler le nombre de messages transitant dans le réseau, ce qui en ferait le mécanisme le plus consommateur en bande passante étudié.

Mécanisme Pull et APull

Nous allons discuter des avantages communs aux deux mécanismes Pull et APull. En plus de ne pas être soumis aux inconvénients dus à la nature unidirectionnelle du mécanisme Push, l'utilisation d'un échange de messages de type REQUEST/RESPONSE par ces mécanismes permet d'envisager les avantages suivants :

– Les messages peuvent être mis à profit pour obtenir des informations sur le réseau
– Il est possible d'utiliser le protocole ICMP pour déployer ces mécanismes.

Le premier avantage permet en particulier de connaître en permanence le délai d'acheminement

aller-retour entre les noeuds sur lesquels le mécanisme est déployé. Cette information est parfois utile pour par exemple, estimer d'autres paramètres sur la qualité du lien.

Le second avantage est lié à la nature des mécanismes de type Pull et APull : la complexité est du côté du noeud qui détecte l'incident et la seule tache du noeud qui reçoit un message REQUEST est de retourner immédiatement un message REPSONSE. Par conséquent, il est possible d'utiliser des messages ICMP de type Echo Request et Echo Response avec le mécanisme de détection d'incident. Il est ainsi possible de déployer le mécanisme de détection d'incident pour détecter les incidents vers un grand nombre de noeuds, sans configuration préalable de ceux-ci, puisque la plupart des machines connectées à un réseau IP implémentent le protocole ICMP.

La différence entre les mécanismes Pull et APull est assez importante. Nos résultats ont montré que le mécanisme APull pouvait détecter des incidents avec un TMD de quelques centaines de millisecondes, tandis que le protocole de type Pull ne pouvait détecter un incident qu'à partir d'un TMD d'une seconde. Par conséquent, lorsque le TMD est inférieur à 1 seconde, le mécanisme APull est le seul à pouvoir être utilisé. Cependant, ce dernier présente d'autres inconvénients en comparaison du protocole Pull. Plus particulièrement, le nombre de faux positifs et le taux de détection tardive sont élevés avec ce mécanisme, surtout lorsque les conditions réseaux sont dégradées et même pour des TMD élevés. À l'inverse, le mécanisme Pull offre des taux de détection tardive nuls et des fréquences d'apparition de faux positifs basses, et souffre moins d'un réseau dégradé que le mécanisme APull. Par conséquent, et même si le mécanisme APull est légèrement moins consommateur de bande passante que le mécanisme Pull, il semble difficile de recommander son utilisation si le TMD demandé est supérieur à 1 seconde.

3.6 Validation expérimentale

Nous allons maintenant valider les résultats obtenus par les simulations et présentés dans la section précédente en effectuant des mesures expérimentales. Nous avons ainsi implémenté les différents mécanismes de détection d'incident étudiés et avons réalisé de mesures pour vérifier la validité des résultats de simulation.

Nous avons effectué des mesures dans deux catégories de réseau de test : un réseau virtualisé et le réseau Internet.

3.6.1 Réseau virtualisé

Nous avons décidé d'utiliser un réseau virtualisé de façon à reproduire de façon déterministe les conditions de réseau modélisées dans nos simulations. Ainsi, avec la virtualisation, il a été possible de reproduire le comportement du réseau simulé, tel que décrit dans la section 3.3.3. L'ensemble des paramètres, délai minimum d'acheminement (D), délai variable d'acheminement (V), MTTF et MTTR ont ainsi été reproduit. Nous avons utilisé le système « OpenVZ » [69] pour la virtualisation ainsi que l'outil « Linux Traffic Control » [29] pour contrôler le délai dans le réseau.

Les caractéristiques du réseau utilisé pour la simulation et pour le réseau virtualisé étant identiques, il va être possible de vérifier si les résultats de performance obtenus par simulation sont corrects, pour des conditions de réseau équivalentes.

Les tableaux 3.6, 3.7 et 3.8 présentent la répartition des temps de détection obtenus dans le réseau virtualisé, par rapport aux temps de détection obtenus avec la simulation. Les résultats sont présentés en fonction du TMD et pour les différentes configurations de réseaux. La table 3.5 présente les configurations utilisées dans le réseau virtualisé lors des expérimentations. Pour un TMD et une configu-

TAB. 3.5: Paramètres utilisés pour les expérimentations dans le réseau virtualisé

	Réseau :			
Paramètres :	1	2	3	4
D (ms)	40	40	100	100
V (ms)	8	20	20	50
A	0,999	0,99	0,999	0,99

ration de réseau donnée, chaque ligne indique le pourcentage de détection dans le réseau virtualisé dont le temps est supérieur au temps maximum mesuré dans la simulation ($>$ Max), compris entre le quartile supérieur et le temps maximum mesuré dans la simulation (Q++), compris entre la valeur médiane et le quartile supérieur mesuré dans la simulation (Q+), compris entre le quartile inférieur et la valeur médiane mesurée dans la simulation (Q-), compris entre la valeur minimum et le quartile inférieur mesuré dans la simulation (Q- -) et inférieur à la valeur inférieure mesurée dans la simulation ($<$ Min).

On observe que la répartition des temps de détection obtenus dans le réseau virtualisé est proche de celle obtenue par simulation. En effet, il est très rare que les temps de détection observés dans le réseau virtualisé soient supérieurs ou inférieurs aux valeurs maximales ou minimales des temps de détection obtenus par simulation. Lorsque cela se produit (le plus souvent avec le mécanisme Pull), c'est pour des valeurs de TMD trop basse pour être utilisées avec le mécanisme concerné. La majorité des temps de détection observés dans le réseau virtualisé coïncident avec les temps obtenus par simulation. On peut cependant noter que les temps observés avec le mécanisme Pull se répartissent légèrement dans la médiane supérieure des temps obtenus par simulation, alors que les temps observés pour le mécanisme APull se répartissent plutôt sur la médiane inférieure. Avec le mécanisme Push, les temps observés dans le réseau virtualisé semblent plus équitablement répartis.

On peut donc constater que les temps de détection obtenus par simulation sont proches de ceux pouvant être obtenus en expérimentation réelle, dans un réseau virtualisé. La plage de temps de détection obtenus par simulation est semblable à celle observée dans l'expérimentation, pour des valeurs de TMD suffisamment grande pour être utilisées par le mécanisme. Comparées aux valeurs obtenues par expérimentation, les valeurs obtenues par simulation diminuent légèrement le temps de détection du mécanisme Pull et augmentent légèrement le temps de détection du mécanisme APull, mais de façon trop peu importante pour remettre en cause la validité de la simulation.

Nous allons maintenant étudier les résultats expérimentaux des performances des mécanismes. Les tableaux 3.9, 3.10 et 3.11 présentent le nombre de faux positifs par minute, le taux de détection tardive et la bande passante utilisée obtenus dans le réseau virtualisé et dans les résultats de simulation. Les résultats sont présentés pour les différentes configurations de réseaux, identiques à celles utilisées dans les tableaux précédents. Pour chaque ligne du tableau, qui indique une valeur de TMD, les résultats sont présentés sous la forme « valeur obtenue par expérimentation / valeur obtenue par simulation ».

On observe que les performances obtenues dans le réseau virtualisé sont similaires à celles obtenues par simulation. En effet, aucun écart significatif n'est observé avec le mécanisme Push. De plus, les seuls écarts importants ne sont constatés pour les autres mécanismes que pour des valeurs de TMD basses, pour lesquels le mécanisme ne peut pas être utilisé dans de bonnes conditions. C'est en particulier le cas pour la bande passante utilisée, qui est expérimentalement plus faible avec les mécanismes de type Pull et APull pour des valeurs de TMD basses qu'elle ne l'est prédite par simulation. En effet avec ces mécanismes, si la détection n'est pas effectuée dans de bonnes conditions, en cas

TAB. 3.6: Répartition des temps de détection observés avec l'expérimentation dans le réseau virtualisé, en pourcentage, par rapport aux mesures obtenues par simulation, pour le mécanisme Push.

TMD (ms)	40				100				240			
Réseau :	1	2	3	4	1	2	3	4	1	2	3	4
> Max	1	1	0	2	1	0	1	2	2	2	1	3
Q++	25	18	18	27	27	26	29	17	23	20	27	19
Q+	22	31	31	29	30	25	24	22	25	27	36	27
Q-	29	22	27	19	18	21	19	26	22	23	16	24
Q−−	22	26	24	20	21	25	27	33	28	26	17	27
< Min	1	2	0	3	3	3	0	0	0	2	3	0
TMD (ms)	480				960				1280			
Réseau :	1	2	3	4	1	2	3	4	1	2	3	4
> Max	1	1	0	2	1	1	1	2	1	0	2	1
Q++	20	21	25	25	26	18	20	26	25	25	29	24
Q+	25	29	33	27	18	28	30	28	23	24	27	32
Q-	20	20	21	21	32	22	19	16	27	20	20	15
Q−−	33	29	20	24	23	30	27	26	24	30	22	28
< Min	1	0	1	1	0	1	3	2	0	1	0	0
TMD (ms)	2560				5120				10240			
Réseau :	1	2	3	4	1	2	3	4	1	2	3	4
> Max	0	1	2	0	4	1	1	2	2	2	0	24
Q++	25	25	25	29	19	22	24	28	30	29	25	0
Q+	21	26	23	25	22	26	20	30	19	24	31	0
Q-	26	19	20	14	24	25	33	21	22	15	23	55
Q−−	25	27	29	28	30	25	22	18	24	29	20	0
< Min	3	2	1	4	1	1	0	1	3	1	1	21

TAB. 3.7: Répartition des temps de détection observés avec l'expérimentation dans le réseau virtualisé, en pourcentage, par rapport aux mesures obtenues par simulation, pour le mécanisme Pull.

TMD (ms)	40				100				240			
Réseau :	1	2	3	4	1	2	3	4	1	2	3	4
$>$ Max	0	0	0	0	0	50	x	0	22	33	11	15
Q++	50	0	0	0	100	0	x	0	33	11	22	21
Q+	33	0	0	0	0	0	x	0	22	27	27	31
Q-	0	100	0	0	0	50	x	50	16	11	22	5
Q$--$	16	0	0	100	0	0	x	50	5	16	16	26
$<$ Min	0	0	100	0	0	0	x	0	0	0	0	0
TMD (ms)	480				960				1280			
Réseau :	1	2	3	4	1	2	3	4	1	2	3	4
$>$ Max	37	21	15	12	32	20	12	18	5	1	5	3
Q++	27	19	20	19	28	39	32	30	32	25	23	27
Q+	19	25	24	31	20	24	19	20	23	24	23	27
Q-	15	21	26	23	15	11	28	6	15	31	28	27
Q$--$	0	11	13	12	3	3	7	26	23	16	19	14
$<$ Min	0	0	0	0	0	0	0	0	0	0	0	0
TMD (ms)	2560				5120				10240			
Réseau :	1	2	3	4	1	2	3	4	1	2	3	4
$>$ Max	1	0	0	0	1	1	0	0	0	0	0	0
Q++	38	25	34	21	34	26	21	22	23	35	30	31
Q+	23	26	25	27	23	35	34	27	30	28	21	28
Q-	18	23	23	34	15	16	27	35	25	18	21	24
Q$--$	18	25	16	16	25	18	16	14	20	16	26	15
$<$ Min	0	0	0	0	0	0	0	0	0	0	0	0

TAB. 3.8: Répartition des temps de détection observés avec l'expérimentation dans le réseau virtualisé, en pourcentage, par rapport aux mesures obtenues par simulation, pour le mécanisme APull.

TMD (ms)	40				100				240			
Réseau :	1	2	3	4	1	2	3	4	1	2	3	4
$>$ Max	0	0	0	0	0	0	0	0	0	5	0	0
Q++	61	75	0	0	14	25	66	18	40	20	14	42
Q+	23	0	0	25	16	25	33	9	16	42	14	20
Q-	15	25	16	0	34	27	0	27	21	20	31	22
Q$--$	0	0	50	50	36	22	0	45	21	11	38	14
$<$ Min	0	0	33	25	0	0	0	0	0	0	0	0
TMD (ms)	480				960				1280			
Réseau :	1	2	3	4	1	2	3	4	1	2	3	4
$>$ Max	0	5	0	0	0	3	0	3	0	0	0	0
Q++	23	39	26	35	32	29	19	26	19	28	15	28
Q+	18	24	20	24	21	25	15	21	25	23	20	25
Q-	21	13	33	22	23	22	21	22	25	21	22	19
Q$--$	36	17	18	16	23	18	43	26	30	26	41	26
$<$ Min	0	0	0	0	0	0	0	0	0	0	0	0
TMD (ms)	2560				5120				10240			
Réseau :	1	2	3	4	1	2	3	4	1	2	3	4
$>$ Max	0	0	0	0	0	0	0	0	0	0	0	0
Q++	8	30	14	37	15	21	9	12	20	33	16	37
Q+	25	18	21	20	15	17	21	18	23	18	21	21
Q-	20	32	28	25	28	28	30	37	29	16	17	14
Q$--$	43	18	36	16	40	28	38	31	27	31	44	26
$<$ Min	1	0	0	0	0	3	0	0	0	0	0	0

TAB. 3.9: Performances du mécanisme Push mesurées avec l'expérimentation dans le réseau virtualisé, par rapport à celles obtenues par simulation

Q_{false} (faux/min) :

	Réseau			
TMD (ms)	1	2	3	4
40	0.00/0.39	11.21/18.14	13.43/13.85	91.42/81.19
100	0.00/0.06	0.33/0.98	0.12/0.21	5.17/8.14
240	0.00/0.05	0.72/0.51	0.00/0.06	0.65/0.78
480	0.00/0.04	x/0.44	0.00/0.05	x/0.49
960	0.00/0.04	0.61/0.44	0.00/0.05	0.21/0.47
1280	0.00/0.04	0.00/0.38	0.00/0.04	0.31/0.40
2560	0.00/0.02	2.64/0.16	0.00/0.02	0.56/0.17
5120	0.00/0.00	0.00/0.04	0.00/0.00	0.00/0.04
10240	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
30720	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00

R_{late} (%) :

	Réseau			
TMD (ms)	1	2	3	4
40	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
100	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
240	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
480	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
960	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
1280	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
2560	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
5120	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
10240	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
30720	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00

Q_{msg} (message/min) :

	Réseau			
TMD (ms)	1	2	3	4
40	4426.59/4615.30	4867.39/4613.56	4315.14/4620.51	4752.90/4631.10
100	1833.21/1818.16	1814.51/1817.89	1897.19/1818.14	1819.68/1816.85
240	756.86/750.02	752.09/750.17	750.07/749.98	751.02/749.76
480	375.21/375.03	372.71/375.29	376.41/375.01	374.96/375.08
960	357.06/361.48	363.44/361.72	362.04/361.46	361.42/361.53
1280	233.67/230.80	23.97/231.05	230.57/230.79	234.42/230.95
2560	82.40/87.48	87.11/87.61	87.81/87.48	85.42/87.59
5120	38.99/38.97	40.74/39	36.76/38.97	39.24/39
10240	18.53/18.48	18.59/18.49	18.39/18.48	18.64/18.49
30720	6.10/5.96	5.91/5.96	6.18/5.96	5.77/5.96

TAB. 3.10: Performances du mécanisme Pull mesurées avec l'expérimentation dans le réseau virtualisé, par rapport à celles obtenues par simulation

Q_{false} (faux/min) :

	Réseau			
TMD (ms)	1	2	3	4
40	0.00/0.00	0.00/0.66	0.00/0.00	0.00/0.58
100	0.00/0.00	0.00/0.56	0.00/0.00	0.00/0.56
240	0.00/0.00	0.00/0.39	0.00/0.09	0.00/0.51
480	0.00/0.04	0.00/0.39	0.00/0.04	0.00/0.42
960	0.00/0.04	9.41/0.37	0.00/0.04	1.11/0.43
1280	0.00/0.03	0.42/0.33	0.00/0.03	0.00/0.35
2560	0.00/0.01	0.00/0.14	1.03/0.01	0.68/0.15
5120	0.35/0.00	1.99/0.04	0.00/0.00	6.04/0.04
10240	0.00/0.00	1.52/0.01	0.00/0.00	0.00/0.01
30720	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00

R_{late} (%) :

	Réseau			
TMD (ms)	1	2	3	4
40	100/94.95	100/93.90	96.55/100	96.97/99.99
100	100/99.31	100/99.83	100/100	100/100
240	100/100	100/100	100/100	100/99.97
480	100/100	100/99.99	100/99.99	100/99.37
960	100/81.26	88.68/64.13	53.57/33.03	50/26.95
1280	3.85/0.00	1.85/0.00	3.85/0.00	3.64/0.00
2560	1.82/0.00	0.00/0.00	0.00/0.00	0.00/0.00
5120	1.92/0.00	1.89/0.00	0.00/0.00	0.00/0.00
10240	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
30720	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00

Q_{msg} (message/min) :

	Réseau			
TMD (ms)	1	2	3	4
40	554.65/2495.69	409.42/1970.78	231.21/999.19	119.88/800.99
100	843.29/2497.90	533.12/1984.52	x/999.16	119.88/800.64
240	1529.11/2497.37	1196.48/1980.30	644.90/998.89	508.77/798.65
480	1926.53/2496.99	1607.02/1977.33	801.64/998.76	651.27/797.45
960	2198.65/2496.94	1886.28/1977.81	888.70/998.74	751.37/797.69
1280	342.06/365.50	339.18/349.71	282.78/299.73	271.26/278.22
2560	109.28/108.29	108.31/107.04	103.30/101.69	100.86/99.59
5120	55.66/55.35	55.46/55.02	54.09/53.57	53.55/52.89
10240	18.21/16.48	18.21/16.54	18.11/16.31	18.08/16.35
30720	.08/4.32	5.08/4.35	5.06/4.31	5.10/4.34

TAB. 3.11: Performances du mécanisme APull mesurées avec l'expérimentation dans le réseau virtualisé, par rapport à celles obtenues par simulation

Q_{false} (faux/min) :

	Réseau			
TMD (ms)	1	2	3	4
40	0.00/0.42	5.56/2.07	0.00/0.19	0.00/1.16
100	0.00/0.43	7.96/2.10	0.00/0.19	4.17/1.12
240	0.00/0.42	4.23/2.10	0.00/0.19	0.68/1.08
480	0.00/0.43	3.43/2.05	0.64/0.19	1.22/1.08
960	0.29/0.12	1.52/0.83	0.00/0.10	0.85/0.67
1280	0.12/0.07	1.07/0.58	0.00/0.07	0.80/0.50
2560	0.00/0.03	0.55/0.27	0.00/0.02	0.00/0.19
5120	0.20/0.02	0.34/0.16	0.15/0.01	0.00/0.11
10240	0.00/0.01	0.35/0.06	0.05/0.01	0.65/0.04
30720	0.14/00	0.42/0.02	0.16/0.00	0.41/0.01

R_{late} (%) :

	Réseau			
TMD (ms)	1	2	3	4
40	100/99.97	100/99.88	97.62/100	93.33/99.95
100	92.45/97.13	98.11/97.74	100/99.99	100/99.74
240	21.82/13.24	72.22/55.79	98.11/97.99	100/98.31
480	0.00/0.38	10.71/5.74	28.30/29.46	87.04/78.36
960	0.00/0.01	0.00/1.04	5.26/3.54	34.55/28.65
1280	0.00/0.02	1.79/1.15	1.89/4.25	28.57/24.07
2560	0.00/0.05	0.00/1.70	1.79/3.82	37.04/24.38
5120	0.00/0.00	0.00/0.19	0.00/0.35	1.85/5.55
10240	0.00/0.00	0.00/0.11	0.00/0.11	1.79/2.17
30720	0.00/0.00	0.00/0.02	0.00/0.02	0.00/0.53

Q_{msg} (message/min) :

	Réseau			
TMD (ms)	1	2	3	4
40	1469.80/2474.31	1814.83/2020	227.10/990.55	182.85/809.36
100	2127.62/2473.13	1764.96/2017.50	593.93/990.01	564.55/807.35
240	2248.10/2473.29	1877.38/2014.10	833.53/989.54	701.55/803.73
480	2378.19/2473.30	1939.31/2016.35	934.64/989.51	723.56/802.16
960	472.71/471.71	452.08/454.23	357.75/366.56	330.82/337.56
1280	214.97/211.27	210.71/208.69	188.99/187.11	176.85/179.73
2560	110.39/103.43	106.09/103.73	101.42/97.22	95.79/95.79
5120	47.50/43.87	46.74/44.47	45.60/42.68	45.50/42.85
10240	17.67/15.49	17.86/15.79	16.94/15.33	17.48/15.55
30720	5.05/4.32	5.13/4.41	5.05/4.30	5.15/4.38

de détection tardive ou de faux positif, la fréquence d'envoi des messages en sera perturbée, ce qui explique les écarts observés.

Il existe toutefois quelques différences de comportement observé dans l'exploitation expérimentale des mécanismes à souligner. Par exemple, le taux de détection tardive du mécanisme Pull déployé dans l'expérimentation n'est pas tout à fait nul, pour les TMD inférieurs à 5 secondes, contrairement à ce qui avait été montré dans la simulation. De même, le nombre de faux positifs du mécanisme APull est plus élevé dans les mesures expérimentales que ce qui a été mesuré en simulation, pour les réseaux dits « dégradés » (Réseau 2 et 4). Cependant, ces différences sont faibles et nous considérons qu'elles ne remettent pas en question la validité de nos simulations.

À l'aide des expérimentations effectuées dans le réseau virtualisé, nous avons pu montrer qu'avec des conditions de réseaux identiques (même délai d'acheminement et disponibilité dans le réseau), l'implémentation des mécanismes de détection d'incident se comporte de manière similaire à ce qui avait été prédit dans nos simulations. Par conséquent, nous pouvons affirmer que la modélisation du comportement des mécanismes utilisée dans les simulations présentées dans la section précédente est correcte. Afin de valider totalement nos résultats, il convient de démontrer que les conditions réseau utilisées pour nos simulations sont réalistes et n'interfèrent pas avec les résultats présentés. Pour cela, une mesure des performances des mécanismes déployés sur Internet a été effectuée.

3.6.2 Réseau Internet

Afin de s'assurer que la modélisation du comportement du réseau utilisé dans nos simulations était suffisamment réaliste pour permettre une mesure correcte des performances des mécanismes de détection d'incident, il est nécessaire de vérifier le comportement de ces mécanismes en condition réelle, sur le réseau Internet.

Pour cela, nous avons déployé les mécanismes de détection d'incident entre différents noeuds connectés à Internet. Nous avons ensuite effectué des mesures sur le comportement des mécanismes. À l'issu de ces mesures, nous avons été capables d'étudier le comportement du réseau et ainsi d'inférer les valeurs des différents paramètres décrivant le comportement du réseau utilisés dans nos simulations :

- D est choisi comme le plus petit délai mesuré entre les noeuds reliés à Internet.
- V est la différence entre le délai moyen mesuré et D
- A est choisi comme étant le taux de perte, c'est-à-dire le nombre de messages reçus divisé par le nombre de messages envoyés.
- Le MTTR est fixé arbitrairement à 1 seconde et le MTTF est calculé comme précédemment par :

$$MTTF = \frac{A.MTTR}{1-A}$$

Nous avons ainsi pû réaliser des simulations dont le réseau simulé utilise ces paramètres. Nous avons ensuite comparé les résultats obtenus aux résultats des mesures expérimentales. Il est ainsi possible de valider les résultats de simulation présentés dans la section 3.5 mais aussi de s'assurer que la modélisation du comportement d'un réseau utilisée dans les simulations est suffisamment réaliste pour permettre une simulation fidèle des mécanismes de détection d'incident. Deux déploiements sur Internet ont ainsi été réalisés. Le tableau 3.12 montre les différents paramètres inférés des conditions réseau mesurées et utilisés dans les simulations

Les tableaux 3.13, 3.14 et 3.15 présentent la répartition des temps de détection obtenus dans le réseau Internet, par rapport aux temps de détection obtenus avec la simulation. Comme précédemment,

TAB. 3.12: Paramètres des réseaux utilisés dans les expérimentations Internet

	Réseau :	
Paramètres :	1	2
D (ms)	22	161
V (ms)	4	2
A	0,993	0,999

TAB. 3.13: Répartition des temps de détection observés avec l'expérimentation dans le réseau Internet, en pourcentage, par rapport aux mesures obtenues par simulation, pour le mécanisme Push.

TMD (ms)	40		100		240		480		960		1280	
Réseau :	1	2	1	2	1	2	1	2	1	2	1	2
$>$ Max	0	0	0	0	0	2	3	0	0	1	1	0
Q++	24	19	27	31	21	28	25	19	26	20	23	29
Q+	24	23	27	17	28	23	31	30	19	35	22	21
Q-	25	33	21	28	29	24	22	25	25	17	24	22
Q$--$	26	25	24	23	21	22	16	26	30	25	30	28
$<$ Min	1	0	1	1	1	1	3	0	0	2	0	0
TMD (ms)	2560		5120		10240		30720					
Réseau :	1	2	1	2	1	2	1	2				
$>$ Max	1	1	0	0	24	1	1	2				
Q++	23	31	30	32	3	27	25	20				
Q+	20	21	25	29	7	26	22	29				
Q-	31	18	23	20	58	21	29	17				
Q$--$	25	29	21	19	0	25	23	30				
$<$ Min	0	0	1	0	8	0	0	2				

les résultats sont présentés en fonction du TMD et pour les différentes configurations de réseaux, et pour un TMD et une configuration de réseau donnée, chaque ligne indique le pourcentage de détection dans le réseau virtualisé dont le temps est situé dans une plage de quartiles de temps de détection mesurés dans la simulation.

La répartition des temps de détection obtenus dans le réseau Internet est proche de celle obtenue par simulation. En effet, de la même manière à ce qui avait été observé avec le réseau virtualisé, les temps de détection observés dans le réseau Internet sont généralement supérieurs ou inférieurs aux valeurs maximales ou minimales des temps de détection obtenus par simulation. Comme précédemment, les temps de détection observés sur Internet diffèrent le plus de ceux obtenus par simulation pour les valeurs de TMD basses, pour lesquels le mécanisme n'est pas utilisable dans de bonnes conditions. De plus, les temps observés pour le mécanisme APull sont plutôt inférieurs à ceux prédits par la simulation. Ce n'est pas le cas pour les mécanismes Push et Pull, dont les répartitions des temps de détection correspondent à celles obtenues par simulation.

Ceci confirme donc que les temps de détection obtenus par simulation sont proches de ceux pouvant être obtenus en expérimentation réelle. Les valeurs obtenues par simulation ont même tendance à augmenter légèrement les temps de détection du mécanisme APull, ce qui suppose un comportement meilleur pour ce mécanisme en utilisation réelle que ce qui a été présenté dans la section 3.5. On peut considérer que les temps de détection obtenus par simulation sont représentatifs du comportement des

TAB. 3.14: Répartition des temps de détection observés avec l'expérimentation dans le réseau Internet, en pourcentage, par rapport aux mesures obtenues par simulation, pour le mécanisme Pull.

TMD (ms)	40		100		240		480		960		1280	
Réseau :	1	2	1	2	1	2	1	2	1	2	1	2
$>$ Max	0	100	0	6	2	7	3	6	1	12	9	5
Q++	17	0	65	46	11	30	21	31	4	23	15	12
Q+	42	0	22	29	63	30	23	28	3	23	18	40
Q-	30	0	12	11	16	23	40	20	72	21	33	18
Q$--$	10	0	0	8	7	7	12	13	18	18	20	22
$<$ Min	0	0	0	0	0	0	0	0	1	0	1	0

TMD (ms)	2560		5120		10240		30720	
Réseau :	1	2	1	2	1	2	1	2
$>$ Max	3	3	0	0	1	0	0	0
Q++	24	37	29	23	20	30	23	28
Q+	32	16	21	32	29	32	17	26
Q-	10	22	30	23	27	21	30	9
Q$--$	29	20	18	20	20	12	28	35
$<$ Min	0	0	0	0	0	1	0	0

TAB. 3.15: Répartition des temps de détection observés avec l'expérimentation dans le réseau Internet, en pourcentage, par rapport aux mesures obtenues par simulation, pour le mécanisme APull.

TMD (ms)	40		100		240		480		960		1280	
Réseau :	1	2	1	2	1	2	1	2	1	2	1	2
$>$ Max	23	0	39	0	37	0	33	0	31	0	1	0
Q++	35	0	12	7	36	18	22	0	27	3	16	0
Q+	13	0	21	13	11	10	20	23	14	3	28	19
Q-	10	0	18	23	12	25	21	19	12	25	28	23
Q$--$	14	1	9	40	4	45	2	57	13	66	25	57
$<$ Min	4	99	0	17	0	0	2	0	2	0	0	0

TMD (ms)	2560		5120		10240		30720	
Réseau :	1	2	1	2	1	2	1	2
$>$ Max	0	1	0	0	0	0	11	0
Q++	21	0	14	0	26	22	21	24
Q+	18	22	25	21	21	9	11	15
Q-	10	14	30	26	28	22	16	13
Q$--$	49	61	27	50	23	44	39	47
$<$ Min	0	0	1	1	0	1	0	0

TAB. 3.16: Performances du mécanisme Push mesurées avec l'expérimentation dans le réseau Internet, par rapport à celles obtenues par simulation

	Q_{false} (faux/min) :		R_{late} (%) :	
	Réseau			
TMD (ms)	1	2	1	2
40	0.00/0.01	0.15/0.05	0.00/0.00	0.00/0.00
100	0.27/0.00	0.07/0.06	0.00/0.00	0.00/0.00
240	0.03/0.00	0.04/0.06	0.00/0.00	0.00/0.00
480	0.60/0.00	0.78/0.05	0.00/0.00	0.00/0.00
960	0.05/0.00	0.02/0.04	0.00/0.00	0.00/0.00
1280	0.05/0.00	0.07/0.04	0.00/0.00	0.00/0.00
2560	0.36/0.00	0.28/0.01	0.00/0.00	0.00/0.00
5120	0.00/0.00	0.16/0.01	0.00/0.00	0.00/0.00
10240	0.07/0.00	0.04/0.00	0.00/0.00	0.00/0.00
30720	0.07/0.00	0.19/0.00	0.00/0.00	0.00/0.00

	Q_{msg} (message/min) :	
	Réseau	
TMD (ms)	1	2
40	3631.40/4617.60	4724.74/4615.40
100	1810.17/1819.13	1802.31/1818.18
240	755.64/753.08	960.94/750.03
480	378.31/378.21	324.28/375.00
960	367.71/364.29	287.44/361.45
1280	243.46/233.45	231.35/230.78
2560	57.81/88.54	67.21/87.47
5120	43.59/39.40	59.98/38.96
10240	18.54/18.56	16.31/18.48
30720	6.00/5.99	4.73/5.96

mécanismes.

Nous allons maintenant étudier les performances des mécanismes déployés sur Internet. Les tableaux 3.16, 3.17 et 3.18 présentent le nombre de faux positifs par minute, le taux de détection tardive et la bande passante utilisée obtenus dans le réseau virtualisé et dans les résultats de simulation. Les résultats sont présentés pour les différents déploiements. Comme précédemment, pour chaque ligne du tableau, qui indique une valeur de TMD, les résultats sont présentés sous la forme « valeur obtenue par expérimentation / valeur obtenue par simulation ».

On observe que les performances obtenues dans Internet sont globalement similaires à celles obtenues par simulation. De même que pour le réseau virtualisé, les seuls écarts importants ne sont constatés que pour des valeurs de TMD trop basses pour utiliser correctement le mécanisme. On peut cependant noter que pour l'ensemble des mécanismes, le risque d'apparition de faux positifs semble plus important en conditions expérimentales qu'avec les simulations. Cette observation est cependant épisodique, et peut être la conséquence d'observation sur une expérimentation trop courte. De plus, comme précédemment, le taux de détection tardive du mécanisme Pull déployé dans l'expérimentation n'est pas tout à fait nul, contrairement à ce qui avait été montré dans la simulation. Ces différences

TAB. 3.17: Performances du mécanisme Pull mesurées avec l'expérimentation dans le réseau Internet, par rapport à celles obtenues par simulation

	Q_{false} (faux/min) :		R_{late} (%) :	
	Réseau			
TMD (ms)	1	2	1	2
40	0.00/0.00	0.00/0.00	100.00/99.32	94.12/100.00
100	0.00/0.00	1.49/0.00	100.00/84.30	100.00/100.00
240	2.21/0.00	0.00/0.00	100.00/97.52	100.00/100.00
480	0.00/0.00	0.00/0.04	100.00/100	100.00/100.00
960	0.00/0.00	0.00/0.04	7.27/98.85	36.36/24.97
1280	0.14/0.00	0.09/0.03	9.43/0.00	5.56/0.00
2560	0.06/0.00	1.16/0.01	1.75/0.00	0.85/0.00
5120	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
10240	0.00/0.00	0.00/0.00	0.85/0.00	0.00/0.00
30720	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00

	Q_{msg} (message/min) :	
	Réseau	
TMD (ms)	1	2
40	466.67/3898.14	705.60/735.62
100	456.03/3900.22	710.09/735.64
240	423.39/3939.29	577.98/735.70
480	451.51/3997.51	596.70/735.73
960	475.60/3987.88	656.67/735.72
1280	366.80/356.99	264.01/270.75
2560	110.89/109.26	99.92/98.12
5120	56.23/55.69	53.19/52.56
10240	18.30/17.70	18.01/16.22
30720	5.09/4.75	5.06/4.31

TAB. 3.18: Performances du mécanisme APull mesurées avec l'expérimentation dans le réseau Internet, par rapport à celles obtenues par simulation

	Q_{false} (faux/min) :		R_{late} (%) :	
	Réseau			
TMD (ms)	1	2	1	2
40	18.12/0.73	7.84/0.06	88.64/99.22	100.00/100.00
100	17.98/0.73	2.31/0.06	66.04/34.29	100.00/100.00
240	1.90/0.73	1.18/0.05	22.86/0.69	100.00/99.99
480	1.76/0.72	0.00/0.05	13.33/0.00	0.00/5.99
960	0.52/0.09	0.00/0.05	0.00/0.00	0.00/0.04
1280	0.61/0.04	0.00/0.04	0.00/0.00	0.00/0.29
2560	0.00/0.00	0.00/0.02	0.00/0.01	1.85/0.68
5120	0.10/0.00	0.05/0.01	0.00/0.00	0.00/0.00
10240	0.21/0.00	0.00/0.01	0.00/0.00	0.00/0.00
30720	0.05/0.00	0.01/0.00	0.00/0.00	0.00/0.00

	Q_{msg} (message/min) :	
	Réseau	
TMD (ms)	1	2
40	875.53/4436.43	518.58/725.78
100	866.33/4436.49	660.50/725.78
240	865.01/4436.38	657.78/725.79
480	868.99/4434.92	687.92/725.79
960	573.45/514.97	319.62/322.28
1280	225.62/222.50	179.20/174.49
2560	113.08/112.70	98.12/93.48
5120	47.30/48.86	43.41/41.82
10240	17.54/17.37	17.28/15.17
30720	4.89/4.84	4.70/4.27

restent faibles et nous considérons qu'elles ne remettent pas en question la validité de nos simulations.

Les résultats des expérimentations réalisées sur Internet permettent de confirmer la validité de la simulation utilisée pour mesurer les performances des mécanismes de détection d'incident. Nous avons en effet pu vérifier que la modélisation du réseau utilisé dans cette simulation permet de représenter correctement l'influence qu'ont les conditions réelles d'un réseau sur les mécanismes de détection d'incident. De plus, nous avons vu que lorsque ces conditions sont similaires, les performances expérimentales des mécanismes sont similaires à celles obtenues par simulation. Nous pouvons ainsi affirmer que les résultats décrivant les performances des mécanismes de détection d'incident présentés dans la section 3.5 sont valides.

3.7 Conclusion

Dans ce chapitre, nous avons étudié les mécanismes permettant la détection d'un incident affectant une communication entre deux noeuds à l'aide de l'envoi de messages sondes dans le réseau. L'objectif de cette étude était de déterminer comment utiliser au mieux ces mécanismes lorsque leurs utilisateurs ont des besoins spécifiques en fiabilité. Pour cela, nous avons étudié comment les configurer aux mieux, pour ensuite les évaluer, en fonction d'un certain besoin en fiabilité pour une communication exprimé par le TMD, le temps maximum de présence d'un incident non détecté dans le réseau. L'évaluation des mécanismes a consisté à mesurer leurs capacités à détecter un incident en un temps inférieur au temps TMD, ainsi que les conditions dans lesquelles s'opérait cette détection.

Nous avons étudié 3 types de fonctionnement de ces mécanismes, les types Push, Pull et APull. Nous avons tout d'abord étudié l'influence des différents paramètres de configuration sur les performances de ces mécanismes. Une fois ces paramètres déterminés, nous avons pu évaluer les performances des mécanismes en fonction d'un objectif TMD donné et les comparer entre eux. Ces résultats, obtenus par simulation, ont été vérifiés à l'aide d'une implémentation des mécanismes déployée dans des réseaux virtualisés ainsi que sur Internet.

Nous avons observé que le mécanisme de type Push, est le seul à permettre la détection d'un incident en un temps inférieur à quelques centaines de millisecondes. Les autres mécanismes ne peuvent atteindre un temps aussi court, ou alors le nombre de faux positifs générés est trop important pour que le mécanisme soit utilisable. Cependant, à cause de son fonctionnement unidirectionnel, il est parfois nécessaire d'utiliser un mécanisme de type Pull ou APull.

Les mécanismes Pull et APull diffèrent car le mécanisme de type APull permet de détecter en quelques centaines de millisecondes, mais le risque d'apparition de faux positifs est assez important avec ce mécanisme. Le mécanisme Pull ne permet pas une détection en moins d'une seconde, mais les risques de voir apparaitre un faux positif sont très faibles.

Cette étude a ainsi permis de mettre en évidence la possibilité de détecter un incident affectant une communication entre deux noeuds reliés à Internet en un temps donné, qui peut être aussi petit que quelques centaines voir dizaines millisecondes. La consommation de bande passante par le mécanisme de détection d'incident, qui décroît lorsque le TMD objectif augmente, est en effet acceptable pour tous les mécanismes.

Afin de compléter ce travail, il serait intéressant d'étudier les mécanismes de détection d'incident dit « passif » qui n'utilisent pas de messages sondes spécifiques à la détection d'incident, mais qui observent le trafic échangé entre les noeuds du réseau afin de déterminer si un incident affecte une communication. Les résultats présentés dans ce chapitre pourraient servir à ce type de mécanisme de détection puisque les problèmes de faux positifs et de détection tardive, qui ont été étudiés ici, se rencontrent aussi avec ces mécanismes. De plus, puisque les mécanismes de détection passifs né-

cessitent un échange de trafic entre les noeuds pour fonctionner, un mécanisme hybride pourrait être utilisé : on utilise le sondage passif lorsque des communications sont échangées et l'envoi de messages sondes lorsqu'il n'y en a pas. Ceci permettrait probablement la détection des incidents avec une consommation moindre des ressources du réseau.

Dans cette étude, nous avons considéré qu'un incident affectait une communication entre deux noeuds lorsque ceux-ci étaient incapables d'échanger des messages entre eux. Cependant, nous aurions pu élargir notre définition d'incident affectant une communication. Par exemple, un incident pourrait être déclaré lorsque certains besoins de qualité de service, tels que le délai d'acheminement ou la bande passante disponible dans le réseau, ne sont pas satisfaits. Il conviendrait alors d'adapter et d'étudier le comportement des mécanismes pour la détection de tels incidents.

Les mécanismes de détection d'incident sont un composant essentiel des systèmes de rétablissement réseau. Ces travaux nous ont permis d'étudier en détail ce composant, et ainsi nous pourrons proposer dans la suite de ce document un système de rétablissement réseau plus efficace. En effet, dans ce type de système, lorsque des besoins de fiabilité ont été exprimés pour une communication, il est essentiel de prévoir les temps de détection des incidents, ainsi, le temps de rétablissement total d'une communication défaillante sera anticipé.

Chapitre 4

Routage P2P pour la fiabilité des communications

4.1 Introduction

L'amélioration de la fiabilité des communications dans les réseaux est un problème courant. Comme nous l'avons vu, lorsqu'un incident affecte un réseau IP et empêche la délivrance des communications entre les noeuds, des mécanismes de rétablissement sont utilisés pour trouver un chemin alternatif dans le réseau qui ne soit pas affecté par cet incident. Ces mécanismes sont généralement déployés par les opérateurs, mais ne sont pas conçus pour prendre en compte les besoins spécifiques d'une communication.

Lorsque la fiabilité d'une communication entre utilisateurs est un besoin fort, l'utilisation exclusive des systèmes de rétablissement réseau déployés par les opérateurs peut ne pas s'avérer satisfaisante. En effet, ces communications ont des besoins spécifiques qui nécessitent une prise en charge adaptée par le mécanisme de rétablissement. De plus, les mécanismes des opérateurs peuvent défaillir ou être dans l'impossibilité de satisfaire un besoin de l'utilisateur, lorsque le rétablissement consécutif à un incident n'est pas assez rapide par exemple. Ce problème est plus particulièrement marqué pour les communications de bout en bout sur Internet, qui traversent plusieurs réseaux d'opérateurs et qui par conséquent dépendent de plusieurs systèmes de rétablissement, qui ne coopèrent pas entre eux, pour assurer leur délivrance en cas d'incident.

Deux problèmes s'opposent à la possibilité d'amélioration de la fiabilité des communications sensibles entre utilisateurs : la non-coopération des systèmes de rétablissement déployés par les différents opérateurs, évoquée plus haut, ainsi que le temps et les ressources requises pour déployer de nouveaux mécanismes de rétablissement, plus efficaces, qui nécessiteraient le remplacement des routeurs situés au coeur des réseaux. Nous allons donc nous intéresser à une solution qui s'affranchit de ces deux contraintes.

Dans ce chapitre, nous allons présenter un mécanisme de rétablissement réseau, déployé par les utilisateurs et basé sur un système de routage P2P destiné à la fiabiliser les communications sensibles d'un réseau. Ces systèmes, dont certains ont déjà été présentés dans la section 2.5.3, effectuent des opérations de routage sur les noeuds des utilisateurs situés en bout de réseau. Ceci permet par exemple de faire transiter les communications entre deux utilisateurs via un troisième. Par conséquent, si une communication entre deux utilisateurs est affectée par un incident, ce procédé permet son rétablissement si ces utilisateurs sont en mesure de communiquer avec un même utilisateur tiers.

Lorsque la fiabilité des communications échangées par les utilisateurs est importante, il faut que le mécanisme de rétablissement réseau soit en mesure de rétablir une communication affectée par un incident en un temps assez court pour satisfaire les besoins en fiabilité des utilisateurs. Pour cela, notre système tient compte des besoins d'un utilisateur lorsqu'il initie une communication de manière à les satisfaire au mieux. De plus, le système s'assure de ne consommer que les ressources réseau strictement nécessaires pour permettre la satisfaction des besoins exprimés.

Les intérêts de notre système par rapport à ceux existants sont les suivants : nous proposons un système de routage P2P qui prend en compte les besoins de l'utilisateur pour protéger une communication. En effet, en fonction d'un besoin de fiabilité spécifié par l'utilisateur pour une communication donnée, notre système va permettre, lorsque cela est possible, le rétablissement d'une communication lorsque celle-ci est affectée par un incident en un temps assez court pour satisfaire ces besoins. De plus, nous proposons une implémentation de ce système qui permet son déploiement dans tous types de réseau et qui permet la prise en charge de tout type de communication IP. Enfin, notre système introduit quelques concepts novateurs pour le rétablissement rapide d'une communication lorsque la criticité de celle-ci le requiert.

Le reste des sections de ce chapitre sont les suivants : dans la section prochaine, nous présenterons notre système et son fonctionnement. Dans la section 4.3, nous expliquerons comment le système a

été implémenté. La section 4.4 sera consacrée à l'évaluation de notre système. Nous conclurons enfin ce chapitre dans une dernière section.

4.2 Présentation du système

Dans cette section, nous allons décrire le principe de fonctionnement du système. Pour cela, nous allons détailler la façon dont est organisé le réseau overlay, comment sont prises en charge les communications à protéger par le système et comment celui-ci réagit lorsqu'un incident affecte ces communications. De plus, nous justifierons les différents choix de conception effectués.

4.2.1 Contexte d'utilisation

Notre système est dédié à la protection des communications sensibles, qui nécessitent une plus grande fiabilité que les communications normales. Ainsi, le but de notre système n'est pas d'améliorer la fiabilité de toutes les communications d'un réseau, mais de se concentrer sur celles qui ont été désignées par les utilisateurs comme les plus sensibles, et qui méritent que des ressources supplémentaires soient consacrées à leur fiabilité.

Notre système se base sur un réseau overlay de type P2P. Ainsi, il est nécessaire qu'un ensemble de noeuds overlays soit déployé pour former un réseau overlay. Les communications pourront ensuite transiter par chacun de ces noeuds. Les problèmes de sécurité liés à l'éventuelle présence de noeuds malveillants parmi les noeuds du réseau overlay ne seront pas abordés. Ainsi, nous considérons que notre système doit être déployé par des noeuds dits « de confiance », c'est-à-dire ceux dont on est sûr qu'ils ne commettront pas d'actes malveillants à l'encontre des utilisateurs légitimes du système.

Nous considérons enfin que notre système est destiné à être déployé sur un réseau composé d'un nombre limité de noeuds participants. En effet, comme expliqué au paragraphe précédent, les noeuds participants sont de confiance et par conséquent il est difficilement envisageable que leur nombre soit très important. De plus, nous montrerons dans le chapitre 5 qu'au-delà d'une certaine limite, l'utilisation d'un plus grand nombre de noeuds n'apporte pas d'amélioration au fonctionnement du système. Par exemple, 100 noeuds participants sont considérés comme un très grand nombre pour notre système. Par conséquent, les problèmes de passage à l'échelle de notre système lors de l'utilisation d'un très grand nombre de noeuds ne seront qui brièvement abordés ici et ne sont pas pris en considération lors de sa conception.

4.2.2 Objectifs du système

L'objectif de notre système est l'amélioration de la fiabilité des communications des utilisateurs, à leur demande et en fonction de leurs besoins. Lorsqu'un incident affecte la bonne délivrance d'une communication, le but de notre système est de tout mettre en oeuvre pour rétablir l'acheminement de cette communication jusqu'à son destinataire. De plus, la rapidité avec laquelle est rétablie cette communication sera en fonction des besoins préalablement exprimés par les utilisateurs qui participent à cette communication. Notre système sera conçu pour être le plus économe en ressources réseau utilisées. Cependant, les ressources utilisées seront fonction des besoins de l'utilisateur pour la fiabilité d'une communication. Elles pourront être importantes si les besoins des utilisateurs le sont. Enfin, le déploiement de notre système doit être le plus aisé possible. Il doit pouvoir fonctionner pour tout type de communication IP et ne pas nécessiter d'intervention autre dans le réseau que celle de ces utilisateurs.

4.2.3 Construction du réseau overlay

Nous allons tout d'abord expliquer comment est organisé le réseau overlay, c'est-à-dire l'organisation du réseau virtuel formé par les noeuds participants au système.

Noeuds participants

Dans notre système, les noeuds participants au réseau sont un ensemble de noeuds qui collaborent afin de permettre une plus grande fiabilité des communications qui transitent par eux. Pour cela, chaque noeud rejoint le système grâce à une application exécutée par leur système d'exploitation.

Dans notre système, la communication à protéger doit être établie entre un des noeuds du réseau overlay et un noeud quelconque du réseau. En effet, il n'est pas nécessaire que le noeud « destination » de la communication à protéger soit membre du réseau overlay. Pour cela, des techniques particulières, reposant par exemple sur la traduction d'adresse réseau (NAT pour Network Address Translation), sont utilisées (voir la description de NATRON à la section 2.5.3).

Dans l'état actuel de nos travaux, le noeud établissant une communication à protéger doit nécessairement faire partie du réseau overlay. Cette restriction nous a permis de simplifier les opérations de mise en place de la protection des communications. Cependant, il est tout à fait envisageable de mettre en place des mécanismes qui permettraient à un noeud extérieur au réseau overlay de faire transiter une communication à protéger par celui-ci pour améliorer sa fiabilité. Les noeuds participants au réseau overlay seraient semblables à des « proxys » par qui les noeuds extérieurs au système font transiter les communications à protéger. Pour permettre cela, un protocole de communication spécifique entre les noeuds extérieurs et les noeuds du système devrait être utilisé afin, par exemple, de permettre au noeud extérieur de spécifier au noeud du système ses besoins en fiabilité.

Topologie du réseau overlay

La topologie du réseau overlay de notre système est de type Full Mesh. En effet, chacun des noeuds du système est capable d'acheminer une communication à chacun des autres noeuds directement, c'est-à-dire sans qu'il soit nécessaire que celle-ci transite par un autre noeud du système.

À la différence d'un système tel que RON, l'utilisation d'un réseau Full Mesh n'entraîne pas une consommation de ressource élevée, qui empêche le passage à l'échelle du système. En effet, dans le système RON chaque lien du réseau overlay est périodiquement mesuré par l'envoi de messages sondes. Ce n'est pas le cas de notre système : les liens overlays ne sont pas systématiquement mesurés. L'envoi de messages sondes pour la mesure des liens n'est effectué que si ce lien est utilisé par une communication à protéger. Ainsi, la consommation de ressources réseau dans RON augmentait avec le carré du nombre de noeuds présents dans le système, alors que dans notre cas, nous le verrons plus bas, cette consommation augmente linéairement avec le nombre de communications à protéger par le système.

L'utilisation de la topologie Full Mesh nécessite que chaque noeud du système ait connaissance de tous les autres noeuds. Ceci peut poser un problème si le nombre de noeuds dans le système est très grand. Nous laisserons ce problème de côté, car comme nous l'avons dit plus haut, notre système n'est pas destiné à être utilisé par des réseaux composés d'un très grand nombre de noeuds.

Pour rejoindre le système, un noeud contacte un noeud quelconque faisant déjà partie du réseau overlay pour lui demander la liste des noeuds participant au système. Une fois que cette liste lui est retournée, le noeud arrivant va contacter individuellement chacun des noeuds de cette liste de manière à se faire connaitre de ceux-ci. Les noeuds contactés répondent au noeud arrivant que sa présence a

bien été prise en compte. À l'issu de ce processus, le noeud arrivant fait pleinement partie du réseau overlay.

4.2.4 Mise en place d'une communication

Nous allons voir dans cette partie quels sont les mécanismes mis en place lorsqu'une communication à protéger débute. Dans notre système, lorsqu'aucun incident n'est présent, une communication est acheminée normalement, par la route IP « classique ». Cependant, nous verrons qu'en fonction d'un besoin en fiabilité demandé par l'utilisateur et son application, des mécanismes de détection d'incident et des chemins de secours potentiels sont mis en place.

Besoin de fiabilité demandé par l'utilisateur

Afin de mettre en place les mécanismes de protection d'une communication adaptés aux besoins de l'utilisateur en terme de fiabilité, il est nécessaire d'exprimer ce besoin. Divers paramètres influent sur celui-ci, tels que l'importance qu'attache l'utilisateur à la bonne délivrance du service, mais aussi la sensibilité de l'application à une interruption de la livraison des communications, par exemple.

Afin d'exprimer le besoin en fiabilité, nous allons introduire le Temps Maximum d'Interruption Toléré (TMIT). Ce temps est la durée maximum tolérée durant laquelle un noeud participant à une communication peut ne pas recevoir le trafic émis par l'autre noeud. C'est en fait le temps maximum toléré pour opérer le cycle de rétablissement, décrit dans la section 2.4.1.

En fonction des besoins de l'application et de fiabilité pour une communication, le TMIT est déterminé par l'utilisateur et transmit à notre système. Le système met alors en place les mécanismes permettant de rétablir une communication si celle-ci était affectée par un incident qui entrainerait une interruption de la livraison de la communication durant un temps supérieur au TMIT.

Détection d'incident sur le chemin principal

En l'absence d'incident, les communications protégées par notre système continuent à être acheminées par la route « classique », calculée par les protocoles de routage IP. Il est cependant nécessaire de vérifier périodiquement si cette route est affectée par un incident afin de déclencher le mécanisme de rétablissement qui permettra de maintenir une communication.

Le temps avec lequel il est nécessaire de détecter un incident dépend du TMIT. En effet, plus le TMIT est faible et plus il est nécessaire de détecter un incident rapidement. Nous allons exprimer le Temps Maximum de Détection (TMD) qui est le temps maximum pour détecter un incident pour permettre de satisfaire le TMIT. On a :

$$TMD = TMIT - \alpha$$

La détection d'incident n'est que la première partie du cycle de rétablissement et le TMD est nécessairement inférieur au TMIT. Nous expliciterons la valeur de α plus loin dans la section 4.2.6. Il faut noter que le TMD utilisé dans cette partie est en tout point semblable au temps maximum de présence d'un incident non détecté dans le réseau utilisé dans la partie précédente.

Notre système n'impose pas l'utilisation d'un mécanisme de rétablissement particulier. Il permet en effet d'utiliser un mécanisme de détection totalement externe au système, qui peut, par exemple, utiliser l'envoi de messages sondes, ou encore la notification matérielle. Celui-ci doit cependant pouvoir assurer au mieux la détection d'un incident en temps inférieur au TMD qui lui est demandé. De

plus, il peut optionnellement être muni d'un état « Attention » lorsque la présence d'un incident est suspectée, mais pas encore confirmée. Dans ce cas, le mécanisme de rétablissement alerte notre système de la probable apparition d'un incident afin que ce dernier anticipe son rétablissement au plus tôt.

Dans l'état actuel, notre implémentation utilise un mécanisme de détection d'incident de type Pull lorsque le TMD est supérieur à 1 seconde et de type APull sinon. Ces mécanismes ont été décrits dans la partie précédente. L'état « Attention » pour ces mécanismes est déclenché dès qu'un message est perdu.

Chemins de secours potentiels

Afin de contourner un incident affectant le chemin principal utilisé pour acheminer une communication, le système fait emprunter un chemin de secours à la communication pour l'acheminer jusqu'à sa destination. Lors de l'établissement d'une communication à protéger, il est nécessaire de considérer quels sont les chemins qui seront éventuellement utilisés en cas d'apparition d'un incident sur le chemin principal avant l'apparition de cet incident. Nous considérons en effet que cette approche proactive est plus adaptée à un rétablissement rapide des communications.

Différents travaux (voir les sections 2.5.3, 5.2.3 et 5.4.3) sur la portée des mécanismes de routage P2P ont montré que pour contourner un incident affectant les communications dans Internet, il suffisait dans la plupart des cas de faire transiter ces communications par un seul noeud overlay tiers. Ainsi, dans notre système, nous appuierons sur ce fait pour déterminer les chemins de secours à utiliser en cas d'incident affectant une communication.

Lorsqu'une communication à protéger est établie par un noeud A vers un noeud B, notre système va selectionner k chemins $c_1, c_2, \ldots, c_k$, ayant pour extrémité les noeuds A et B, et transitant par un noeud membre du réseau overlay, distincts de A et B, comme étant les chemins de secours potentiels. Nous appellerons les noeuds overlays par qui transitent ces chemins les noeuds de transit.

L'algorithme du choix des k chemins de secours potentiels n'est pas imposé dans notre système. Idéalement, lorsqu'un incident affecte un chemin entre les noeuds A et B, l'algorithme du choix des noeuds doit maximiser la probabilité de proposer des chemins de secours non affectés par une panne. Pour cela, l'idée la plus couramment admise, développée dans la section 2.5.2, est que l'algorithme doit se baser sur la topologie du réseau sous-jacent au réseau overlay de manière à proposer des chemins de secours qui soient le plus topologiquement distincts du chemin principal utilisé pour les communications entre A et B (c'est-à-dire que ces chemins doivent avoir le moins d'équipements réseaux en commun possible). L'état actuel des connaissances ne permet malheureusement pas de proposer une implémentation satisfaisante de cette solution.

Dans l'état actuel, notre implémentation ne dispose que d'un algorithme qui sélectionne k chemins de secours potentiels en sélectionnant aléatoirement pour chacun un noeud de transit parmi les noeuds overlays. Cependant, l'utilisation d'un algorithme qui sélectionne k chemins de secours potentiels utilisant chacun un noeud de transit choisi parmi ceux dont l'adresse IP est la plus distincte des adresses de A et B est envisagée.

Cet algorithme est exécuté par le noeud utilisant le système de routage P2P dont est issue la communication à protéger. Une fois les chemins de secours potentiels déterminés, il n'est pas nécessaire de signaler aux noeuds de transit traversés par ces chemins qu'ils ont été « choisis ».

Détection d'incident sur les chemins de secours potentiels

Tout au long du déroulement de la communication à protéger, il est nécessaire de vérifier qu'aucun incident n'affecte les chemins de secours potentiels. Pour cela, un mécanisme de détection d'incident est mis en place pour chacun de ces chemins de secours, d'une façon similaire à ce qui est fait pour le chemin principal. Cependant, le temps maximum pour détecter un incident affectant un chemin de secours potentiel peut être plus long que celui du chemin principal. Nous considérons que le temps TMD du mécanisme de détection d'un chemin de secours potentiel doit être égal à β fois celui du chemin principal.

Notre système n'impose pas l'utilisation d'un mécanisme de détection d'incident particulier pour les chemins alternatifs potentiels. De plus, ce mécanisme peut être distinct de celui utilisé pour surveiller le chemin principal. De plus, en cas d'utilisation d'un mécanisme de détection d'incident par envoi de messages sondes, c'est le mécanisme de routage P2P qui est utilisé pour acheminer ces messages. Par conséquent, il n'est pas nécessaire d'adapter spécifiquement le mécanisme pour permettre son utilisation pour détecter les incidents affectant les chemins de secours potentiels.

Si un chemin de secours potentiel est affecté par un incident, il convient de retirer ce chemin de la liste des k chemins de secours potentiels et d'en sélectionner un nouveau parmi les chemins overlays possibles.

4.2.5 Apparition d'un incident

Nous allons voir dans cette section les mécanismes mis en place lorsqu'un incident affectant le chemin principal est détecté : nous verrons quels sont les différents mécanismes mis en place lors de la détection d'un incident et comment sont acheminées les communications par le chemin de secours.

Vérification et sélection du chemin de secours

Lorsqu'un incident est signalé sur le chemin principal, il convient de vérifier immédiatement si les chemins de secours potentiels sont affectés par cet incident. En effet, bien que la présence d'incidents sur ces chemins soit contrôlée périodiquement, la fréquence de ces contrôles est plus faible que pour le chemin principal et il est possible qu'un incident qui affecte le chemin principal et un ou plusieurs chemins de secours potentiels ne soit pas encore détecté par les mécanismes de détection des chemins de secours au moment où il est signalé par le mécanisme du chemin principal.

Pour cela, notre système demande à chacun des mécanismes de détection d'incident déployés sur les chemins de secours potentiels d'effectuer une vérification immédiate de la présence d'un incident sur leur chemin. Ces mécanismes informent notre système du résultat de cette vérification au plus tôt. Par exemple, si le mécanisme de détection d'incident utilisé pour l'envoi de messages sondes, notre système va déclencher l'envoi immédiat de messages sondes vers tous les chemins de secours potentiels. Dès qu'un message sonde a parcouru le chemin à vérifier, le mécanisme de détection d'incident informe notre système que le chemin n'est pas affecté par un incident et peut être utilisé.

Le choix du chemin de secours à utiliser pour acheminer les communications peut ensuite être effectué parmi les chemins de secours déclarés comme valides par leurs mécanismes de rétablissement. Bien que plusieurs solutions puissent être envisagées, nous nous proposons de choisir le chemin dont le mécanisme de détection d'incident aura déclaré la validité en premier. Les communications entre le noeud source et le noeud destination sont alors acheminées par ce chemin et plus par le chemin IP.

Dans le cas où l'ensemble des k chemins de secours potentiels est affecté par un incident, la recherche d'un chemin de secours valide doit être étendue à d'autres noeuds parmi les membres du réseau overlay. De plus, l'utilisation de chemin transitant par plus d'un noeud overlay pour joindre la

destination peut être envisagée. Dans l'état actuel de notre implémentation, lorsque cela se produit, l'ensemble des noeuds présents dans le réseau overlay restant est sélectionné comme chemins de secours potentiels et interrogé immédiatement. Comme précédemment, nous proposons de sélectionner le chemin de secours à emprunter dont le mécanisme de détection d'incident aura signalé sa validité en premier.

Acheminement des communications sur le chemin de secours

Une fois le chemin de secours vérifié et sélectionné, il est possible de rétablir l'acheminement de la communication jusqu'à sa destination. Ceci est effectué par un routage par la source : la succession des noeuds à emprunter est indiquée dans chacun des paquets de la communication. Ainsi, le noeud dont est issue la communication ajoute aux paquets de celle-ci les adresses des différents noeuds à traverser pour emprunter le chemin de secours sélectionné. Les paquets sont ensuite transférés aux différents noeuds de cette liste jusqu'à la destination.

Il est nécessaire de mettre en place un mécanisme spécifique pour permettre l'acheminement des communications de réponse, qui transitent dans le sens -noeud destination- vers -noeud origine- de la communication. En effet, lorsque les communications sont acheminées sur le chemin overlay de secours, les différents noeuds overlays de ce chemin n'ont pas connaissance de la communication à protéger : ils ne font que relayer les communications jusqu'à la destination.

Puisque le système est invisible pour les applications, le noeud destination va répondre au dernier noeud IP lui ayant transmis la communication, c'est-à-dire le dernier noeud du chemin overlay utilisé pour acheminer la communication (il faut noter que ce noeud et le noeud destination sont les mêmes si le noeud destination est un noeud participant au système de routage P2P mais cela ne change pas la problématique exposée ici). Il faut par conséquent que celui-ci sache si le chemin actuellement utilisé pour cette communication est un chemin IP ou un chemin overlay, et dans ce cas, de quelle succession de noeuds est composé ce chemin overlay.

Pour cela, lorsqu'un noeud overlay constate qu'une communication acheminée sur un chemin overlay dont il est le dernier noeud transite par lui, il mémorise cette communication et le chemin overlay utilisé pour son acheminement. Lorsque la réponse à cette communication parvient au noeud, celui-ci sait ainsi par quel chemin l'acheminer pour qu'elle atteigne la communication. La mémorisation d'une communication et de son chemin overlay associé par un noeud cesse quand il constate que ce chemin est inutilisé depuis une certaine durée ou que celui-ci reçoit un message spécifique, envoyé par le noeud ayant établi la communication à protéger, indiquant que le chemin overlay ne sera plus utilisé.

Détection d'incident sur le chemin de secours

Lorsque le chemin de secours est utilisé pour acheminer la communication, il est nécessaire de mettre en place le mécanisme de détection d'incident approprié à ses besoins de fiabilité. Pour cela, un mécanisme de détection d'incident dont le temps maximum de détection TMD est choisi en fonction du temps $TMIT$ spécifié par l'utilisateur au début de la communication doit être mis en place sur ce chemin. Bien sûr, le mécanisme de détection d'incident utilisé lorsque le chemin n'était qu'un chemin de secours potentiel, non utilisé pour acheminer la communication, doit être supprimé.

L'état « Attention » du mécanisme de détection d'incident

Comme nous l'avons dit, le mécanisme de détection d'incident utilisé sur le chemin principal peut se placer dans un état « Attention » lorsque la probabilité de présence d'un incident est forte, mais

qu'elle n'est pas confirmée. Ceci permet à notre système de déclencher plus tôt les différentes étapes préalables au rétablissement réseau décrites plus haut.

Voici les actions déclenchées dès que le mécanisme de détection d'incident signale un état « Attention » :

- Vérification des chemins de secours potentiels et sélection du chemin de secours
- Début de l'acheminement des communications sur le chemin de secours, tout en maintenant l'acheminement sur le chemin principal IP. Dans ce cas, les communications sont diffusées deux fois dans le réseau. Ce mécanisme sera appelé par la suite double acheminement
- Mise en place du mécanisme de détection d'incident sur le chemin alternatif en fonction de TMIT

Si le mécanisme de détection confirme par la suite l'incident, il suffira au système de stopper l'acheminement sur le chemin IP pour compléter le processus de rétablissement. Si ce n'est pas le cas, le chemin de secours sélectionné devra être à nouveau considéré comme un chemin de secours potentiel. Le mécanisme de détection d'incident est réinitialisé et les communications n'y sont plus acheminées.

Mécanisme pour la réversion

Une fois le chemin de secours déterminé et utilisé pour acheminer à nouveau les communications jusqu'à leur destination, il est nécessaire de mettre en place les mécanismes qui permettront la réversion, c'est-à-dire de rétablir l'acheminement sur l'ancien chemin principal (la route Internet calculée par le routage IP), lorsque l'incident qui l'affecte sera terminé.

Pour cela, l'ancien chemin principal doit être considéré comme un chemin de secours potentiel et un nouveau mécanisme de détection d'incident y est déployé. Comme pour les chemins de secours potentiels, ce mécanisme a pour temps maximum de détection une valeur égale à β fois celle du nouveau chemin principal.

Si le mécanisme de détection reporte que l'ancien chemin principal n'est plus affecté par un incident, il est alors possible d'opérer une réversion, c'est-à-dire de rétablir l'acheminement des communications sur le chemin IP. Ceci peut avoir pour avantage de meilleures performances pour l'acheminement, si par exemple le chemin IP dispose de plus de débit que le chemin overlay. Cependant, enclencher systématiquement la réversion pourrait aussi avoir des inconvénients, en particulier en cas de panne transitoire affectant le chemin IP. Dans ce cas, l'acheminement des communications alternerait fréquemment entre le chemin IP et le chemin overlay de secours, ce qui nuirait à la qualité des transmissions : par exemple, on pourrait observer de fréquentes coupures dans la livraison du trafic à la destination ou encore un désequencement des paquets. Par conséquent, nous avons décidé de ne pas imposer la réversion dans notre système, mais d'en laisser le choix à l'utilisateur.

4.2.6 Choix des valeurs pour les constantes utilisées

La constante α

Nous l'avons dit, la constante α intervient dans la relation

$$TMD = TMIT - \alpha$$

qui détermine le temps maximum de détection d'un incident (TMD) en fonction du temps maximum d'interruption toléré d'une communication (TMIT). α doit être choisie de manière à satisfaire le TMIT lors du rétablissement d'une communication par notre système.

Pour cela, il faut s'intéresser au cycle de rétablissement, présenté dans la section 2.4.1. Nous constatons que pour que le TMIT soit respecté, il est nécessaire que le temps de rétablissement, c'est-à-dire le temps nécessaire au cycle de rétablissement complet, soit inférieur au TMIT. Nous constatons qu'une fois un incident détecté, les étapes « de temporisation », de « notification », de « l'opération de rétablissement » et de « réacheminement du trafic » sont encore nécessaires pour terminer le cycle de rétablissement.

Dans notre système, il n'y a pas d'étape de temporisation, puisque nous donnons la priorité au rétablissement rapide des communications, ni de notification, puisque le routage par la source n'impose pas de reconfiguration des noeuds overlay pour permettre rétablissement. Par contre, notre système a une étape d'opération de rétablissement, qui correspond à l'étape « Vérification et sélection du chemin de secours » décrite dans la section 2.4.1, ainsi qu'une opération de réacheminement, qui correspond au temps d'acheminement des données jusqu'à la destination de la communication une fois le chemin de secours à utiliser sélectionné.

Ainsi, le temps du cycle de rétablissement, dans notre système, est égal à la somme du temps de détection d'un incident, du temps de sélection du chemin de secours, et du temps de réacheminement. Le temps de sélection du chemin de secours est égal au temps pris par le message qui a été le plus rapide à parcourir les chemins de secours potentiels lors de leur vérification. Ce temps est ainsi égal au min(Temps d'acheminement aller-retour des chemins de secours). Le temps de réacheminement du trafic sera différent selon que l'on considère le temps de réacheminement de la source de la communication vers la destination ou l'inverse, mais dans les deux cas, il ne sera pas supérieur au temps d'acheminement aller-retour des communications sur le chemin de secours sélectionné.

Puisque le temps d'acheminement varie constamment au cours du temps, il n'est pas possible de prévoir les temps d'acheminement évoqués plus haut, en particulier au moment de l'établissement de la communication, lorsqu'il faut déterminer la valeur TMD du mécanisme de détection d'incident. Ainsi, nous nous proposons de donner une approximation du temps d'acheminement sur le chemin de secours par celle mesurée sur le chemin principal.

Ainsi, nous utilisons une approximation du temps nécessaire au rétablissement complet des communications comme étant la somme du temps de détection de l'incident ajoutée à deux fois le temps d'acheminement aller-retour des communications sur le chemin principal. Bien que le temps d'acheminement sur le chemin de secours soit généralement supérieur à celui du chemin principal, nous pensons que cette différence sera compensée par :

- Le choix d'un temps aller-retour pour le réacheminement des communications alors que ce temps est correspond normalement à un acheminement unidirectionnel
- Le fait que le TMD est un temps maximum et que le temps effectivement nécessaire à détecter d'un incident puisse être plus court.
- Si l'état « Attention » du mécanisme de détection d'incident est utilisé, les étapes de sélection du chemin de secours et d'acheminement des communications sur ce chemin sont anticipées.

Ainsi, le temps la constante α est exprimée par :

$$\alpha = 2.\text{Temps d'acheminement des communications aller-retour mesuré sur le chemin IP entre la source et la destination de la communication à protéger}$$

Par conséquent, le temps TMD est calculé ainsi :

$TMD = TMIT - 2$.Temps d'acheminement des communications aller-retour
mesuré sur le chemin IP entre la source et la destination
de la communication à protéger

La constante k

La constante k est le nombre de chemins de secours potentiels à sélectionner. Lorsqu'un incident se déclare sur le chemin principal, c'est un de ces chemins qui sera utilisé comme chemin de secours. Le choix de k est important, car il est nécessaire de sélectionner un nombre de chemins de secours potentiels suffisamment grand pour limiter le risque qu'ils soient tous affectés par un incident lorsqu'il se déclare, tout en ne choisissant pas une valeur trop grande qui entrainerait une consommation de ressource importante due à l'utilisation périodique de mécanismes de détection d'incident sur ces chemins.

Lorsque l'on considère les communications entre des noeuds situés en bout de réseau, on a vu dans la section 2.3.1 que si un incident affecte les communications d'un noeud vers un autre, il n'est pas rare que les communications vers plusieurs autres noeuds soient aussi affectées. Ainsi, pour que notre système soit capable de trouver un chemin de secours alternatif qui n'est pas concerné par un incident qui affecte le chemin principal, il est important d'envisager l'utilisation d'un nombre suffisamment grand de chemins de secours potentiels. Le choix de ce nombre dépend aussi de l'algorithme utilisé pour choisir ces chemins. Plus l'algorithme est efficace et plus il choisira des chemins de secours potentiels peu susceptibles d'être affectés par l'incident et plus la constante k pourra être choisie petite.

Nous proposons d'utiliser la valeur $k = 8$ pour le nombre de chemins de secours potentiels. Nous étudierons plus en détail et justifierons ce choix dans le chapitre 5 consacré à la mesure de la portée des mécanismes de routage P2P.

La constante β

La constante β représente le facteur du temps maximum de détection (TMD) d'un incident demandée pour les chemins de secours potentiels, par rapport à celui du chemin principal. Nous considérons que le TMD pour les chemins de secours potentiel n'a pas à être trop faible, car la détection d'incident sur ces chemins est de moindre importance puisqu'ils ne sont pas utilisés pour acheminer les communications et que lorsqu'un incident se déclare sur le chemin principal, les chemins de secours sont vérifiés avant d'être utilisés.

Nous proposons d'utiliser la valeur $\beta = k$. Ainsi, les ressources consommées par un mécanisme de détection d'incident sur un chemin de secours potentiel seront plus faibles si k est grand, ce qui compensera le nombre plus élevé de mécanismes de détection d'incident déployé.

4.3 Implémentation

Dans cette section, nous allons décrire l'implémentation de notre logiciel de routage P2P pour la fiabilité des communications.

4.3.1 Les messages utilisés

Différents messages sont utilisés dans notre système pour son fonctionnement. Ceux-ci servent à la gestion du réseau overlay, à l'acheminement des communications ou à la détection d'incident. Tous les messages utilisés par notre système utilisent le protocole de transport UDP. Les ports utilisés par UDP peuvent être choisis par l'utilisateur.

Les messages de type OVERLAY

Le rôle des messages de type OVERLAY est de gérer le réseau overlay, c'est-à-dire de permettre l'arrivée et le départ des noeuds de ce réseau. Il existe 7 types de message.

OVERLAY JOIN

Ce message est envoyé par un noeud overlay à un autre lorsque le premier désire établir une relation avec le second, c'est-à-dire la mise en place d'un lien overlay pouvant être utilisé pour acheminer une communication entre les deux noeuds.

Lors de la réception de ce message, un noeud retourne à l'émetteur un message JOIN-ACK.

OVERLAY JOIN-ACK

Ce message est envoyé en réponse à un message OVERLAY JOIN et indique que le noeud qui émet ce message est prêt à établir un lien overlay avec le noeud destinataire.

Une fois ce message reçu par son destinataire, qui avait précédemment envoyé un message JOIN, celui-ci considère le lien overlay avec le noeud émetteur comme établi et lui envoie un message JOIN-ACKACK

OVERLAY JOIN-ACKACK

Ce message est envoyé pour confirmer la réception du message OVERLAY JOIN-ACK au noeud l'ayant émis.

Une fois ce message reçu par son destinataire, celui-ci considère le lien overlay avec le noeud émetteur comme établi.

OVERLAY LEAVE

Ce message est envoyé par un noeud overlay pour signaler à un autre qu'il va quitter le réseau overlay. Par conséquent, lors de la réception de ce message par le noeud destinataire, le lien overlay qui le relie au noeud émetteur est supprimé et celui-ci envoie un message LEAVE-ACK au noeud émetteur afin de confirmer la bonne prise en compte de ce message. Dans le cas contraire, le noeud émetteur peut réémettre le message OVERLAY LEAVE.

OVERLAY LEAVE-ACK

Ce message est envoyé pour confirmer la réception d'un message LEAVE au noeud l'ayant émis. Une fois le message reçu, le noeud overlay considère que le lien overlay le reliant au noeud émetteur est correctement supprimé.

LIST-REQ

Ce message est envoyé par un noeud à un autre pour lui demander l'envoi de la liste des noeuds membres du réseau overlay. Lors de la réception de ce message, un noeud retourne à l'émetteur un message LIST.

LIST

Ce message contient une liste d'adresses de noeuds membres du réseau overlay. Une fois ce message envoyé par un noeud overlay à un autre, il sera possible pour le noeud destinataire de demander à chacun des noeuds de la liste d'établir un lien overlay avec lui à l'aide du message JOIN.

Les messages de type DATA

Les messages de type DATA servent à l'acheminement des communications à protéger. Il en existe 2 types.

DATA

Le rôle de ce message est d'acheminer les communications à protéger lorsque celles-ci empruntent un chemin overlay à la place du chemin IP traditionnel. Pour cela, le message de type DATA utilise un entête qui décrit la communication à protéger ainsi que le chemin overlay par laquelle la faire transiter. Ensuite, le message de type DATA contient le paquet IP non modifié des communications à protéger.

L'entête d'un message DATA contient notamment la description de la communication protégée (actuellement les adresses des noeuds qui participent à cette communication) et le nombre et les adresses des noeuds constituant le chemin overlay par lequel acheminer la communication.

Lors de la réception d'un message DATA, un noeud observe sa position dans la liste des adresses constituant le chemin overlay. S'il existe un noeud qui lui succède dans le chemin, le noeud lui transmet le message sans le modifier. Par contre, si le noeud est le dernier noeud du chemin (ou noeud de sortie), celui-ci « décapsule » le message afin de retrouver le paquet IP original. Ce paquet est ensuite « réinjecté » dans le réseau IP et sera acheminé de manière transparente par ce dernier jusqu'à sa destination.

De plus, comme il l'a été évoqué dans la section 4.2.5, afin de permettre l'acheminement par le système de routage P2P des communications dans le sens du retour, un noeud de sortie doit mémoriser le chemin à utiliser par cette communication. Pour cela, le noeud maintient une table qui à une description de communication associe un chemin overlay. Le noeud réalise ensuite l'interception du trafic qui correspond à la description de la communication (voir la section 4.3.2, plus bas), consulte cette table, et crée un nouveau message de type DATA pour acheminer les données par le chemin overlay approprié.

UNHANDLE

Le rôle du message DATA UNHANDLE est d'indiquer explicitement à un noeud de stopper la prise en charge d'une certaine communication. En effet, il est nécessaire de signaler aux noeuds de sortie overlays qui mémorisent les chemins overlays à utiliser pour la protection des communications qu'ils peuvent stopper l'interception du trafic et ne plus mémoriser le chemin overlay à emprunter.

Le message UNHANDLE contient la description de la communication concernée et est envoyé par le noeud qui a mis en place la protection de la communication lorsque l'utilisateur décide d'arrêter la protection de celle-ci ou lorsqu'après un rétablissement de la communication par le système overlay, le chemin IP est à nouveau disponible et que l'on décide de le réutiliser pour acheminer la communication.

Les messages de type PROBE

Les messages de type PROBE servent à implémenter une solution de détection d'incidents par envoi de messages sondes au sein du système de routage P2P. Ces messages contiennent notamment la description de la communication concernée par la détection d'incident et le nombre et les adresses des noeuds constituant le chemin overlay sur lequel l'incident est recherché. Ainsi, ces messages sont acheminés par les noeuds overlays de la même façon que pour les messages de type DATA.

Il faut noter que les messages de type PROBE ne peuvent être utilisés pour détecter des incidents entre des noeuds overlays uniquement. Dans le cas où la destination d'une communication à protéger ne serait pas membre du réseau overlay, l'utilisation de message « compréhensibles » par tous les noeuds IP, tels que les messages ICMP ECHO REQUEST, est prévue.

4.3.2 Interception et réinjection transparente du trafic

Afin de permettre la prise en charge des communications par notre système de manière transparente (c'est-à-dire sans besoin de modifications) pour les applications, il est nécessaire d'intercepter le trafic, c'est-à-dire les paquets IP, issus de ces applications. L'interception du trafic consiste à rediriger les paquets IP des communications à prendre en charge vers notre application avant que ceux-ci ne soient traités par le système d'exploitation. Pour cela, nous avons utilisé la cible « QUEUE » du pare-feu Netfilter utilisé dans le système d'exploitation Linux[88].

Ainsi, lorsque l'utilisateur demande la protection d'une communication réalisée entre deux noeuds, une règle Netfilter est mise en place et indique que les communications entre ces noeuds doivent être dirigées vers la cible « QUEUE ». Notre application peut alors récupérer les paquets IP interceptés et peut décider, en fonction des mesures sur l'état du réseau, si le paquet doit être acheminé par la route IP classique ou par un chemin constitué de noeuds overlay. Notons que le même procédé est opéré par les noeuds overlays de sortie pour la prise en charge du trafic dans le sens du retour.

Lorsqu'un paquet IP a été acheminé par notre système via un chemin overlay, il est ensuite nécessaire de le réintroduire dans le réseau IP tel qu'il était lorsque l'application l'a émis. Il est ainsi nécessaire que ce paquet soit réémis sans que le système d'exploitation sur lequel est déployé notre logiciel ne le modifie. Pour cela, nous utilisons un socket réseau de type « Raw Socket » [48].

4.3.3 Architecture du logiciel

La figure 4.1 représente l'architecture de notre logiciel. Nous allons expliquer quel est le rôle de ses différents composants.

- OSocketOverlay : Ce composant prend en charge la réception et l'émission des messages propres au système overlay. Il fait remonter les messages overlays au OManager qui va les interpréter et il reçoit des ordres de ce dernier pour l'envoi de messages overlay.
- OSocketIP : Ce composant prend en charge l'interception et la réinjection des paquets IP correspondant aux communications à protéger. Lors de l'interception d'un paquet IP, il interroge l'OComProtector qui correspond à la communication à protégé liée à ce paquet pour déterminer

s'il doit être acheminé par le réseau IP, auquel cas il est immédiatement réinjecté dans le réseau IP, ou bien s'il doit être acheminé via un chemin overlay, et dans ce cas un message overlay de type DATA contenant le paquet IP est crée et est envoyé par l'intermédiaire du OSocketOverlay. Ce composant reçoit aussi des ordres du OManager pour la réinjection de paquet dans le réseau IP.

- OManager : Ce composant est le coeur de notre système. Il coordonne les actions entre les différents composants du système. Par exemple, il a pour rôle d'interpréter les messages overlays et ainsi de maintenir à jour le composant OTopology, de prévenir les objets OIncidentDetector lors de l'arrivée de messages de réponse, de créer un objet OComProtector dédié à la protection d'une communication à la demande de l'utilisateur.
- OComProtector : Ce composant prend en charge la gestion d'une communication à protéger. Un objet du type de ce composant est créé pour chacune des communications à protéger par le noeud dans notre système. Le rôle de ce composant est de déterminer et maintenir à jour le chemin principal et la liste des chemins de secours potentiels pour l'acheminement d'une communication. Pour cela, ce composant utilise un algorithme de choix de chemin implémenté par un objet de type OPathSelector. De plus, ce composant a la charge de la mise en place des mécanismes de détection d'incident de type OIncidentDetecor sur ces différents chemins qui l'informent de l'état des chemins qu'ils surveillent.
- OPathSelector : Ce composant a pour but d'implémenter un algorithme de choix des chemins de secours potentiels pour une topologie et une communication à protégée donnée. Différentes implémentations peuvent être utilisées. Parmi celles disponibles, citons OPathSelectorRandom, qui choisit un chemin de secours transitant par un noeud overlay choisi aléatoirement parmi l'ensemble des noeuds overlays.
- OIncidentDetector : Ce composant a pour but d'implémenter un mécanisme de détection d'incident affectant un chemin donné. La rapidité de la détection d'incident est précisée par l'objet OComProtector lors de la création de ce composant. Ce composant l'informe en permanence de l'état du chemin observé.
- OUserInterface : Ce composant fait le lien entre le composant OManager et l'utilisateur du système. Pour cela, il propose une interface à l'utilisateur qui lui permet par exemple de démarrer ou d'arrêter la protection d'une communication ou de recevoir des informations sur l'état du système.

4.3.4 Utilisation

Notre logiciel est implémenté par le langage Java qui est utilisable sur tout type de système d'exploitation. Cependant, l'utilisation du pare-feu Netfilter et des sockets de type Raw pour l'interception et la réinjection du trafic impose l'utilisation de notre logiciel sur un système Linux, avec un utilisateur disposant de droits privilégiés. Néanmoins, une option « acheminer uniquement » existe dans notre logiciel pour stipuler que le noeud sur lequel il est utilisé ne doit pas jouer le rôle d'intercepteur ou d'injecteur de trafic. Dans ce cas, les mécanismes liés au système d'exploitation Linux ne sont plus utilisés et notre logiciel peut être utilisé sur tout type de système supportant le langage Java, avec un utilisateur non privilégié. Dans ce cas, le noeud ne peut pas se charger de la protection des communications, mais peut uniquement relayer les messages utilisés par notre système.

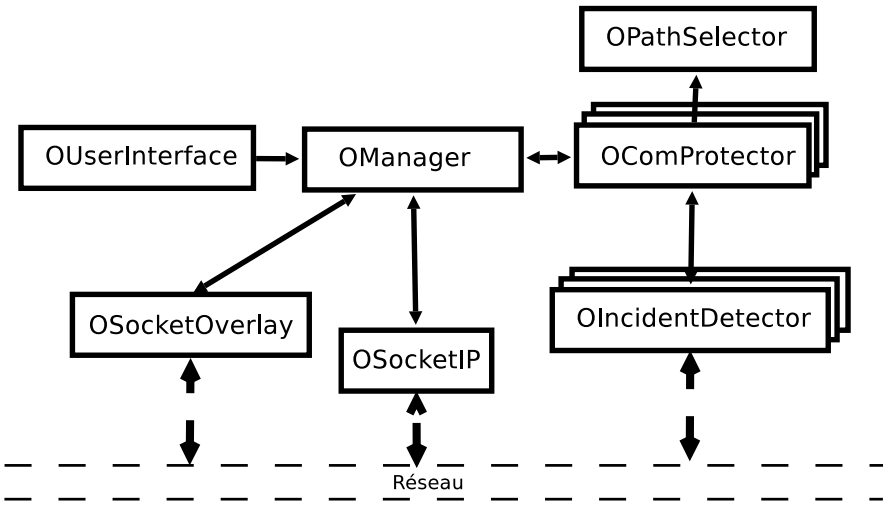


FIG. 4.1: Architecture du logiciel de routage P2P

4.4 Évaluation

Dans cette section, nous allons présenter les résultats de nos évaluations du système de rétablissement overlay des communications. Pour permettre cette évaluation, nous avons déployé notre logiciel sur plusieurs machines dans un réseau virtualisé, mais aussi sur le réseau Internet.

4.4.1 Critères de performance étudiés

Afin d'évaluer notre système, nous étudierons ses performances selon les critères de performances des mécanismes de rétablissement, abordés dans la section 2.4.1. Ce sont en effet ces critères qui doivent être étudiés pour démontrer l'efficacité d'un système de rétablissement réseau. Dans cette section, nous nous consacrerons particulièrement à l'étude du temps de rétablissement nécessaire au mécanisme pour rétablir une communication affectée par un incident ainsi qu'aux ressources consommées par notre système. Les autres points seront abordés brièvement ici, mais étudiés en détail dans le chapitre suivant.

4.4.2 Plateformes de test

Deux plateformes de test ont été mise en place pour évaluer notre système. Ces tests ont été réalisés sur un réseau virtualisé et sur Internet. Par conséquent, ils ont été effectués en conditions « réelles » : contrairement à une simulation, nous avons effectué des mesures sur du vrai trafic réseau, les applications utilisées dans nos tests ne sont pas modifiées spécifiquement pour celui-ci, et enfin, notre logiciel a été déployé sur différents systèmes exécutant des systèmes d'exploitation existants.

Réseau virtualisé

La première plateforme utilisée pour nos tests est un réseau de machines virtualisées. La virtualisation permet d'exécuter plusieurs systèmes à l'intérieur d'une même machine physique. Nous avons utilisé une solution de virtualisation basée sur OpenVZ[69] pour nos tests. Les machines exécutent le système d'exploitation Linux.

Le réseau virtualisé est composé de 30 machines, liées en réseau. Afin d'émuler le comportement des machines utilisateurs communiquant entre elles via Internet, nous les avons toutes connectés à un réseau virtuel, implémenté sur le système « hébergeant » les machines virtuelles par un pont réseau sur lequel est relié les interfaces réseaux virtuelles de chacune de ces machines. Les machines sont ainsi toutes capables de communiquer deux à deux.

Pour émuler le délai d'acheminement des communications entre deux machines, nous avons, comme dans le chapitre précédent, utilisé une distribution de Pareto paramétrée par les paramètres D et V. D représente alors le délai minimum d'acheminement aller-retour entre deux machines, et V le paramètre de variation, afin que $D+V$ soit le délai d'acheminement aller-retour moyen observé. Afin de déterminer ces paramètres pour chaque pair de machines du réseau, nous leur avons attribué aléatoirement des coordonnées dans un espace à deux dimensions et nous avons utilisé un délai d'acheminement minimal D proportionnel à leur distance dans l'espace à deux dimensions. Cette proportion est définie de telle sorte que la plus grande distance possible entre deux machines corresponde à un délai D de 500 ms. Le paramètre V est ensuite choisi comme égal à $D/10$. Pour introduire ce délai d'acheminement dans le réseau virtuel, nous avons utilisé l'outil « Linux Traffic Control » [29].

Plusieurs milliers de scénarios de test différents ont été mesurés sur cette plateforme.

Réseau Internet

Nous avons aussi réalisé des tests sur un réseau de 8 machines reliées à Internet par 5 fournisseurs européens d'accès à Internet distincts.

Quelques centaines de scénarios de test ont pu être mesurés sur cette plateforme.

4.4.3 Scénarios de test

Afin de réaliser notre évaluation, nous avons mis en place des scénarios de test. Le déroulement d'un scénario de test est composé des étapes suivantes :

- l'établissement d'une communication à protéger entre deux noeuds dans le réseau
- la déclaration du niveau de fiabilité demandé pour cette communication par l'utilisateur à notre système
- la mise en place par le système des mécanismes pour la protection de la communication
- l'apparition d'un incident affectant la communication
- le rétablissement de la communication

Par conséquent, les différents paramètres qui varieront au cours des différents scénarios sont les suivants :

- Les machines qui réalisent la communication à protéger
- Le temps maximum d'interruption toléré (TMIT) demandé par l'utilisateur
- Le moment d'apparition d'un incident affectant la communication

Le moment d'apparition de l'incident doit être suffisamment grand pour que l'ensemble des opérations de mise en place de la protection de la communication par le système ait été accompli. Il est ainsi choisi aléatoirement et l'incident est provoqué dans le réseau de test.

4.4.4 Résultats des mesures

Nous allons maintenant présenter les résultats des mesures effectuées durant nos différents scénarios de test.

4.4.5 Temps de rétablissement réseau

Observation des temps de rétablissement obtenus

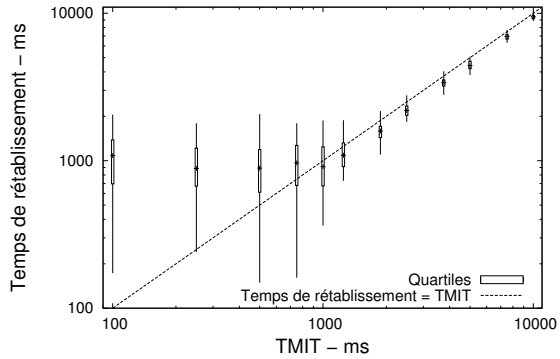
Nous allons étudier dans cette section les temps de rétablissement réseau obtenus par notre système. Nous rappelons que le temps de rétablissement réseau est le temps pendant lequel la livraison d'une communication à son destinataire est interrompue, lorsque celle-ci est affectée par un incident. Le rôle de notre système est de permettre d'acheminer à nouveau les communications de manière à satisfaire les besoins de l'utilisateur. Pour cela, notre système doit permettre le rétablissement des communications en un temps inférieur au TMIT, le temps maximal d'interruption toléré spécifié préalablement par l'utilisateur.

Les graphiques 4.2a,4.2b,4.2c et 4.2d montrent la répartition des temps de rétablissement, noté T_{ret}, en fonction du temps TMIT désiré. Ces graphiques représentent respectivement les résultats des mesures effectuées dans les situations suivantes : réseau virtualisé sans utilisation de l'état « Attention » du mécanisme de détection d'incident, réseau virtualisé avec utilisation de l'état « Attention » du mécanisme de détection d'incident, réseau internet sans utilisation de l'état « Attention » du mécanisme de détection d'incident, réseau virtualisé avec utilisation de l'état « Attention » du mécanisme de détection d'incident. Les répartitions des temps de rétablissement observés sont exprimées sous forme de quartiles : pour chaque valeur de $TMIT$ étudiée, le trait supérieur indique les valeurs des 25 % plus grands temps de rétablissement observés, le trait inférieur indique les valeurs des 25 % plus faibles temps de rétablissement observés, la « boite » indique les valeurs des 50 % temps de rétablissement observé restant et la croix indique la valeur médiane. De plus, sur chaque graphique, la droite $T_{ret} = TMIT$ est tracée. Ainsi, les temps de rétablissement se trouvant en dessous de cette droite indiquent que le système a réussi à rétablir les communications en un temps inférieur au TMIT.

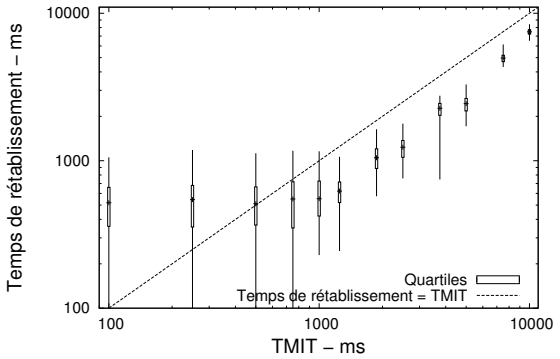
Lorsque l'état « Attention » du mécanisme n'est pas utilisé, on observe dans les graphiques 4.2a et 4.2c, qu'il faut que le TMIT soit supérieur à 1 seconde pour que le temps de rétablissement soit inférieur à ce TMIT dans au moins la moitié des scénarios. En effet, il est rare d'observer des temps de rétablissement inférieur à 1 seconde. Lorsque le TMIT vaut 2 secondes, on observe dans environ 75 % des scénarios un temps de rétablissement inférieur au TMIT. Enfin, pour permettre un rétablissement en un temps inférieur au TMIT pour tous les scénarios, il faut que le TMIT soit de l'ordre de 5 secondes environ. Lorsque l'état « Attention » du mécanisme de détection d'incident est utilisé, les temps de rétablissement sont plus courts : il suffit que le TMIT soit supérieur à 500 ms pour que dans la moitié des scénarios le temps de rétablissement lui soit inférieur. On observe de plus que pour un TMIT légèrement supérieur à 1 seconde, le temps de rétablissement sera inférieur à ce temps dans la totalité des scénarios. On observe de plus que les temps de rétablissement sont largement inférieurs au TMIT lorsque celui-ci est supérieur à quelques secondes.

Lorsque l'on compare les résultats obtenus dans le réseau virtualisé à ceux obtenus dans le réseau Internet, on ne constate que de légères différences : les temps de rétablissement observés dans le réseau Internet semblent légèrement plus élevés, et pour un même TMIT, leur répartition est plus variée, c'est-à-dire que les temps observés sont moins équitablement répartis entre le temps minimum et le temps maximum observés qu'avec le réseau virtualisé. On peut attribuer ce phénomène à la plus grande variété des délais d'acheminements entre les noeuds dans le réseau Internet, pour qui l'inégalité triangulaire n'est pas respectée [111, 81]. Cependant, les résultats obtenus dans ces deux réseaux sont assez similaires, ce qui nous permet de confirmer la validité des expérimentations menées dans le réseau virtualisé.

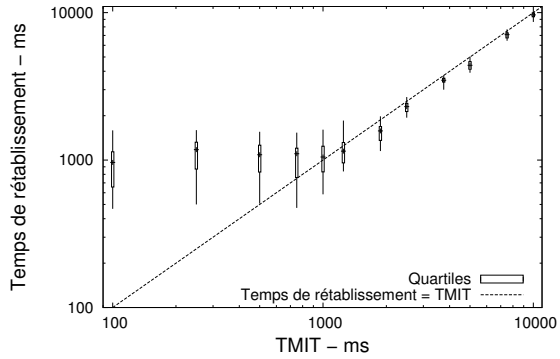
On constate que l'utilisation de l'état « Attention » du mécanisme de détection d'incident pour anticiper les opérations de rétablissement et enclencher le double acheminement permet une plus



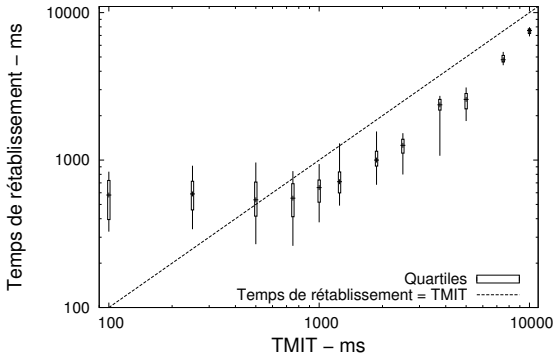
(a) Réseau virtualisé - Pas d'utilisation de l'état «Attention»



(b) Réseau virtualisé - Utilisation de l'état «Attention»



(c) Réseau Internet - Pas d'utilisation de l'état «Attention»



(d) Réseau Internet - Utilisation de l'état «Attention»

FIG. 4.2: Répartition des temps de rétablissement mesurés en fonction du temps *TMIT* demandé, pour les différentes plateformes de test, avec ou sans l'utilisation de l'état «Attention» du mécanisme de détection d'incident.

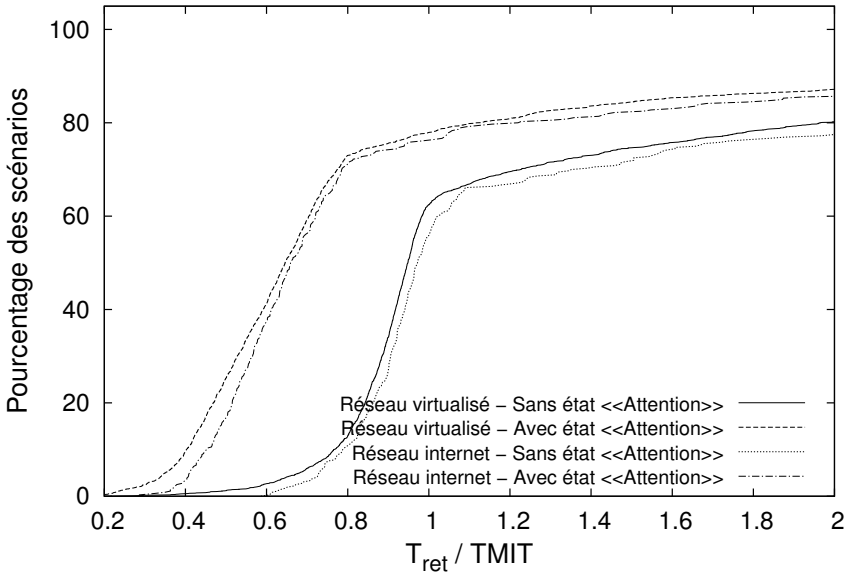


FIG. 4.3: Fonction de répartition du rapport entre le temps de rétablissement et le $TMIT$ demandé

grande chance pour le système de réussir à rétablir les communications en un temps inférieur au TMIT, particulièrement lorsque celui-ci est compris entre 500 ms et 1 seconde. Afin de confirmer ce constat, le graphique 4.3 montre la fonction de réparation des valeurs du temps de rétablissement T_{ret} divisé par le TMIT, pour l'ensemble des scénarios mesurés. Un rapport $\frac{T_{ret}}{TMIT}$ supérieur à 1 indique que le système n'a pas été en mesure de rétablir les communications en un temps inférieur au TMIT demandé par l'utilisateur. Les différentes courbes représentent les différentes situations dans lesquels ont été réalisées les mesures : réseau Internet ou virtualisé et utilisation ou non de l'état « Attention ». Dans ce graphique, le pourcentage des scénarios indiquant un temps de rétablissement inférieur au TMIT n'est que peu représentatif des réelles performances du système. En effet, les mesures sont réalisées sur l'ensemble des scénarios qui utilisaient des TMIT variés. Les informations à retenir dans ce graphique sont par conséquent les différences entre les situations représentées par les écarts entre les différentes courbes.

Dans ce graphique, on observe clairement le bénéfice apporté par l'utilisation de l'état « Attention ». Lorsqu'il n'est pas utilisé, le temps de rétablissement était inférieur au TMIT dans environ 60 % des scénarios mesurés. Ce nombre passe à 75 % environ avec l'utilisation de l'état « Attention ». On observe de plus que les temps de rétablissement sont plus souvent largement inférieurs avec l'utilisation de l'état « Attention » que sans : par exemple, lorsqu'il est utilisé, le temps de rétablissement est inférieur de moitié au temps TMIT pour 20 % des scénarios environ, alors que s'il ne l'est pas, ce nombre est quasi nul. On observe enfin, comme précédemment que l'écart entre les mesures réalisées dans le réseau Internet et celles réalisées dans le réseau virtualisé est faible.

Nous avons pu constater que notre système est capable de rétablir une communication affectée par un incident en moins d'une seconde. Pour cela, l'utilisation de l'état « Attention » est très conseillée. Il permet en effet, en anticipant les opérations de rétablissement lorsque la présence d'un incident est soupçonnée, d'assurer un temps de rétablissement inférieur à une seconde et de permettre un temps de rétablissement inférieur à 500 ms dans environ un cas sur deux. L'utilisation de cet état n'est par contre pas nécessaire lorsque le TMIT demandé par l'utilisateur est de plusieurs secondes. Nous verrons dans

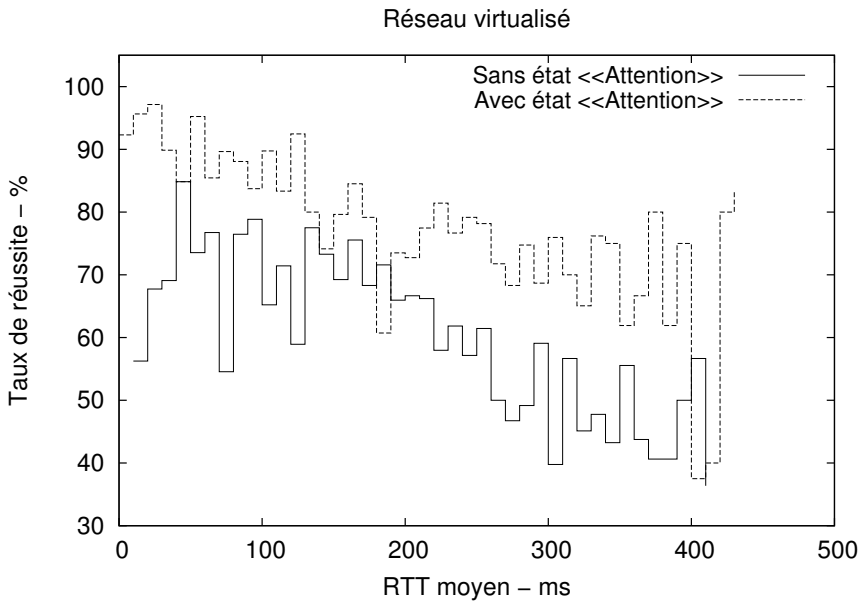


FIG. 4.4: Taux de réussite du rétablissement en fonction du RTT entre les noeuds impliqués dans la communication

la suite l'impact qu'a l'utilisation de l'état « Attention » sur la consommation de ressource réseau.

Influence du délai d'acheminement sur les temps de rétablissement

Nous allons étudier l'influence de la « distance », c'est-à-dire le temps d'acheminement aller-retour (RTT) moyen, entre les noeuds impliqués dans la communication à protéger dans le réseau, sur le temps de rétablissement. Le graphique 4.4 montre le taux de réussite, c'est-à-dire le nombre de fois que notre système a rétabli une communication en un temps inférieur au TMIT demandé en fonction du RTT moyen entre les noeuds impliqués dans cette communication. Les différentes courbes indiquent les résultats avec l'utilisation ou non de l'état « Attention ». Seules les mesures effectuées dans le réseau virtualisé ont été représentées, les scénarios dans le réseau Internet étant trop peu nombreux pour produire des résultats significatifs.

Tout d'abord, on observe comme précédemment un écart important entre le taux de réussite obtenu avec l'utilisation de l'état « Attention » et celui obtenu sans. Cet écart, variable, est d'environ 20 %. Ensuite, on observe que le taux de réussite décroit légèrement lorsque le RTT augmente : par exemple, avec l'utilisation de l'état « Attention » le taux de réussite est supérieur à 85 % lorsque le RTT est inférieur à 100 ms, compris entre 70 % et 80 % lorsque le RTT est compris entre 200 ms et 300 ms et diminue encore légèrement si le RTT est supérieur. Une diminution similaire du taux de réussite est observée lorsque l'état « Attention » n'est pas utilisé.

Le délai d'acheminement moyen des communications entre les noeuds réalisant la communication à protéger a par conséquent une influence sur le taux de réussite du mécanisme de rétablissement, c'est-à-dire sur les chances de voir une communication affectée par un incident rétablie en un temps inférieur au TMIT spécifié par l'utilisateur. Ceci peut être compris par les opérations effectuées par les différentes étapes du processus de rétablissement. Le délai d'acheminement a en effet une influence sur chacune de ces étapes. Par exemple, nous avons vu dans le chapitre précédent que les performances

des mécanismes de détection d'incident par envoi de messages sondes sont meilleures lorsque le délai d'acheminement entre les noeuds est court. De même, les étapes de choix du chemin alternatif et de réacheminement, qui termine le processus de rétablissement dans notre système, seront plus longues lorsque le délai d'acheminement moyen entre les noeuds est long. Par conséquent, il apparait que notre mécanisme fonctionne mieux lorsque le délai d'acheminement entre les noeuds est faible. Cependant, puisque les différentes étapes du processus de rétablissement qui sont nécessaires au rétablissement de la communication sont liées à ce délai, on peut considérer que cette diminution des performances observée est normale : de nombreux systèmes déployés sur ce type de réseau voient les performances être affectées par le délai d'acheminement.

4.4.6 Ressources réseau consommées

Dans cette section, nous allons étudier la consommation de ressources réseau, et plus particulièrement la bande passante, nécessaire au bon fonctionnement de notre système. Les différentes sources de consommation de la bande passante sont :

- La bande passante nécessaire au mécanisme de détection d'incident
- La bande passante nécessaire à la gestion du réseau overlay
- La bande passante supplémentaire due aux entêtes ajoutés aux paquets des communications protégées lors de leur acheminement par le routage P2P.
- Éventuellement, la bande passante supplémentaire consommée lors de l'acheminement de la communication une deuxième fois sur le chemin de secours, si l'état « Attention » et le double acheminement sont utilisés.

Ressources réseau consommées par les mécanismes de détection d'incident

La bande passante consommée par le mécanisme de détection d'incident dépend bien entendu du mécanisme de détection d'incident utilisé. Si l'on note $c(TMD)$ la bande passante nécessaire au fonctionnement du mécanisme pour détecter un incident avec un temps maximum de détection TMD, la bande passante C consommée par les mécanismes de détection d'incident sur le chemin principal et les k chemins de secours potentiels dédiés à la protection d'une communication de temps maximum d'interruption toléré TMIT est :

$$\begin{aligned} C(TMIT) &= C(TMD+\alpha) \\ &= c(TMD) + k.c(\beta.TMD) \\ &= c(TMD) + k.c(k.TMD) \end{aligned}$$

Nous allons étudier l'évolution de la consommation de bande passante par les mécanismes de détection d'incident déployés par notre système en fonction du TMIT demandé par l'utilisateur. Pour cela, nous allons réutiliser les résultats des mesures effectuées dans la partie précédente. En effet, dans notre implémentation utilisée pour ce test, les mécanismes de détection d'incident utilisés sont les mêmes que ceux étudiés dans le chapitre 3 : il s'agit du mécanisme APull lorsque le TMD est inférieur à 1 seconde ou du mécanisme Pull, dans le cas contraire. Le graphique 4.5 présente la consommation de bande passante, en fonction du TMIT demandé, réalisée par le mécanisme de détection d'incident déployé sur le chemin principal, par un des mécanismes de détection déployé sur un chemin de secours potentiel, ainsi que par le total composé du mécanisme déployé sur le chemin principal et des $k = 8$ mécanismes déployés sur les chemins de secours potentiels.

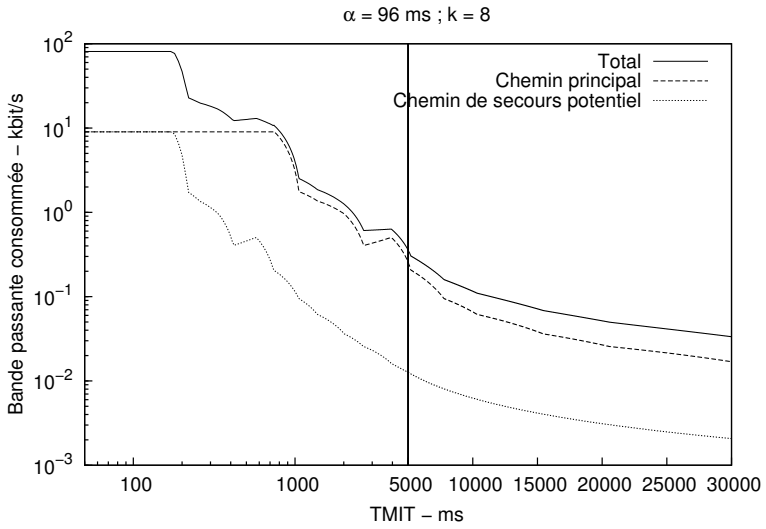


FIG. 4.5: Bande passante consommée par les mécanismes de détection d'incident, en fonction du temps $TMIT$ demandé

On observe une diminution de la bande consommée par les mécanismes de détection d'incident lorsque le temps TMIT augmente, lorsque le TMIT est supérieur à 1 seconde. Dans le cas contraire, la consommation de bande passante est plus stagnante : en effet dans ce cas, le mécanisme déployé sur le chemin principal fonctionne « au maximum », et il ne peut consommer plus. Ceci se produit aussi pour les mécanismes déployés sur les chemins de secours potentiels lorsque le TMIT est inférieur à 200 ms. Dans ce cas, la consommation de bande passante est d'environ 80 kbit/s. Lorsque le TMIT vaut 1 seconde, elle est d'environ 4 kbit/s. Elle diminue ensuite régulièrement avec l'augmentation du TMIT : elle est de 0.35 kbit/s pour un TMIT de 5 secondes et 0,12 kbit/s pour un TMIT de 10 secondes.

La consommation de bande passante par les mécanismes de détection d'incident dépend par conséquent de la demande de fiabilité par l'utilisateur, donnée par le TMIT. En effet, si cette demande est forte, la consommation le sera aussi. Par contre, si la demande de fiabilité est modérée, la consommation de bande passante sera faible. Il faut souligner que même si le TMIT demandé est faible, la consommation totale de bande passante par les mécanismes de détection d'incident reste tolérable dans les réseaux IP actuels.

Nous avons vu que le paramètre β a une influence sur la consommation de bande passante des mécanismes de détection d'incident sur les chemins de secours potentiels. Nous avons choisi d'utiliser la valeur $\beta = k$ en particulier dans le but de compenser la consommation lors de l'utilisation d'un nombre k grand. En effet, plus k, le nombre de chemins de secours potentiel est grand et plus le TMD des mécanismes de détection déployés sur ces chemins est grand, ce qui réduit leur consommation de bande passante. Le graphique 4.6 présente la consommation totale de bande passante par les mécanismes de détection d'incident en fonction de k, pour diverses valeurs de TMIT.

On observe que lorsque k n'est pas trop faible, ce qui correspond à une utilisation réaliste de notre système, la bande passante consommée par les mécanismes de détection d'incident varie peu avec le nombre k de chemins de secours potentiels. Une légère diminution de la consommation peut

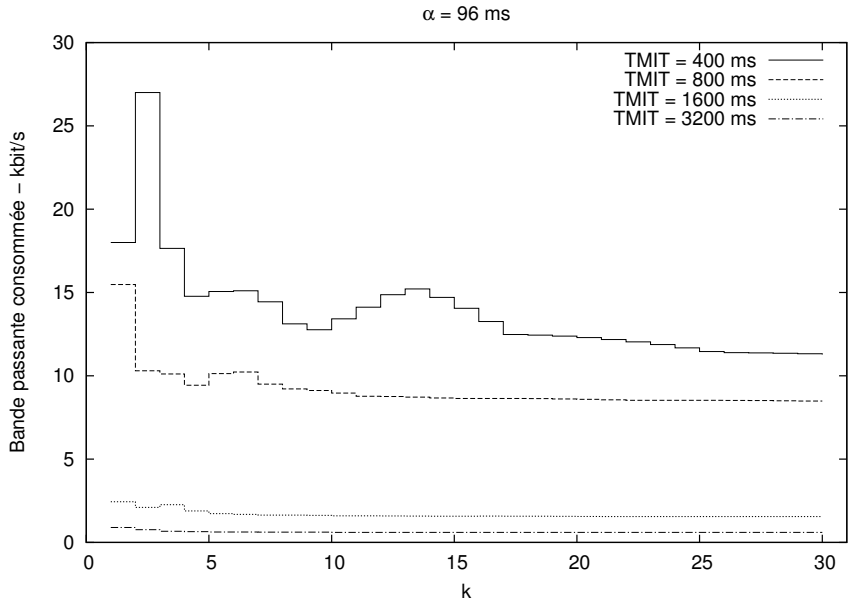


FIG. 4.6: Bande passante consommée par les mécanismes de détection d'incident, en fonction du nombre *k* de chemins de secours potentiels

même être observée. Par conséquent, nous considérons que le choix de la valeur *k* pour la constante β est intéressant : il permet l'utilisation d'un grand nombre de chemins de secours potentiels, tout en gardant une maitrise des ressources consommées par les mécanismes de détection d'incident.

Pour conclure cette section, soulignons que la bande passante consommée globalement par notre système par les différents mécanismes de détection d'incident ne dépend que des besoins des utilisateurs en terme de fiabilité pour ses communications et du nombre de communications à protéger. On peut ainsi considérer que la bande passante consommée par les mécanismes de détection d'incident est proportionnelle aux besoins des utilisateurs et que le passage à l'échelle de cette partie du système est assuré.

Ressources réseau consommées pour la gestion du réseau overlay

La bande passante nécessaire à la gestion du réseau overlay est consommée par les messages de type OVERLAY, décrits dans la section 4.3.1. Ces échanges sont limités : ils ne s'effectuent que lors de l'arrivée et du départ des noeuds. Cependant, lors de l'arrivée ou du départ d'un noeud, ces échanges s'effectuent entre ce noeud et tous les autres. De plus, la récupération de la liste des noeuds présents dans le réseau overlay lors de l'arrivée d'un noeud peut être consommatrice de ressource si le nombre de noeuds présent dans la liste est important, ce qui est le cas si le nombre de noeuds présent dans le réseau est grand et si la liste contient l'ensemble de ces noeuds.

Dans le contexte d'utilisation de notre système, nous pensons néanmoins que la bande passante réseau consommée par la gestion du réseau overlay peut être négligée lors de l'étude des ressources réseau globalement consommées par notre système. En effet, nous destinons notre système à être déployé sur des noeuds qui coopèrent pour fiabiliser leurs communications sensibles. Pour permettre cela, le nombre de noeuds participant au réseau overlay n'a pas besoin d'être important (nous justifierons ce point dans le chapitre 5) et de plus, ce type d'utilisation entraine une fréquence d'arrivée et

de départ des noeuds dans le réseau overlay plutôt faible. Si toutefois la consommation de ressources réseau due à la gestion du réseau overlay s'avérait trop importante, il serait envisageable de limiter le nombre de noeuds dont a connaissance chacun des noeuds du réseau overlay de manière à limiter ces échanges de messages.

Ressources réseau consommées par l'ajout des entêtes aux données

Nous allons maintenant nous intéresser à la bande passante additionnelle qui est consommée par les entêtes ajoutés aux paquets lorsqu'ils sont transportés par le réseau overlay. Nous rappelons que ces entêtes sont uniquement ajoutés lors de l'acheminement des données par le système de routage P2P sur le chemin de secours, donc uniquement lorsque la route de niveau IP est affectée par un incident.

Ce type de message, appelé message DATA, contient les informations suivantes :

- L'indication que le message est un message de type DATA
- La description de la communication à protégée
- Le nombre de noeuds overlays à traverser pour acheminer le paquet jusqu'à la destination
- Les adresses des noeuds overlays à traverser
- Le paquet de données à protéger

Avec notre implémentation actuelle, voici comment sont représentées ces informations :

- L'indication du type de message : un champ de 3 octets
- La description de la communication à protéger : les adresses IP des noeuds sources et destination de la communication, pour une taille totale de 8 octets
- Le nombre de noeuds overlays à traverser : un champ de 1 octet
- Les adresses de noeuds overlays à traverser : les adresses IP des noeuds, soit 4 octets par noeud à traverser
- Le paquet de données à protéger : C'est un paquet IP comprenant un entête de 20 octets et des données utiles de taille arbitraire

Dans la suite de nos mesures, nous considérerons que 3 noeuds overlays sont traversés lors de l'acheminement d'une communication sur un chemin de secours : le noeud overlay source, le noeud relai, par qui transite la communication et le noeud destination. En effet, comme montré dans les sections 2.5.3, 5.2.3 et 5.4.3, un seul noeud relai est généralement nécessaire pour contourner un incident. Par conséquent, avec notre implémentation, la taille de l'entête ajoutée à chaque paquet IP lors de leur acheminement par le système de routage P2P est de $3+8+1+4.3=24$ octets. De plus, chaque message de ce type est transporté par un paquet IP utilisant le protocole de transport UDP, ce qui ajoute un entête d'une taille de 28 octets.

Cependant, il est tout à fait envisageable de diminuer cette taille en modifiant l'implémentation. Il serait ainsi possible, par exemple, de représenter les informations de l'entête de la manière suivante :

- L'indication du type de message : un champ de 1 octet
- La description de la communication à protéger peut être supprimée de l'entête, car elle peut être retrouvée dans l'entête du paquet IP contenu dans le message
- Le nombre de noeuds overlays à traverser : un champ de 1 octet
- Les adresses des noeuds à traverser : des adresses propres au réseau overlay, de taille 1 octet, peuvent être utilisées pour représenter les noeuds.
- Le paquet IP à protéger.

Ainsi, dans les mêmes conditions que précédemment, la taille d'un tel entête serait de $1+1+3.1=5$ octets.

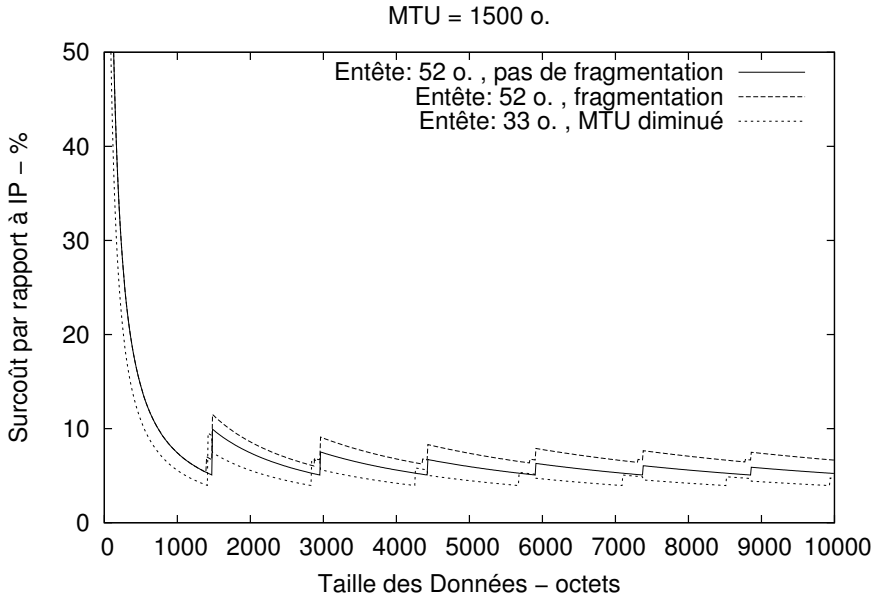


FIG. 4.7: Surcoût en bande passante entrainé par l'ajout d'entêtes pour le routage P2P

Il est aussi envisageable d'utiliser d'autres techniques d'implémentation pour le routage dans notre système de routage P2P, telles que le routage IP par la source (voir la section 2.4.2) ou les tunnels IP afin de diminuer la taille de l'entête.

L'ajout d'un entête aux paquets IP peut conduire à un autre phénomène susceptible de consommer des ressources réseau : la fragmentation des paquets. Un équipement réseau impose en effet une taille maximale pour un paquet IP : c'est le Maximum Transmission Unit (MTU). Si la taille d'un paquet IP est supérieure au MTU d'un équipement réseau, le paquet IP doit être fragmenté : ses données utiles sont séparées et réparties dans plusieurs paquets IP. Les données sont ensuite « réassemblées » par le noeud destination de la communication.

La fragmentation peut survenir dans notre système si un paquet IP, de taille inférieure au MTU, devient plus importante que le MTU une fois celui-ci incorporé dans un message overlay puis transporté par un nouveau paquet IP et UDP. Une manière de parer à ce problème est d'informer l'application que le MTU du réseau est inférieur à sa valeur réelle de manière à ce qu'un paquet IP de la taille du MTU « diminué » incorporé dans un entête pour le routage par le système de routage P2P soit de taille inférieure ou égale au MTU réel du réseau.

La figure 4.7 présente le surcoût en bande passante consommée par les paquets transmis dans le réseau lors de l'utilisation du routage P2P en fonction de la taille des données à transférer. Les données comprennent les entêtes de la couche transport et les informations des couches supérieures. Le surcoût est mesuré par rapport à la bande passante qui serait utilisée pour la transmission des mêmes données dans un paquet IP standard. Plusieurs implémentations possibles pour l'entête overlay sont représentées : Un entête d'une taille de 52 octets (telle qu'utilisée dans notre implémentation du système) sans prise en compte de la fragmentation IP, un entête de 52 octets avec prise en compte de la fragmentation IP et enfin une implémentation « optimisée » : un entête de 52 octets avec un MTU corrigé à 1444 octets afin d'éviter la fragmentation des paquets IP.

On observe que, suivant les implémentations, le surcoût de l'acheminement dans le réseau overlay dû aux entêtes est compris entre environ 5 % pour l'implémentation « optimisée » et 10 % pour

l'implémentation actuelle avec fragmentation. L'augmentation périodique du surcoût est observée dès que la quantité de données à envoyer nécessite la création d'un nouveau paquet. On observe enfin que le surcout tend à converger lorsque la quantité de données à envoyer grandit. Le surcoût se stabilise ainsi à 5,1 % avec l'implémentation utilisant un entête de 52 octets et sans fragmentation, 6,7 % avec la même implémentation et la fragmentation IP et enfin 4,0 % avec l'implémentation « optimisée ».

Le surcoût de consommation de bande passante dû à l'encapsulation des paquets pour être pris en charge par le système de routage P2P est donc limité à moins de 7 % dans tous les cas. Ce surcoût nous semble raisonnable, d'autant qu'il peut être diminué par des implémentations plus optimisées. De plus, ce surcout n'est présent que pour le trafic acheminé dans le réseau overlay, donc lorsqu'un incident affecte le réseau ou dans le cas de l'utilisation du double acheminement, lorsque l'état « Attention » a été déclaré par le mécanisme de détection d'incident. Par conséquent, ces ressources ne sont pas consommées en permanence, mais uniquement lorsque la fiabilité des communications le justifie.

Ressources réseau consommées par le double acheminement

Pour terminer notre étude des ressources réseau consommées par notre système, nous allons étudier les ressources consommées par le mécanisme de double acheminement. Lorsqu'il est utilisé, l'état « Attention » du mécanisme de détection d'incident déployé sur le chemin principal déclenche la recherche d'un chemin de secours valide, puis l'acheminement supplémentaire de la communication par ce chemin, sans interrompre l'acheminement de la communication sur le chemin principal, jusqu'à ce la présence de l'incident ait été confirmée par le mécanisme de détection. L'acheminement double de la communication entraîne une consommation de bande passante supplémentaire que nous allons mesurer.

Afin de mesurer les ressources consommées par le double acheminement d'une communication, il est nécessaire de mesurer les périodes durant lesquelles ce double acheminement est effectué. Deux cas peuvent se produire :

- Le double acheminement est consécutif d'un état « Attention » déclaré par le mécanisme de détection d'incident en réaction à l'apparition d'un incident qui nécessite le réacheminement des communications par notre système. Dans ce cas, le double acheminement est qualifié de légitime.
- Le double acheminement est consécutif d'un état « Attention » déclaré par le mécanisme de détection d'incident qui ne sera pas suivi par la confirmation de la présence d'un incident nécessitant le réacheminement des communications. Ceci peut se produire en présence d'un incident ponctuel qui entraîne la perte d'un message sonde utilisé par le mécanisme de détection, par exemple. Dans ce cas, le double acheminement sera qualifié d'illégitime.

Le graphique 4.8 présente la durée moyenne durant laquelle notre système réalise un double acheminement, légitime ou illégitime, en fonction du TMIT, sur une durée de 1 jour. Ces mesures ont été effectuées dans le réseau virtualisé. Afin de déclencher régulièrement l'état « Attention » du mécanisme de détection d'incident, des incidents aléatoires ont été introduits dans ce réseau de façon à ce que la disponibilité du chemin reliant les deux noeuds réalisant la communication à protéger soit de $A = 0.999$. La répartition de la durée des incidents suit le même procédé que celui déjà décrit dans le chapitre précédent à la section 3.3.3.

Tout d'abord, on observe qu'avec la configuration de réseau utilisée, la grande majorité des périodes de double acheminement sont de nature illégitime. En effet, quel que soit le TMIT, les périodes de double acheminement légitime sont environ 100 fois moindres que les périodes illégitimes. Lorsque le TMIT est inférieur à 1000 ms, le graphique montre que le double acheminement est effectué environ durant environ 150 minutes par jour. Cette durée décroît ensuite rapidement : elle est comprise

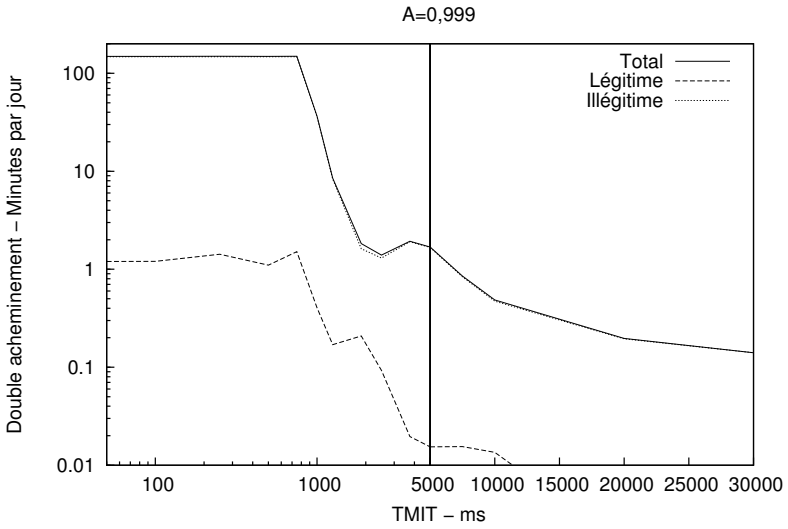


FIG. 4.8: Durée moyenne des périodes de double acheminement, en fonction du $TMIT$ demandé

entre 1 et 2 minutes lorsque le TMIT est situé entre 1 et 5 secondes, puis elle décroit pour ne plus être que de 8 secondes par jour avec un TMIT de 30 secondes.

Les périodes de double acheminement, qui sont pour la très grande majorité illégitimes donc inutiles, sont importantes pour les TMIT faibles. Ceci va conduire à une consommation inutile des ressources réseau, puisque, pendant le double acheminement, la communication est transmise deux fois dans le réseau. Ceci est particulièrement problématique lorsque la communication à protéger requiert une bande passante importante, mais le double acheminement reste nécessaire, en particulier lorsque le TMIT demandé par l'utilisateur est faible, puisqu'il permet un temps de rétablissement plus court. Si les périodes de double acheminement illégitimes sont si nombreuses, c'est à cause du mécanisme de détection d'incident utilisé. En effet, ce dernier déclenche l'état « Attention » précédent le double acheminement dès qu'un de leurs messages sondes est perdu. En fonction de la disponibilité du réseau, ceci peut se produire fréquemment, et plus particulièrement lorsque le TMIT est inférieur à une seconde, car c'est le mécanisme de type APull qui est utilisé.

La consommation de bande passante due au double acheminement n'est donc pas négligeable. De plus, cette consommation est le plus souvent inutile, puisqu'aucun incident n'affectait réellement la communication sur le chemin principal. Cependant, la bande passante ainsi consommée est très variable. Elle dépend du TMIT demandé par l'utilisateur, mais aussi de la disponibilité du réseau et de la bande passante requise pour acheminer la communication. Nous considérons cependant que cette consommation est nécessaire, étant donné les gains de temps de rétablissement apportés par le double acheminement.

Conclusion sur la consommation de ressources réseau

Afin de conclure cette section consacrée à l'étude des ressources réseau consommées par notre système, nous allons étudier un scénario d'exemple. Nous considérons une communication à protéger unidirectionnelle, réalisée entre deux noeuds reliés par un réseau de disponibilité $A = 0.999$, de délai

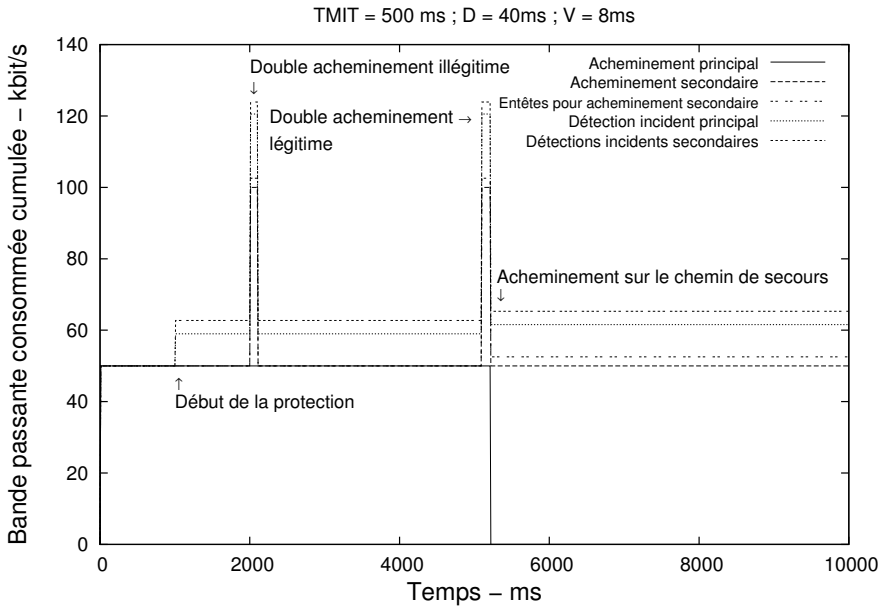


FIG. 4.9: Bande passante totale consommée par le système de routage P2P, au cours du temps

d'acheminement aller-retour minimal $D = 40ms$ et de délai d'acheminement aller-retour moyen $D + V = 48ms$. Le débit de la communication à protéger est constant et est de 50 kbit/s. Le graphique 4.9 présente la bande passante consommée dans le réseau par les différentes sources de consommation, en fonction du temps.

Au temps t=0, la communication commence et la seule bande passante utilisée est celle nécessaire à la communication. Au temps t=1000ms, notre système commence la protection de la communication et la bande passante supplémentaire nécessaire au fonctionnement des différents mécanismes de détection d'incident est utilisée dans le réseau. Au temps t=2000ms, le mécanisme de détection d'incident se place dans l'état « Attention » et un double acheminement débute. Ce double acheminement est illégitime puisqu'aucun incident qui justifie le réacheminement de la communication n'est présent dans le réseau. La bande passante supplémentaire consommée correspond alors à l'acheminement de la communication par le système de routage P2P, ainsi que les entêtes nécessaires à cet acheminement, ainsi qu'au mécanisme de détection d'incident déployé sur le chemin de secours utilisé. Le double acheminement cesse après 90 ms. Au temps t=5000ms, un incident suffisamment long pour justifier le réacheminement des communications par le réseau overlay apparait. Le double acheminement reprend alors durant 110 ms, puis l'incident est confirmé et le chemin de secours sélectionné. L'acheminement de la communication est réalisé sur ce chemin de secours. La bande passante est utilisée dans le réseau par les entêtes ajoutés aux paquets de la communication, ainsi qu'aux mécanismes de détection d'incident.

On peut constater que la consommation de ressources par notre système est modérée. En effet, la proportion de bande passante consommée par les mécanismes de notre système est faible comparée à la bande passante de la communication à protéger. En particulier, dans l'exemple étudié ici, la bande passante utilisée par la communication est relativement faible : 50 kbit/s et cette proportion aurait encore été plus basse pour une communication à protéger nécessitant un plus fort débit. Ce constat est moins vrai lors des périodes de double acheminement. Cependant, nous pouvons constater que ces périodes sont brèves. Par conséquent, si leur fréquence d'apparition n'est pas trop importante, la

consommation de ressource entrainée par ce phénomène reste limitée.

Nous allons maintenant conclure cette section consacrée à la consommation de ressource par notre système. Nous avons étudié en détail la consommation de bande passante entrainée par l'utilisation de notre système et montré qu'elle est essentiellement due aux mécanismes de détection d'incident, aux entêtes ajoutés aux paquets pour être acheminés dans le réseau overlay et au double acheminement de la communication. Par conséquent, la quantité de ressources consommées ne dépend pas du nombre de noeuds présents dans le système, mais est liée aux besoins de fiabilité des utilisateurs pour leurs communications. Plus particulièrement, c'est le nombre de communications à protéger ainsi que la fiabilité demandée pour ces communications qui influera sur la quantité de ressources consommées. Ainsi, le passage à l'échelle est possible, puisque les ressources consommées augmentent proportionnellement avec le nombre de communication présentent dans le réseau. Nous avons de plus mis en évidence des pistes pour réduire cette consommation, par exemple en optimisant la taille des entêtes ou en améliorant les mécanismes de détection d'incident afin de limiter les périodes de double acheminement illégitime.

4.4.7 Autres critères de performance

Nous allons brièvement discuter des autres critères de performance des mécanismes de rétablissement réseau, qui ont été présentés dans la section 2.4.1. Les observations effectuées durant les tests de ce chapitre ne permettent pas de mesurer en détail ces critères, cependant, nous étudierons certains d'entre eux dans le chapitre suivant.

La portée d'un mécanisme de rétablissement mesure la possibilité d'un mécanisme à rétablir une communication en réaction à l'apparition d'un incident. En fonction du fonctionnement du mécanisme de rétablissement et de la nature de l'incident, il n'est en effet pas toujours possible de rétablir un incident. Par exemple, pour qu'un mécanisme de routage soit en mesure de rétablir une communication, il faut qu'il existe un chemin alternatif dans le réseau et que le protocole de routage soit en mesure d'acheminer les communications sur ce chemin.

La capacité de notre système à rétablir une communication entre deux noeuds repose sur l'existence d'un chemin IP utilisable entre ces noeuds et un troisième noeud tiers. Cette possibilité dépend entre autres de la topologie du réseau reliant ces noeuds et de l'étendue de l'incident. Ainsi, les mesures effectuées dans cette partie ne nous permettent pas de nous prononcer sur la portée de notre système dans le cas d'une utilisation en condition réelle. En effet, les mesures effectuées dans le réseau virtualisé ne peuvent être significatives, car la topologie utilisée dans ce réseau n'est pas représentative de celle d'un réseau tel qu'Internet et de plus, les incidents ont été créés, et par conséquent leurs étendues étaient connues à l'avance. De plus, les mesures effectuées dans le réseau Internet ont été trop peu nombreuses pour être significatives. Nous pensons cependant que l'étude de la portée de notre mécanisme est essentielle et le chapitre suivant y sera grandement consacré.

Un autre critère de performance des mécanismes de rétablissement est la qualité des chemins alternatifs utilisés pour réacheminer une communication après que son chemin primaire est été affecté par un incident. De même que pour la portée du mécanisme, ce critère dépend fortement du réseau connectant les noeuds participants à notre système et de sa topologie. Par conséquent, les résultats obtenus dans le réseau virtualisé ne peuvent être utilisés pour mesurer ce critère. De même que précédemment, nous considérons que nos résultats obtenus sur le réseau Internet sont trop peu nombreux pour être interprétés. Cependant, ce critère de performance sera étudié dans le chapitre suivant.

Enfin, le dernier critère évoqué dans la section 2.4.1 concernait la stabilité du mécanisme, c'est-à-dire la capacité du mécanisme à ne déclencher le processus de rétablissement que lorsque cela est approprié, lorsque la durée de l'incident le justifie, par exemple. Dans notre système, cette propriété

repose sur le mécanisme d'incident utilisé. En effet, c'est ce dernier qui alerte le mécanisme pour enclencher le processus de rétablissement lorsqu'il détecte un incident entre deux noeuds réalisant une communication à protéger. L'implémentation actuelle de notre système utilise de la détection d'incident par envoi de messages sondes dans le réseau. Par conséquent, la stabilité de notre système dépend des performances de ces mécanismes, et en particulier de leur fréquence d'apparition de faux positifs. Comme nous l'avons vu, cette fréquence est basse lorsque le temps de détection maximum demandé est supérieur à une seconde. Dans le cas contraire, cette fréquence est relativement élevée : environ un faux positif toutes les trois minutes. Par conséquent, les performances de notre système en terme de stabilité, dans l'état actuel de son implémentation, sont bonnes lorsque le TMIT demandé par l'utilisateur est supérieur à 1 seconde. Elles le sont moins lorsque le TMIT est inférieur, mais ceci, dans l'état de nos connaissances, est nécessaire pour assurer un TMIT bas.

4.5 Conclusion

Dans cette partie, nous avons présenté un système de rétablissement réseau utilisant le routage P2P pour permettre le réacheminement des communications en cas d'incident. L'objectif de ce système est de permettre l'amélioration de la fiabilité des communications, décidée par les utilisateurs et déployable sur tout type de réseau IP. Ainsi, après avoir présenté le fonctionnement de ce système, nous avons mesuré son aptitude à rétablir une communication affectée par un incident, en fonction du besoin de fiabilité spécifié par l'utilisateur pour cette communication, exprimé par le temps maximum d'interruption toléré (TMIT).

Notre système utilise le routage P2P pour réacheminer le trafic affecté par un incident. Pour cela, la technique du routage par la source est utilisée en coordination d'un mécanisme de détection d'incident, présenté dans le chapitre précédent. De plus, pour permettre un rétablissement rapide des communications affectées par un incident, notre système maintient une liste de chemins de secours potentiels pouvant être utilisés en cas de défaillance du chemin principal. Enfin, l'utilisation d'un état « Attention », déclaré par le mécanisme de détection d'incident lorsque celui-ci soupçonne la présence d'un incident, permet d'accélérer le rétablissement d'une communication grâce au double acheminement de celle-ci.

Nous avons ainsi étudié la capacité de notre système, une fois que l'utilisateur a spécifié le temps TMIT pour une communication, à rétablir la délivrance de cette communication lorsque celle-ci est affectée par un incident. Nous avons observé que dans la plupart des cas, notre système était en mesure de rétablir une communication en un temps inférieur au TMIT lorsque celui-ci était de l'ordre de 1 seconde ou plus. De plus, avec l'utilisation du double acheminement des communications lors du passage du système à l'état « Attention » pour une communication, il est de satisfaire un TMIT d'environ 500 ms. Nous avons aussi étudié les ressources réseau consommées par note système et montré que cette consommation était principalement due au mécanisme de détection d'incident et dépendante du temps TMIT souhaité. Les performances de notre système nous semblent ainsi satisfaisantes, puisqu'il permet d'apporter plus grande fiabilité des communications affectées par un incident, en fonction des besoins spécifiés par l'utilisateur, et entrainant une consommation des ressources réseau en rapport avec le niveau de fiabilité demandé.

Nous pensons ainsi que l'utilisation de notre système permet une amélioration immédiate de la fiabilité des communications dans les réseaux IP . Quel que soit le type protocole utilisé avec IP, notre système peut être déployé. Si un incident se déclare au cours de la communication et nuit à son acheminement, notre système est capable, lorsque cela est possible, de rétablir cette communication en quelques centaines de millisecondes. Il est ainsi possible d'assurer la délivrance des services dont

la disponibilité est cruciale aux utilisateurs lors de l'apparition d'incident dans le réseau.

Le logiciel implémentant le système présenté dans ce chapitre, bien que pleinement fonctionnel, ne peut être considéré que comme un prototype. Pour permettre une utilisation plus massive, certains problèmes d'implémentation devront être corrigés. En particulier, il serait probablement bénéfique de réécrire le programme en un langage permettant une exécution plus rapide, afin de permettre une amélioration du débit d'acheminement des paquets par le logiciel. De plus, les mécanismes d'interception des paquets IP pour permettre leur prise en charge par notre système devraient être plus étudiés, afin par exemple de permettre une utilisation complète de notre logiciel sur des plateformes non Linux.

Afin d'améliorer encore la fiabilité des communications des utilisateurs, notre système pourrait utiliser d'autres techniques que le rétablissement réseau seul. En particulier, les méthodes d'augmentation de la qualité de service sur un réseau overlay, évoquées dans la section 2.5.1, pourraient être utilisées. De même, l'emploi de techniques de « network coding » [20], facilement déployables dans un réseau overlay, pourraient être envisagées.

L'étude des performances de notre système présentée dans cette partie n'est pas complète. En effet, nous n'avons présenté que les performances de notre système lorsque celui-ci était en mesure de rétablir une communication affectée par un incident. Cependant, cette possibilité n'est pas garantie, et dépend de nombreux paramètres, tels que la position de l'incident par rapport à la topologie du réseau et la localisation des noeuds dans celui-ci, ou encore le nombre de noeuds participants dans le réseau overlay. Il convient par conséquent, afin de pouvoir conclure sur les performances de notre système, d'étudier dans quelle mesure celui-ci peut rétablir une communication affectée par un incident. Ces travaux seront présentés dans le chapitre suivant.

Chapitre 5

Évaluation de la portée du routage P2P

5.1 Introduction

Lorsqu'une communication est affectée par un incident, les mécanismes de rétablissement vont permettre de réacheminer cette communication par un autre chemin du réseau, non affecté par l'incident. Ainsi, la transmission du trafic entre les noeuds réalisant cette communication sera rétablie et les services délivrés aux utilisateurs pourront être maintenus. Pour que cette opération soit possible, il est ainsi nécessaire qu'un tel chemin existe et qu'il puisse être utilisé par le mécanisme de rétablissement.

La portée d'un mécanisme de rétablissement, décrite dans la section 2.4.1 est un critère essentiel pour les performances des mécanismes de rétablissement.. La portée est la capacité du mécanisme à rétablir les communications après un incident, c'est-à-dire à proposer un chemin alternatif pouvant être utilisé pour réacheminer la communication.

Nous avons vu que les systèmes basés sur le routage P2P sont des mécanismes de rétablissement réseau intéressants, en particulier car le temps requit pour le rétablissement des communications peut-être très court. Cependant, il est essentiel de s'intéresser à la portée de ce type de mécanisme afin de compléter leurs études.

L'existence d'un chemin de rétablissement dépend bien sûr de la topologie du réseau : il faut qu'une succession de liens permettant l'acheminement des communications entre les noeuds existent dans le réseau. Cependant, cette condition n'est pas suffisante : il faut, de plus, pour qu'une communication soit effectivement rétablie, que le routage dans le réseau soit reconfiguré de manière à ce que la communication emprunte ce nouveau chemin.

Les protocoles de routage dynamiques classiques sont déployés sur l'ensemble des routeurs qui constitue les noeuds du réseau de l'opérateur. Pour assurer la connectivité entre les noeuds, l'ensemble des chemins possibles, constitués des liens liants ces routeurs, peut être utilisé. Mais les mécanismes de rétablissement sont déployés sur des réseaux de différentes échelles. Par exemple, avec le système de routage BGP, les noeuds du réseau sont en réalité les réseaux des opérateurs et les liens sont les accords établis entre les opérateurs pour échanger leur trafic. De même, avec les systèmes de routage P2P étudiés dans ce document, les noeuds du réseau sont les machines des utilisateurs et les liens sont les connexions entre ces machines.

Afin d'apporter une amélioration de la fiabilité des communications des utilisateurs, les systèmes de routage P2P doivent donc être en mesure de proposer un chemin de rétablissement lorsqu'un incident se déclare. Ceci paraît plus difficile pour ce type de système que les protocoles de routage IP. En effet, nous verrons que le routage P2P s'effectue à une échelle plus grossière que le routage IP. Par conséquent, il est plus difficile de trouver un chemin alternatif qui contourne l'incident dans le réseau overlay que dans le réseau IP. Dans ce chapitre, nous allons donc étudier la portée des systèmes de rétablissement basés sur le routage P2P, c'est-à-dire la possibilité de ce type de système de rétablir les communications après un incident réseau, en fonction de la situation.

Ainsi, nous allons voir qu'en fonction du nombre de noeuds participants au réseau overlay, de leur emplacement dans le réseau, ainsi que de la localisation et l'entendue de l'incident, il n'est pas toujours possible pour ces systèmes de rétablir une communication. Nous allons étudier dans quelle mesure un système de routage P2P est capable de rétablir une communication, en particulier en comparaison des systèmes de routage de niveau IP, en présence de différents scénarios d'incidents. Nous étudierons deux systèmes utilisant le routage P2P : un système basé sur le fonctionnement de RON et notre système présenté dans le chapitre 4. Ils seront comparés aux performances des protocoles de routage IGP et EGP traditionnellement utilisés dans les réseaux IP interconnectés.

Le reste de ce chapitre s'organise comme suit : nous nous intéresserons tout d'abord aux contextes de ces travaux, ainsi qu'aux autres travaux de recherche apparentés. Nous expliquerons ensuite la modélisation utilisée pour nos simulations. La section suivante sera consacrée à la présentation des

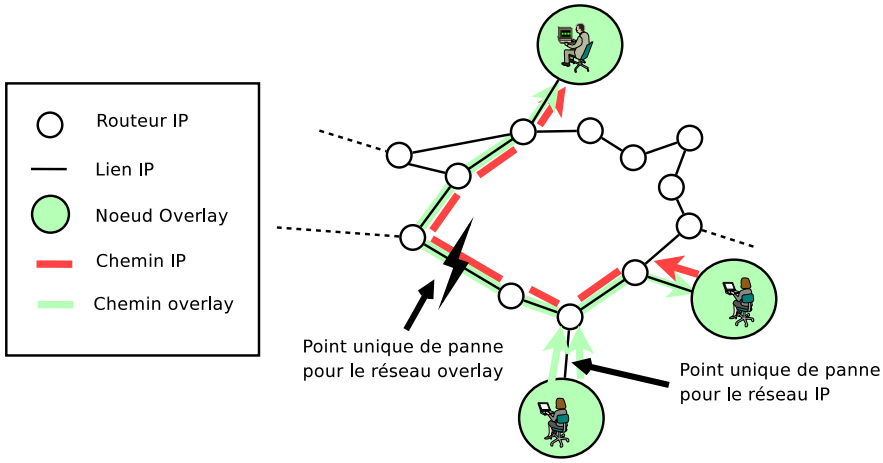


FIG. 5.1: Besoins topologiques du réseau IP pour permettre le rétablissement par le routage P2P

résultats obtenus. Nous conclurons enfin ce chapitre dans une dernière section.

5.2 Contexte et travaux apparentés

5.2.1 Prérequis pour le rétablissement P2P

Comme nous l'avions déjà signalé dans les sections 2.4.1 et 2.5.2, il existe des conditions liées à la topologie du réseau et à la localisation de l'incident pour qu'un mécanisme de rétablissement puisse proposer un chemin alternatif qui contourne cet incident. Pour qu'un mécanisme de rétablissement mis en oeuvre par le routage P2P puisse proposer un chemin alternatif entre deux noeuds de manière à contourner un incident, il doit exister un chemin entre ces noeuds dans la topologie overlay dont les différents liens empruntent des chemins du réseau sous-jacent qui ne sont pas affectés par l'incident. La figure 5.1 illustre cette condition.

Par conséquent, plusieurs facteurs vont avoir une influence sur la possibilité du routage P2P à rétablir une communication affectée par un incident :

- La localisation des noeuds overlays dans le réseau IP
- La topologie du réseau overlay
- Les routes calculées par le protocole de routage du réseau IP et empruntées par les liens overlays
- La localisation des équipements réseau affectés par l'incident dans le réseau sous-jacent

Ainsi, le routage P2P ne peut pas systématiquement rétablir la connectivité dans le réseau après un incident. Pour mesurer son efficacité, il faut donc étudier les situations où il va fonctionner et avec quelle fréquence.

5.2.2 Comparaison avec les autres mécanismes

Lorsque le mécanisme de routage P2P est déployé sur un réseau utilisant un protocole de routage dynamique, ces deux protocoles vont enclencher un mécanisme de rétablissement lorsqu'un incident se produit. On peut donc considérer que ces mécanismes sont en concurrence. Il nous parait donc intéressant d'étudier les interactions entre ces mécanismes et ainsi d'évaluer la plus-value apportée par le routage P2P.

L'utilisation du routage P2P pour le rétablissement réseau à un intérêt si, par rapport aux mécanismes de niveau IP, il apporte un temps de rétablissement plus court, pour un coût, en ressource réseau consommée, acceptable. Ceci peut se produire si le mécanisme de rétablissement déployé par l'opérateur dans le réseau IP est inefficace. Par exemple, celui-ci peut-être incapable ou trop long à rétablir une communication, s'il est mal conçu ou mal utilisé. C'est aussi le cas lors des incidents dits inter-AS, qui affectent plusieurs réseaux d'opérateurs différents. Dans ce cas, c'est généralement le protocole de routage EGP qui est sollicité, c'est-à-dire BGP. Les temps de rétablissement permis par ce protocole sont en général élevés.

Cette étude va ainsi permettre de déterminer dans quelles situations l'utilisation du routage P2P amène une amélioration de la fiabilité des communications de l'utilisateur, par rapport au routage classique utilisé aujourd'hui.

5.2.3 Travaux apparentés

La plupart des études réalisées sur la portée des mécanismes de routage P2P sont issues des mesures expérimentales réalisées par les auteurs de systèmes de routage P2P. Ainsi, une étude[19] réalisée par les auteurs de RON montre qu'en cas d'apparition d'un incident affectant une route du réseau entre 2 des 31 noeuds RON déployés, ce système a été capable de proposer l'utilisation d'une route alternative dans 50 % des cas. Il a aussi été mis en évidence que les noeuds « multi-homé », c'est-à-dire reliés à 2 réseaux d'opérateurs différents pour accéder à Internet, ont une probabilité plus grande (environ trois quarts des cas observés) de voir leurs communications rétablies après un incident[19, 101] en utilisant un système de routage P2P. De plus, il est montré que l'efficacité de RON n'est pas corrélée avec la durée de persistance d'un incident, ni avec la fréquence d'apparition d'incidents affectant la même route. Une autre étude[107], étudiant les incidents affectant les communications entre 120 noeuds, a estimé qu'une route alternative peut être proposée par un système tel que RON pour contourner un incident dans 43 % cas.

D'autres études[2] ont évalué l'efficacité du routage P2P en comparant différentes stratégies de routage. Pour cela, les taux de pertes de paquets des communications entre 30 noeuds ont été mesurés durant 14 jours. Il est montré que le taux de perte global est de 0,42 %, mais que l'utilisation de RON, en sélectionnant le chemin overlay dont le taux de perte mesuré était le plus faible permet de réduire ce chiffre à 0,33 %. L'étude montre aussi que la duplication des communications sur deux chemins en simultanée permet une diminution du taux de perte à environ 0,25 %, selon le choix des chemins. Cette étude confirme ainsi l'intérêt d'utiliser différents chemins pour améliorer la fiabilité des communications dans Internet.

Certains travaux[46] ont utilisé une simulation pour mesurer la capacité du routage P2P à rétablir une communication affectée par un incident en fonction de la topologie du réseau overlay et du réseau sous-jacent. Ces travaux confirment que les deux topologies influent sur cette capacité, comme il l'a été évoqué dans la section 2.5.2. L'efficacité du routage P2P pour rétablir les communications est confirmée si la topologie overlay est adaptée à cette tache. Cependant, ces travaux ne discutent pas de la capacité de rétablissement du routage P2P en fonction des différents scénarios d'incident affectant les communications.

Enfin, certains travaux[91] se sont intéressés à l'interaction entre le routage dynamique réalisé sur la couche IP et celui réalisé sur la couche overlay. Ces travaux montrent en particulier que la non-coopération de ces systèmes amenait à un gaspillage des ressources du réseau.

5.3 Modélisation

Dans cette section, nous allons décrire la façon dont nous avons modélisé le comportement d'un grand réseau, de façon à mesurer l'efficacité du routage P2P déployé sur plusieurs noeuds à travers Internet.

5.3.1 Le réseau utilisé pour la simulation

Nous allons tout d'abord décrire comment la topologie du réseau utilisée pour les simulations a été choisie.

Génération de graphe aléatoire

Pour réaliser nos simulations, nous avons utilisé des topologies de réseau générées aléatoirement par l'outil « BRITE » [56]. De manière à utiliser un graphe de réseau similaire à Internet, nous avons généré des graphes composés de plusieurs sous graphes reliés entre eux afin de représenter l'ensemble des domaines ou Autonomous System (AS) qui forment Internet. L'algorithme que nous avons utilisé pour générer les graphes avec BRITE est celui proposé par Barabasi-Albert[4] et régit la manière dont sont reliés les noeuds entre eux à l'intérieur d'un Autonomous System ainsi que la manière dont les Autonomous System sont reliés entre eux.

Nombre de noeuds dans le réseau

Ainsi, l'architecture du réseau de simulation est constituée d'un ensemble d'AS composés eux-même de plusieurs noeuds. Nous avons voulu représenter par ce réseau une interconnexion de réseaux de grande taille. Ainsi, chaque noeud du réseau simulé représente un routeur qui dessert de nombreuses machines utilisateurs.

Nous avons choisi de représenter un réseau à cette échelle, car nous voulons mesurer l'efficacité du routage P2P déployé à travers tout Internet tout en ne complexifiant pas trop la simulation par la présence d'un très grand nombre de noeuds.

Pour nos simulations, nous générerons des graphes comportant 10 AS, comprenant chacun 35 noeuds.

Choix de noeuds overlay

Certains noeuds du réseau seront sélectionnés comme étant des noeuds overlays, c'est-à-dire qu'ils seront des participants au système de routage P2P mesuré dans la simulation. Nous l'avons dit, les noeuds à l'intérieur des AS de notre réseau ne correspondent pas à une machine d'un utilisateur reliée à Internet. Cependant, nous considérerons que si dans notre réseau un noeud est qualifié d'overlay, c'est qu'au moins une machine utilisateur desservie par ce noeud déploie le système de routage P2P.

Le nombre et la position des noeuds overlay dans le réseau varieront selon les simulations.

5.3.2 Connectivité et routage dans le réseau

Nous allons maintenant expliquer comment nous avons modélisé la connectivité entre deux noeuds du réseau, c'est-à-dire la capacité de ces noeuds à communiquer entre eux. Pour cela, nous avons modélisé le comportement des protocoles de routage, au niveau IP et P2P, qui déterminent la façon dont sont acheminées les communications par les noeuds du réseau.

Routage de niveau IP

Au niveau IP, la connectivité entre les noeuds représente la capacité de ces noeuds à communiquer entre eux sans l'utilisation de technique de routage P2P. Il s'agit donc de la manière habituelle d'acheminer les communications dans un réseau. Pour réaliser cela, chaque noeud du réseau maintient une table de routage de niveau IP, qui, pour chaque destination possible, indique le noeud « next-hop » par qui faire transiter une communication pour permettre de joindre cette destination. Nous allons expliquer la technique de routage utilisée dans notre modélisation pour renseigner ces tables de routage.

Le routage à l'intérieur d'un AS (ou routage intra-AS), suit le principe du chemin de moindre coût : Les tables de routage des noeuds appartenant à un même AS sont renseignés de telle sorte que les chemins qui seront empruntés pour relier ces noeuds soient de coûts, modélisés ici par le nombre de noeuds à traverser, minimum à l'intérieur du graphe de l'AS.

Le routage entre les noeuds appartenant à différents AS, s'effectue en deux étapes.

La première est le routage inter-AS externe : il consiste à déterminer, pour chacun des AS, la succession d'AS à emprunter pour joindre un AS différent. Pour cela, nous considérons l'ensemble des AS comme les sommets d'un graphe. Un lien existe entre deux sommets s'il existe un lien dans le réseau qui relie deux noeuds appartenant à chacun des AS représentés par ces deux sommets.

Dans notre modélisation, le routage inter-AS externe suit le principe du chemin de moindre coût dans le graphe des AS : pour joindre un autre AS, un AS traversera le moins d'autres AS possible. Nous appellerons noeud « servant » un AS destination le noeud qui dans un AS est emprunté en dernier pour atteindre un AS destination.

La deuxième étape du routage inter-AS est le routage inter-AS interne. Il consiste, pour chaque noeud de chaque AS, à renseigner les tables de routage vers des noeuds destination appartenant à un AS différent.

Dans notre modélisation, la table de routage d'un noeud est renseignée de telle façon que le chemin entre ce noeud et le noeud de son AS « servant » l'AS du noeud à joindre soit celui de moindre coût.

L'algorithme 1 effectue la vérification de la connectivité de niveau IP entre deux noeuds.

Algorithme 1 estIPconnecté : Algorithme de test de la connectivité par routage IP

ARGUMENTS: Node from, Node dest
SORTIE: from peut communiquer par IP avec dest

```
Node current ← from
tantque current ≠ dest faire
    Node next ← Node «next hop» dans la table de routage IP de current pour aller à dest
    si pas de lien entre current et next alors
        Retourner Faux
    sinon
        current ← next
    finsi
fin tantque
Retourner Vrai
```

Routage de niveau P2P

Nous allons modéliser deux types de routage P2P dans notre simulation. Le premier type de routage est un routage basé sur le calcul du plus court chemin dans une topologie de réseau overlay de type Full Mesh. Ce mode de routage est similaire à ce que réalise un système tel que RON qui a été

présenté dans la section 2.5.3 et sera nommé « routage RON » par la suite. Le second type de routage étudié est similaire à notre système qui a été présenté dans le chapitre précédent et sera nommé « routage SYS » par la suite.

Nous avons décidé d'étudier ces deux types de routage pour permettre de les comparer, et plus particulièrement pour évaluer notre système par rapport au système RON. On l'a vu en effet, le système RON utilise une topologie de type Full Mesh qui entraîne une consommation de ressource réseau très importante, mais qui permet de toujours découvrir un chemin overlay utilisable pour acheminer les communications lorsqu'un tel chemin existe. Dans notre évaluation, le routage RON sera ainsi utilisé comme le mécanisme de routage P2P « idéal », permettant le rétablissement des communications dans toutes les situations où cela est possible. Nous désirons ainsi évaluer si l'architecture utilisée dans notre système, moins coûteuse que celle de RON pour son fonctionnement, permet toutefois le rétablissement des communications dans un grand nombre de situations.

Routage RON

Dans le routage RON, un lien overlay existe entre chaque noeud overlay. De plus, chaque noeud overlay maintient une table de routage P2P. Pour chaque noeud overlay de destination, la table de routage P2P d'un noeud overlay indique le noeud overlay à qui transmettre une communication de manière à ce que le chemin emprunté par celle-ci soit le chemin de moindre coût dans le réseau overlay. Dans notre modélisation, le coût d'un chemin overlay est égal au nombre de noeuds IP qu'il traverse.

L'algorithme 2 effectue la vérification de la connectivité de niveau P2P pour le mécanisme de routage P2P de type RON. Cet algorithme vérifie la possibilité pour deux noeuds de communiquer entre eux grâce à l'utilisation de ce mécanisme.

Algorithme 2 estRONconnecté : Algorithme de test de la connectivité par routage P2P RON

ARGUMENTS: RONNode from, RONNode dest
SORTIE: from peut communiquer par RON avec dest

```
Node current ← from
tantque current ≠ dest faire
    Node next ← Node «next hop» dans la table de routage RON de current pour aller à dest
    si estIPConnecté(current,next) alors
        current ← next
    sinon
        Retourner Faux
    finsi
fin tantque
Retourner Vrai
```

Routage SYS

Dans le routage SYS, les noeuds n'utilisent pas de table de routage pour acheminer les communications. À la place, les noeuds maintiennent un chemin overlay à utiliser pour le joindre chaque noeud overlay de destination. Ce chemin est choisi comme étant :

– Le chemin « direct », qui va du noeud source au noeud destination en empruntant la route de niveau IP telle que calculée par le routage IP, si ce chemin est utilisable

– Un chemin overlay qui transite par un unique noeud overlay tiers parmi k noeuds possible, sinon.

Pour chaque noeud de destination, il existe une liste des k noeuds tiers possibles distincte. Les noeuds de ces listes sont choisis aléatoirement parmi les noeuds overlays du réseau distincts du noeud overlay source et du noeud overlay destination. Ces listes sont déterminées une unique fois à l'initialisation du mécanisme et ne sont pas modifiées ensuite.

Si le chemin overlay « direct » n'est pas utilisable, le chemin overlay qui sera utilisé parmi les k chemins possible sera celui de moindre coût, en terme de nombre de noeuds IP traversés.

L'algorithme 3 effectue la vérification de la connectivité de niveau overlay pour le mécanisme de routage P2P SYS.

Algorithme 3 estSYSconnecté : Algorithme de test de la connectivité par routage P2P SYS

ARGUMENTS: SYSNode from, SYSNode dest
SORTIE: from peut communiquer par SYS avec dest

```
Chemin p ← chemin utilisé par from pour aller à dest
si p est un chemin direct alors
    si estIPconnecté(from,dest) alors
        Retourner Vrai
    sinon
        Retourner Faux
    finsi
sinon
    Node o ← noeud overlay de transit de p
    si estIPconnecté(from,o) ET estIPconnecté(o,dest) alors
        Retourner Vrai
    sinon
        Retourner Faux
    finsi
finsi
```

Rétablissement réseau après un incident

Nous verrons dans la suite que l'apparition d'un incident dans notre réseau sera modélisée par la suppression d'un ou plusieurs liens ou noeuds de notre réseau. Pour rétablir la délivrance d'une communication qui empruntait ces liens ou noeuds, il est nécessaire de modifier la manière dont elle est acheminée. Dans notre modélisation, c'est le mécanisme de routage qui est chargé de cette tâche.

Pour cela, les principes de routage du réseau modélisé, exposés dans le chapitre précédent, sont appliqués à l'identique, mais dans le réseau affecté par un incident, c'est-à-dire avec un ou plusieurs noeuds ou liens en moins.

Dans la suite de ce chapitre, nous utiliserons les notations « RON », « SYS », « IGP » ou « EGP » pour indiquer qu'un mécanisme de rétablissement a été déclenché. Ainsi, le rétablissement RON correspond à la mise à jour des tables de routage P2P des noeuds overlay déployant le mécanisme de type RON, en accord avec la nouvelle topologie du réseau endommagé. Le rétablissement SYS correspond à la reconfiguration des chemins overlays à utiliser par les noeuds qui déploient le mécanisme de type SYS, dans le réseau endommagé. Le rétablissement IGP correspond à la mise à jour des tables de routage IP des noeuds du réseau suite à la reconfiguration des protocoles de routage intra-AS et inter-

AS interne, dans le réseau endommagé. Enfin, le rétablissement EGP correspond à la mise à jour des tables de routage IP des noeuds suite à la reconfiguration des protocoles de routage intra-AS, inter-AS interne et inter-AS externe dans le réseau endommagé. Ainsi, le rétablissement IGP correspond au rétablissement réseau interne aux différents AS tandis que le rétablissement EGP correspond au rétablissement à l'intérieur et entre les différents AS.

5.3.3 Scénarios d'incident

Nous allons maintenant présenter les différents scénarios d'incidents utilisés lors de nos simulations.

Panne simple de lien intra-AS

Ce scénario, noté « IIntraAS », modélise un incident affectant un lien interne au réseau d'un opérateur. Dans ce scénario, un lien interne à un AS, qui relie deux noeuds membres du même AS, est supprimé. Ce lien est choisi aléatoirement.

Panne simple d'un lien inter-AS

Ce scénario, noté « IInterAS », modélise un incident affectant un lien reliant deux réseaux d'opérateurs. Dans ce scénario, un lien inter-AS, qui relie deux noeuds membres de deux AS différents, est supprimé. Ce lien est choisi aléatoirement.

Panne double

Ce scénario, noté « IDouble », modélise un incident ayant affectant deux liens du réseau. Ce type d'incident peut se produire lorsque plusieurs équipements physiques sont affectés simultanément, mais aussi lorsqu'un seul équipement réseau, utilisé pour le transport de plusieurs liens de niveau IP, est affecté. Dans ce scénario, deux liens intra-AS, choisis aléatoirement, sont supprimés.

Panne d'un routeur

Ce scénario, noté « IRouter », modélise un incident affectant un routeur. Dans ce scénario, un noeud et tous les liens qui y sont raccordés sont supprimés.

Panne catastrophique

Ce scénario, noté « ICatas », modélise un incident exceptionnel et de grande envergure, telle qu'une catastrophe naturelle, affectant le réseau. Dans ce scénario, trois AS sont choisis aléatoirement, avec pour contrainte que chacun ait au moins un lien qui le relie à l'un des autres. À l'intérieur de ces AS, un tiers des noeuds, choisis aléatoirement, ainsi que tous les liens qui y sont reliés, sont supprimés.

Attaque DDoS

Ce scénario, noté « IDDoS », modélise une attaque par déni de service distribué par saturation des équipements réseaux. Pour cela, un noeud cible et 5 noeuds attaquants sont choisis aléatoirement

parmi les noeuds du réseau. On considère ensuite les chemins du réseau utilisés par les noeuds attaquant pour joindre le noeud cible. Chaque lien du réseau utilisé par au moins deux de ces chemins est supprimé.

Bien que lors d'une attaque DDoS, les liens réseau ne sont que surchargés, mais pas complètement inutilisables, comme le modélisent les suppressions effectuées pour ce scénario, nous pensons que cette modélisation nous permettra de mesurer la capacité des systèmes de routage P2P à contourner les liens affectés par une attaque et à proposer un chemin de bonne qualité pour une communication.

5.4 Résultat des simulations

Nous allons maintenant présenter les résultats de nos simulations. Pour les réaliser, nous avons développé notre propre logiciel. Nous avons utilisé la bibliothèque JGrapht[64] pour gérer les graphes modélisant le réseau ainsi que les calculs de plus courts chemins à l'intérieur de ceux-ci.

5.4.1 Critères de performance

Ces travaux visent avant tout à mesurer la capacité du routage P2P à rétablir la connectivité entre les noeuds qui utilisent cette technique dans un grand réseau comme Internet, lorsque ce dernier est affecté par un incident. Par conséquent, le critère, tel qu'évoqué dans la section 2.4.1 sur les critères de performance des mécanismes de rétablissement, concerné par ces travaux est la portée du mécanisme. Ainsi, nous ne nous intéressons pas ici au temps nécessaire pour rétablir la connectivité, ni au coût nécessaire à l'utilisation du routage P2P par exemple.

Par conséquent, le critère de performance principal étudié dans nos simulations est le taux de rétablissement d'un mécanisme. Ce taux représente le ratio entre le nombre de communications qui ont pu être rétablies après qu'elles aient été affectées par un incident par rapport au nombre de communications affectées par cet incident. Nous comparerons les taux obtenus avec les mécanismes P2P à ceux obtenus avec les protocoles de routage IP.

De plus, nous étudierons plus succinctement la qualité des chemins alternatifs proposés par les mécanismes de rétablissement. Pour cela, nous comparerons la longueur de ces chemins par rapport à la longueur des chemins primaires originellement utilisés.

5.4.2 Taux de rétablissement du routage P2P

Dans cette section, nous allons présenter les résultats de simulations obtenus qui concernent le taux de rétablissement des mécanismes de rétablissement P2P, comparés à ceux obtenus par le routage IP. Ainsi, pour chaque pair de noeud overlay dont les communications entre eux sont affectées par un incident, nous avons mesuré la capacité pour ces noeuds à communiquer, grâce au routage IP ou P2P, une fois que le processus de rétablissement des différents systèmes étudiés était achevé.

La figure 5.2 montre le taux de rétablissement, pour les rétablissements de type RON, SYS, IGP et EGP, en fonction des différents scénarios d'incidents. Pour ces résultats, nous avons utilisé 20 noeuds overlays déployés aléatoirement dans le réseau. Le nombre k de chemins alternatifs potentiels pour le mécanisme de type SYS a été fixé à 8.

Nous pouvons remarquer que pour les scénarios IInterAS, IDouble, IRouter and ICatas, les taux de rétablissement des mécanismes RON et SYS sont meilleurs que celui du rétablissement IGP, et légèrement inférieur à celui du rétablissement EGP. Pour le scénario IIntraAS, le taux de rétablissement IGP est identique au taux EGP et meilleur que ceux de RON et SYS. Pour le scénario IDDoS, les taux de rétablissement RON et SYS sont légèrement inférieurs aux taux de rétablissement IGP, tous deux

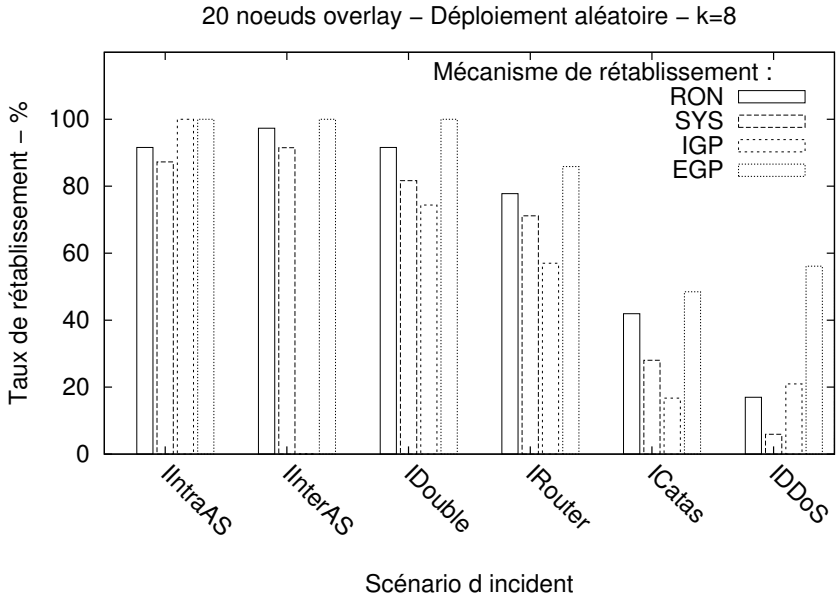


FIG. 5.2: Taux de rétablissement des mécanismes, pour les différents scénarios

largement inférieurs au taux de rétablissement EGP. Dans tous les scénarios, le taux de rétablissement du mécanisme SYS est inférieur à celui de RON. Il est légèrement inférieur pour les scénarios dont le nombre de liens affectés par l'incident est faible (IIntra, IInter, IDouble, IRouter) et plus fortement pour les scénarios dont le nombre de liens affectés par l'incident est important (ICatas, IDDoS).

Les résultats des taux de rétablissement P2P par rapport aux taux IP étaient prévisibles. En effet, le rétablissement effectué par le routage IP permettra toujours de rétablir une communication affectée par un incident si un chemin alternatif existe dans le réseau. Par conséquent, le rétablissement EGP, qui correspond à la reconfiguration complète de l'ensemble des tables de routage IP des noeuds du réseau, rétablira toujours l'ensemble des communications possibles et sera par conséquent meilleur que le rétablissement P2P quelque il soit.

Pour le scénario IIntraAS, le routage IGP est suffisant pour rétablir les communications affectées par l'incident puisque ce dernier est interne aux AS. Cependant, nous pensons que les résultats du rétablissement P2P sont bons. En effet, même avec un petit nombre de noeuds overlay dans le réseau, le taux de rétablissement de ces mécanismes est presque aussi haut que pour le rétablissement EGP, à l'exception du scénario IDDoS, où de trop nombreux noeuds sont affectés par l'incident. De plus, le taux de rétablissement P2P est souvent meilleur que celui du rétablissement IGP. Les taux de rétablissement mesurés ici nous font considérer que le routage P2P est une alternative viable au routage IP pour le rétablissement des communications.

Nous constatons de plus que le taux de rétablissement P2P de type SYS est inférieur à celui de type RON. Ce résultat était attendu. En effet, dans le routage de type RON, l'ensemble des chemins overlays possibles peut être utilisé pour acheminer une communication. Ce n'est pas le cas du mécanisme de type SYS, où seul k (fixé ici à 8) chemins alternatifs, choisis préalablement à l'incident, peuvent être utilisés pour acheminer une communication affectée par un incident. De plus, ces chemins ne peuvent transiter que par un seul noeud overlay, à la différence des différents chemins alternatifs utilisés par le mécanisme de type RON. Il est par conséquent possible que les k chemins alternatifs utilisables par le mécanisme de type SYS soient tous affectés par un incident lorsqu'il se déclare, alors

que d'autres chemins overlays auraient pu être utilisés. C'est ce que fait le mécanisme de type RON et c'est ce qui explique l'écart mesuré entre les taux de rétablissement de ces deux mécanismes.

Cependant, les mécanismes de type RON qui permettent d'utiliser l'ensemble des chemins overlays possibles sont en réalité coûteux à utiliser, comme nous l'avons déjà dit dans la section 2.5.3. Ainsi, il faut considérer les résultats des taux de rétablissement du mécanisme de type RON comme des résultats optimaux pour un mécanisme de routage P2P. De tels résultats ne peuvent être obtenus sans un cout d'utilisation important, tandis que le coût d'utilisation d'un mécanisme de type SYS est beaucoup plus faible, comme nous l'avons décrit dans la section 4.4.6. Par conséquent, nous considérons que les taux de rétablissement du mécanisme de type SYS sont bons, étant donné qu'ils ne sont que légèrement inférieurs à ceux du rétablissement RON, pour les scénarios IIntra, IInter, IDouble, IRouter.

5.4.3 Nombre de sauts overlays des chemins overlays

Nous allons étudier ici le nombre de sauts overlays d'un chemin overlay utilisé pour rétablir une communication affectée par un incident. Le nombre de sauts overlays d'un chemin overlay est le nombre de noeuds overlays traversés par ce chemin. Ce paramètre est important dans la capacité des mécanismes P2P à rétablir une communication, puisque nous avons limité le mécanisme SYS à l'utilisation de chemin overlay comprenant un saut overlay, alors que le mécanisme RON peut utiliser des chemins overlays comprenant un nombre quelconque de sauts overlays. Nous proposons ici d'étudier dans quelle mesure cette limitation influe sur la capacité du mécanisme SYS à rétablir une communication affectée par un incident.

La figure 5.3 présente le pourcentage de communications qui, lorsqu'elles sont affectées par un incident, nécessitent de traverser au moins 2 noeuds overlays intermédiaires pour pouvoir être acheminées par un chemin overlay, parmi l'ensemble des communications qui peuvent être rétablies par un mécanisme de routage P2P. Les communications ainsi concernées sont par conséquent rétablies par le mécanisme RON, mais pas par le mécanisme SYS, car ses chemins overlays ne peuvent traverser plus d'un noeud overlay intermédiaire. Ces résultats sont présentés en fonction du scénario d'incident étudié et ont été réalisés sur un réseau overlay de 20 noeuds déployés aléatoirement. Le paramètre k du nombre de chemins de secours potentiels du mécanisme SYS a été fixé à 8.

On observe que pour les scénarios IIntra, IInter, IDouble et IRouter, le pourcentage de communications devant être rétablies par des chemins overlays de plus de 2 sauts overlays est faible, inférieur à 2,5 %. Par contre, pour les scénarios ICatas et IDDoS, ce taux est plus important, respectivement de 18 et 38 %.

Lors d'incidents de grande envergure, représentés par les scénarios ICatas et IDDoS, qui impactent de nombreux liens du réseau, il est normal qu'il soit parfois nécessaire de faire transiter les communications par plus de deux noeuds overlays pour les rétablir. En effet, la présence de liens affectés en plusieurs endroits du réseau augmente le risque de ne pas trouver de chemin overlay transitant par un noeud overlay unique qui soit utilisable. La figure 5.4 illustre ce cas de figure.

5.4.4 Réseau overlay et performances

Dans cette section, nous allons présenter les effets des différents déploiements des noeuds overlays sur le taux de succès de rétablissement. Nous étudierons ici le mécanisme de routage overlay de type RON, qui, nous l'avons dit, présente les taux de rétablissement optimaux pouvant être obtenus avec le routage P2P. Nous reviendrons sur la mesure des différences entre les mécanismes de type SYS et RON dans la partie suivante.

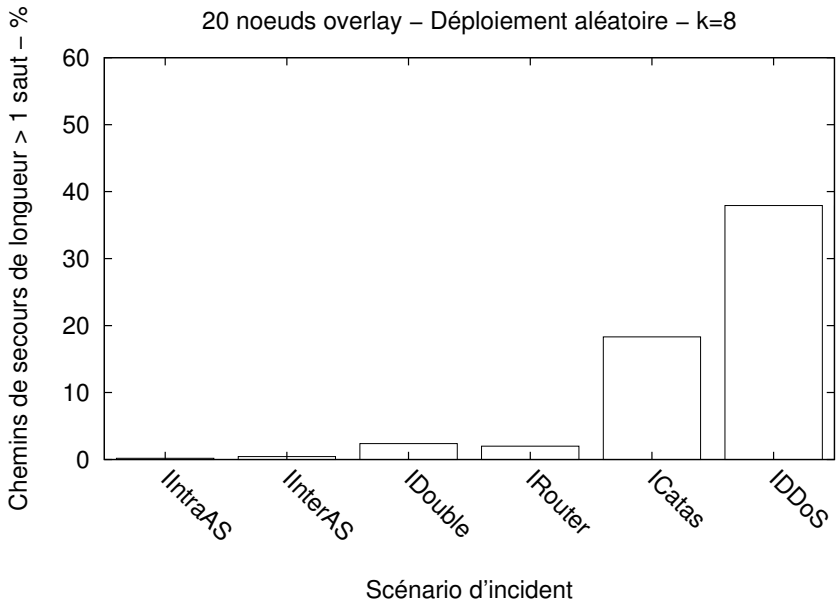


FIG. 5.3: Pourcentage des chemins de secours dont le nombre de sauts overlays est supérieur à 1, pour chaque scénario

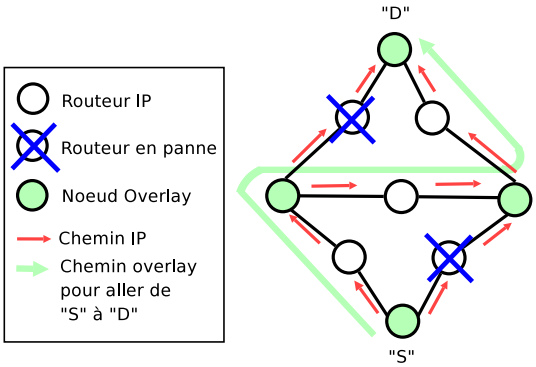


FIG. 5.4: Exemple de situation nécessitant un chemin de secours à 2 sauts overlays

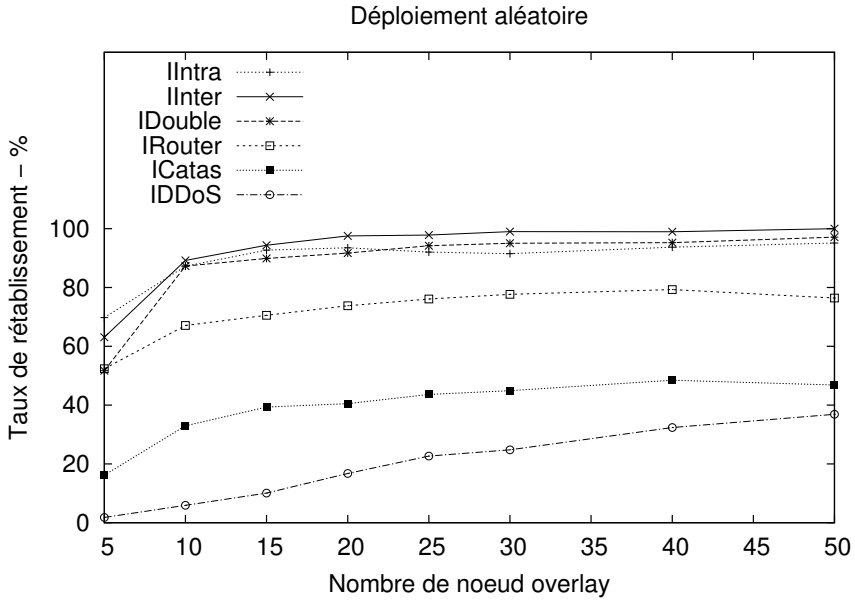


FIG. 5.5: Taux de rétablissement du mécanisme RON en fonction du nombre de noeuds overlay

Nous allons étudier ici l'influence de deux paramètres : le nombre de noeuds overlays et la localisation de ces noeuds dans le réseau IP. Nous avons en effet étudié deux façons différentes de déployer les noeuds overlays : le déploiement « aléatoire », qui sélectionne aléatoirement les noeuds overlays parmi les noeuds IP du réseau et qui avait été utilisé pour les résultats présentés plus haut et le déploiement « réparti », qui choisit les noeuds overlays de façon à ce qu'ils soient répartis parmi les différents AS. Ainsi, le nombre de noeuds overlay par AS est le même, à 1 noeud près, dans tous les AS. À l'intérieur d'un AS, le noeud overlay est choisi aléatoirement parmi les noeuds IP.

La figure 5.5 montre le taux de rétablissement RON, pour chaque scénario d'incident, en fonction du nombre de noeuds overlays dans le réseau. Le déploiement aléatoire a été utilisé. On constate que pour les scénarios InterAS, IDouble, IRouter et ICatas, le taux de rétablissement augmente rapidement jusqu'a l'utilisation de 10 noeuds overlay, puis croit lentement jusqu'à l'utilisation de 20 noeuds, pour ensuite rester approximativement constant si plus de noeuds sont utilisés, sauf pour le scénario ICatas, où 40 noeuds peuvent être utilisé pour obtenir de meilleurs résultats. Ce n'est pas le cas pour le scénario IDDoS où le rétablissement bénéficie de la présence d'un plus grand nombre de noeuds.

Ces résultats montrent que 10 ou 20 noeuds overlays suffisent à obtenir un bon taux de rétablissement, tout du moins pour les scénarios où le nombre de liens affectés n'est pas trop important. Dans ce cas, les bénéfices lors de l'utilisation d'un plus grand nombre de noeuds ne sont pas importants. Cependant, ceci n'est pas vrai pour les incidents de plus grande envergure.

La figure 5.6 montre le taux de rétablissement RON, pour le déploiement aléatoire et pour le déploiement réparti, en fonction du scénario d'incident. Nous avons utilisé 10 noeuds overlays dans ce scénario pour mettre en évidence la façon dont le déploiement réparti peut améliorer le taux de rétablissement. On remarque que pour les scénarios InterAS, IRouter, ICatas et IDDoS, le taux de rétablissement du routage P2P est significativement amélioré par le déploiement réparti en comparaison du déploiement aléatoire. Les taux de rétablissement des deux déploiements sont approximativement identiques pour les scénarios IIntraAS et IDouble.

Ces résultats montrent que la répartition des noeuds overlays à travers l'ensemble du réseau est

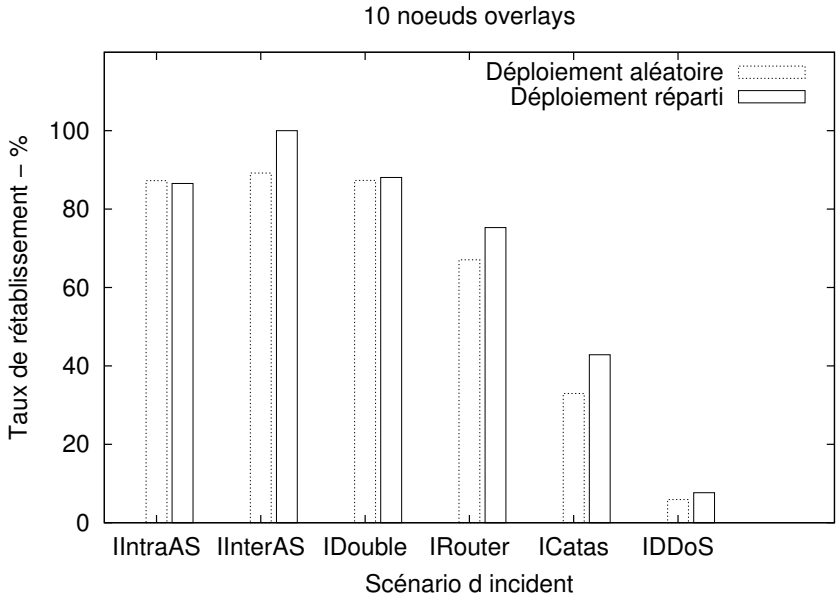


FIG. 5.6: Taux de rétablissement du mécanisme RON en fonction du type de déploiement

bénéfique au taux de rétablissement des mécanismes de routage P2P, en particulier lorsqu'un incident affecte plusieurs domaines ou un lien inter-domaine. En effet, lorsque le déploiement des noeuds overlay est réparti, la probabilité qu'un incident affecte simultanément plusieurs chemins entre ces noeuds est amoindrie puisque ces chemins sont plus distribués à travers le réseau. Par conséquent, il est plus aisé de trouver un ou plusieurs chemins IP utilisables pour former le chemin overlay. C'est pourquoi le taux de rétablissement s'en trouve amélioré.

5.4.5 Spécificités du mécanisme SYS

Dans cette section, nous allons étudier le taux de succès de rétablissement de type SYS comparé à celui de type RON, et en particulier comment le paramètre k du nombre de chemins alternatifs potentiels à utiliser par ce mécanisme influence ce taux. De plus, nous étudierons les effets du déploiement réparti sur l'efficacité de ce mécanisme.

La figure 5.7 montre le rapport entre le taux de rétablissement obtenu par le rétablissement SYS avec celui obtenu par le rétablissement RON, pour les différents scénarios, en fonction du paramètre k du nombre chemins de secours potentiels du mécanisme SYS. Ces mesures ont été réalisées dans un réseau comportant 30 noeuds overlays. Le rapport entre le taux de rétablissement du mécanisme de type SYS et celui du mécanisme de type RON indique la capacité du mécanisme SYS à rétablir les communications affectées par un incident par rapport à un mécanisme de routage P2P idéal, représenté par le mécanisme RON.

On observe que pour les scénarios IIntra, IInter, IDouble, IRouter et ICatas, le rapport des taux de rétablissement croît rapidement lorsque k est inférieur à 8. Ce rapport s'accroit ensuite moins rapidement lorsque k est compris entre 8 et 16. Ensuite, ce rapport stagne, sauf pour les scénarios ICatas et IDDoS pour lesquels le taux de rétablissement du mécanisme SYS peut être amélioré par l'utilisation d'une valeur de k plus grande.

On constate qu'il n'est pas nécessaire d'utiliser un grand nombre k de chemins de secours poten-

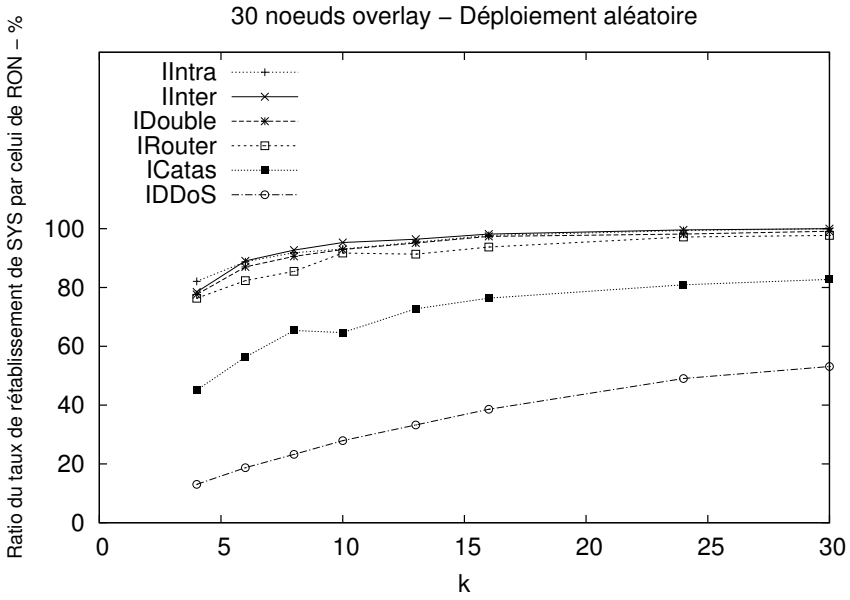


FIG. 5.7: Taux de rétablissement du mécanisme SYS en fonction du nombre *k* de chemins de secours potentiels

tiels pour obtenir un taux de rétablissement satisfaisant pour le mécanisme SYS, en comparaison du mécanisme RON. Cependant, ceci n'est pas vrai pour les scénarios ICatas et IDDoS, dont les incidents sont de grandes étendues.

La figure 5.8 représente le taux de rétablissement SYS en fonction du nombre *k* de chemins de secours potentiels, selon le déploiement aléatoire ou réparti des noeuds, ainsi que le taux de rétablissement du mécanisme RON. Cette mesure a été réalisée avec l'utilisation de 10 noeuds overlays et le scénario d'incident IRouter. On observe qu'avec le déploiement aléatoire des noeuds overlays, le taux de rétablissement du mécanisme SYS est inférieur à celui de RON, pour *k* inférieur à 8. Lorsque *k* est supérieur à 8, le taux de rétablissement des deux mécanismes est équivalent. Lorsque le déploiement réparti est utilisé, le taux de rétablissement du mécanisme SYS est supérieur à celui de RON avec le déploiement aléatoire. De plus, dès que *k* est supérieur à 6, l'écart entre les taux de rétablissement des mécanismes SYS et RON est inférieur à 4 %.

Ces résultats confirment que le déploiement réparti apporte une sensible amélioration des performances des mécanismes P2P. En particulier, SYS dans ce cas se comporte mieux que le mécanisme RON dont les noeuds overlays sont répartis aléatoirement. De plus, si le nombre de chemins de secours potentiels *k* est faible, les performances de ce mécanisme seront plus proches de celle de RON si le déploiement est réparti que si celui-ci est aléatoire.

Nous l'avons vu, le déploiement réparti permet une amélioration notable des performances des mécanismes P2P, en particulier lorsque le nombre de noeuds overlays ou lorsque le nombre *k* de chemins de secours potentiels sont faibles. Toutefois, ce déploiement n'est pas toujours possible, en particulier lorsque le choix des noeuds participants au système de routage P2P n'est pas décidé de manière centralisée. Il est cependant possible d'envisager l'amélioration du taux de succès de rétablissement du mécanisme SYS lors de la sélection des noeuds de relai des chemins de secours potentiels. En effet, si la sélection des noeuds s'effectue de façon répartie dans le réseau, c'est-à-dire en choisissant des noeuds d'AS différents, on peut s'attendre à observer les mêmes améliorations

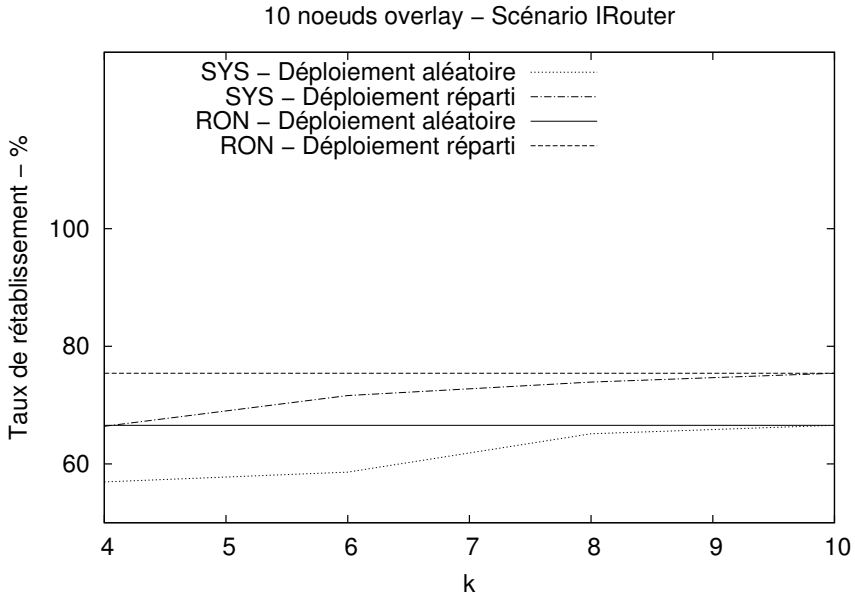


FIG. 5.8: Influence du choix des chemins de secours potentiels sur le taux de rétablissement du mécanisme SYS

du taux de rétablissement qu'avec le déploiement réparti des noeuds overlays. Il est ainsi possible d'apporter une amélioration simple au mécanisme SYS, en essayant de choisir des noeuds de relai appartenant à des AS différents.

5.4.6 Qualité du chemin alternatif

Dans cette section, nous allons étudier la qualité des chemins alternatifs calculés par les mécanismes de rétablissement. Nous allons donc comparer la qualité des chemins alternatifs proposés par les mécanismes RON, SYS et IP. Pour estimer la qualité de ces chemins, nous mesurerons leurs longueurs, représentées ici par le nombre de noeuds IP, ou routeurs, traversés par ceux-ci. Plus court est un chemin et meilleure est sa qualité. Bien que nous ne prétendons pas représenter exactement la qualité d'un chemin, qui dépend de nombreux facteurs, nous pensons que le nombre de routeurs traversés est un de ces facteurs et mérite que l'on s'y intéresse.

La figure 5.9 représente la fonction de répartition de la pénalité des chemins alternatifs, pour le routage IP et P2P de type SYS et RON, pour les déploiements de types aléatoires et répartis. 20 noeuds overlays ont été utilisés lors de ces simulations. La pénalité d'un chemin alternatif est le rapport, exprimé ici en pourcentage, entre la longueur du chemin alternatif utilisé par une communication et celle du chemin IP primaire, initialement utilisé par cette communication avant l'apparition de l'incident. Nous observons que la qualité des chemins alternatifs des mécanismes P2P n'est pas aussi bonne que celle obtenue par le routage IP, et que les chemins alternatifs du mécanisme de type RON sont de meilleures qualités que ceux du mécanisme SYS. En effet, 80 % des chemins IP alternatifs ont une pénalité de 125 %, tandis que 80 % des chemins alternatifs RON ont une pénalité de 170 % et elle est de 195 % pour le mécanisme SYS. Le déploiement des noeuds overlays, aléatoire ou réparti, n'a pas d'impact significatif sur la pénalité des chemins overlays alternatifs.

Ces résultats montrent que les chemins overlays sont le plus souvent plus longs, en terme de

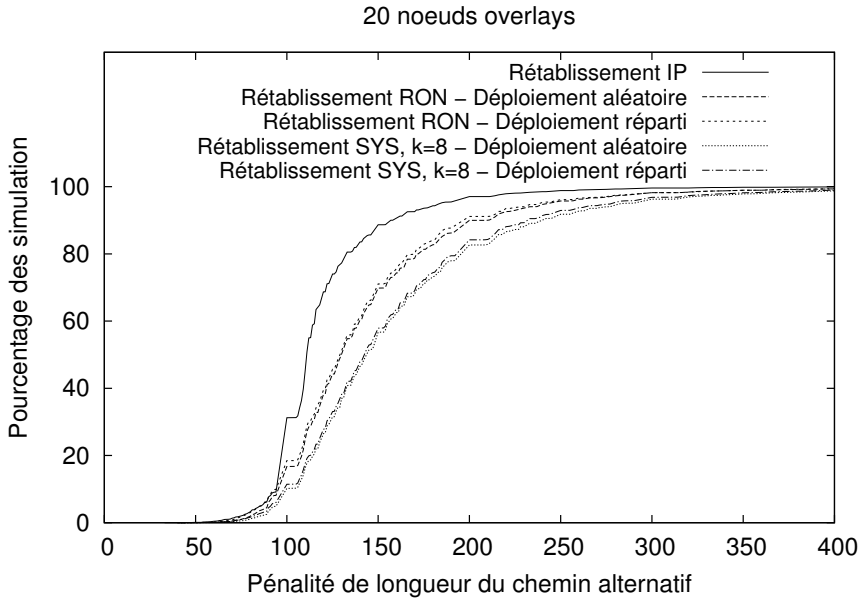


FIG. 5.9: Pénalité du chemin de secours des différents mécanismes

nombre de noeuds traversés, que les chemins IP. Ce résultat était attendu, car le routage IP essaye de minimiser cette longueur, et par conséquent, après un incident, c'est le nouveau plus court chemin qui est utilisé comme chemin alternatif. Avec les mécanismes P2P c'est le plus court chemin entre les noeuds overlays qui est utilisé et par conséquent, moins de chemins alternatifs sont utilisables qu'avec le routage IP.

La figure 5.10 représente la pénalité moyenne des chemins alternatifs, pour le routage P2P de type RON et SYS et le routage IP en fonction du nombre de noeuds overlay utilisé pour le routage RON ou du nombre *k* de chemins de secours potentiels pour le routage SYS. On observe que les pénalités des mécanismes de type RON et SYS diminuent rapidement lorsque le nombre de noeuds overlay ou *k* augmente jusqu'à 20. Ces pénalités diminuent alors plus lentement pour se rapprocher des pénalités mesurées pour le routage IP.

On observe aussi que les pénalités du mécanisme SYS sont légèrement inférieures à celle de RON lorsque le nombre de noeuds overlays utilisés avec le mécanisme RON et le nombre *k* utilisé par le mécanisme SYS sont équivalents et inférieurs à 20. Lorsque ces valeurs sont supérieures à 20, les pénalités observées se confondent.

Ces résultats montrent tout d'abord que les bénéfices à utiliser plus de 20 noeuds overlays ou une valeur de *k* supérieure à 20 sont faibles en ce qui concerne la pénalité des chemins alternatifs. De plus, ces résultats montrent le meilleur comportement du mécanisme SYS lorsque le nombre de noeuds overlay ou le nombre *k* de chemins de secours potentiels pour le mécanisme SYS sont faibles, à valeur équivalente.

5.4.7 Résumé des résultats et discussion

Voici les résultats les plus importants que nous avons observés dans ce chapitre :

– Lors de l'apparition d'un incident, les taux de réussite des mécanismes de rétablissement P2P sont systématiquement meilleurs que ceux des mécanismes de routage IP IGP (reconfiguration

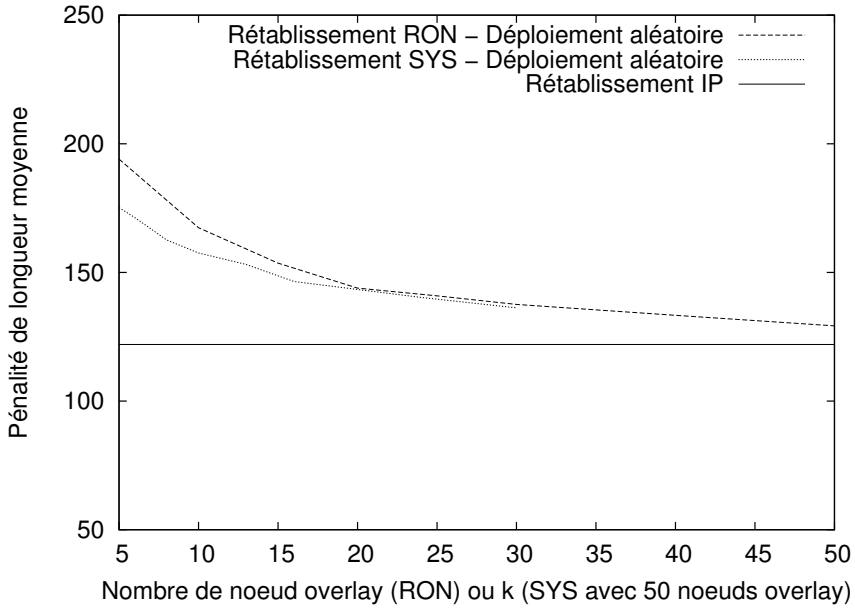


FIG. 5.10: Pénalité des chemins alternatifs des mécanismes de routage P2P en fonction de *k* et du nombre de noeuds overlay

interne aux AS).

- Lors de l'apparition d'un incident de faible envergure (scénario IIntra, IInter, IDouble, IRouter), les taux de réussite des mécanismes de rétablissement P2P sont proches de ceux du routage IP IGP et EGP (reconfiguration interne et entre les AS), mais l'écart est plus important lors d'incident de grande envergure (ICatas et IDDos).
- Lors d'incident de faible envergure, le taux de réussite du mécanisme de type SYS est proche de celui de RON, mais l'écart est plus important lors d'incident de grande envergure.
- La probabilité qu'une communication ait besoin d'un chemin overlay de plus de 2 sauts overlays pour être rétablie est négligeable lors d'incident de faible envergure, mais importante lors d'incident de grande envergure.
- L'utilisation de plus de 20 noeuds overlay n'apporte pas une amélioration importante du taux de réussite de rétablissement ou de la pénalité sur la longueur des chemins alternatifs des mécanismes P2P. Si le déploiement des noeuds overlays est réparti dans le réseau, une dizaine de noeuds seulement peuvent être utilisés pour obtenir les mêmes performances.
- À partir d'une valeur de k=8 chemins de secours potentiels à utiliser par le mécanisme SYS, on constate un taux de réussite de rétablissement satisfaisant. Ce taux est encore meilleur si les noeuds par lesquels transitent les chemins de secours potentiels sont choisis de manière répartie dans le réseau.

Nous allons discuter de la validité des résultats obtenus dans ce chapitre. Rappelons tout d'abord que les réseaux modélisés ici peuvent être considérés comme les réseaux de coeur, destinés à l'acheminement des communications d'un grand nombre d'utilisateurs à travers Internet. Les incidents représentés dans notre modèle n'affectaient donc que cette partie du réseau et les incidents plus « proches » de l'utilisateur n'ont pas été étudiés. Ces incidents, correspondant approximativement à un tiers des incidents affectant les communications de bout en bout des utilisateurs (voir la section 2.3.1 à ce sujet), sont situés dans des parties « moins connectés » du réseau, telles que les points uniques de

panne. Il par conséquent plus difficile, voir impossible, pour les mécanismes de rétablissement de les contourner.

Ainsi, la portée réelle des mécanismes de rétablissement basé sur le routage P2P, telle que perçue par l'utilisateur, est inférieure à celle présentée dans ce chapitre. Différents travaux ont étudié la portée des mécanismes de routage P2P (voir la section 5.2.3) et ont mesuré des taux de rétablissement compris entre un tiers et deux tiers. Notre étude confirme cependant que la portée des mécanismes de routage P2P, y compris celle de notre système, est proche de celle des mécanismes de routage IP.

5.5 Conclusions sur la portée du routage P2P

Dans ce chapitre, nous avons étudié la portée des systèmes de routage P2P, c'est-à-dire la capacité de ces mécanismes à rétablir les communications lorsqu'un incident frappe un réseau tel qu'Internet. En fonction de la situation lors de l'incident, ces mécanismes ne permettent pas systématiquement de rétablir une communication, alors qu'un protocole de routage IP en aurait été capable. En effet, la nature du routage P2P ne permet pas d'exploiter l'ensemble des chemins IP existants dans le réseau, mais uniquement les chemins IP connectant les participants à ce réseau overlay. Il nous est par conséquent paru essentiel de mesurer la portée des systèmes de routage P2P afin de réellement évaluer le gain de fiabilité pouvant être apporté aux communications des utilisateurs.

Plus particulièrement, nous avons évalué notre mécanisme de routage P2P proposé dans le chapitre 4, en le comparant au mécanisme de routage P2P le plus connu, RON, mais aussi aux protocoles de routage IP de type IGP et EGP. Nous avons ainsi étudié la portée de ce mécanisme, mais aussi la qualité des chemins alternatifs proposés lors d'un rétablissement. Nous avons aussi étudié l'influence de certains paramètres de ce mécanisme sur ces performances.

Pour cela, nous avons proposé une modélisation du comportement de ces différents mécanismes dans un grand réseau composé de sous réseaux interconnectés, ainsi que plusieurs scénarios d'incidents affectant les communications de ce réseau. Nos résultats de simulation ont montré que les taux de rétablissement des mécanismes P2P peuvent être presque aussi bons que ceux observés pour le routage IP. Nous avons de plus montré comment le nombre de noeuds overlay et la localisation de ces noeuds influaient sur la portée des mécanismes ainsi que sur la qualité des chemins alternatifs. Ainsi, l'utilisation de noeuds overlays répartis dans tout le réseau apporte un gain important de performances, même lorsqu'un petit nombre de noeuds sont utilisés.

Nous avons de plus montré que la portée de notre système s'approche de celle de RON, qui possède la portée maximale pour un réseau overlay donné. Ainsi, alors que notre système accorde un temps de rétablissement plus court et un coût de fonctionnement contenu comparé à ce que permet RON, sa portée n'est que légèrement moins bonne, voire identique, si suffisamment de noeuds sont utilisés, en particulier lors d'incident de faible envergure. Nous avons de plus montré que l'utilisation d'une valeur k=8 chemins de secours potentiels suffisait à apporter de bons résultats, et qu'il n'était pas nécessaire d'utiliser plus de 20 noeuds dans le réseau pour atteindre des performances optimales. Nous avons enfin souligné l'importance de répartir les chemins de secours potentiels dans le réseau.

Par conséquent, les utilisateurs qui désirent fiabiliser leurs communications peuvent utiliser les systèmes de routage P2P, et en particulier notre système décrit dans le chapitre 4, en tant que protection additionnelle au routage IP. En effet, ces mécanismes permettent le rétablissement de communication lorsque celles-ci sont affectées par des incidents qui sont difficilement gérés par les mécanismes de routage IP classiques déployés par les opérateurs de réseaux, comme les incidents affectant plusieurs AS, ou encore lorsque les mécanismes de rétablissement déployés par les opérateurs ne permettent pas de satisfaire les besoins de fiabilité des utilisateurs.

Ces travaux ont permis de mettre en évidence l'intérêt des mécanismes de rétablissement réseau basé sur le routage P2P lors de l'apparition d'incidents. Néanmoins, nous avons remarqué que face aux incidents de grande envergure, tels que les attaques par déni de service distribué, ces mécanismes avaient plus de difficulté à rétablir une communication. Par conséquent, il serait profitable de proposer des solutions pour adapter ces mécanismes à ce type d'incident. De plus, tout au long de cette étude, nous avons mis en évidence l'importance de la position des noeuds et chemins overlay par rapport à la topologie du réseau IP pour l'efficacité des mécanismes. Malheureusement, la découverte de cette topologie est aujourd'hui techniquement difficile pour les noeuds overlays, qui sont souvent de simples machines utilisateurs. Une évolution de cette situation permet d'envisager d'importantes améliorations des performances des mécanismes de routage P2P.

Chapitre 6

Conclusion générale

Nous l'avons vu, les réseaux IP sont régulièrement affectés par des incidents. Les causes de ces incidents sont d'origine variée : ils peuvent survenir à la suite d'une défaillance matérielle, ou suite à une mauvaise utilisation du réseau par exemple. Pour les utilisateurs de ces réseaux, il est parfois essentiel, voire critique, que leurs communications soient correctement acheminées. C'est pourquoi il est nécessaire de déployer des mécanismes de rétablissement réseau. En effet, lorsqu'un incident affecte une partie du réseau et empêche la délivrance d'une communication, ces mécanismes vont permettre de réacheminer cette communication par une partie du réseau non affectée par l'incident.

Les communications des utilisateurs, en particulier sur Internet, sont susceptibles de traverser une grande variété de réseaux et par conséquent, de dépendre de plusieurs mécanismes de rétablissement. Pour les utilisateurs ayant besoin de fiabilité pour leurs communications, les performances de ces mécanismes sont cruciales. Ils doivent en effet être capables de rétablir la délivrance d'une communication affectée par un incident en un temps suffisamment court pour satisfaire l'utilisateur. De plus, les ressources consommées par ces mécanismes doivent être limitées, afin de rendre leur utilisation possible.

Cependant, les mécanismes de rétablissement, tels que les protocoles de routage, sont généralement déployés par les opérateurs d'un réseau. Par conséquent, les utilisateurs n'ont que peu d'influence sur le fonctionnement de ceux-ci. Ainsi, les performances de ces mécanismes ne sont parfois pas adaptées aux besoins d'un utilisateur, pour une communication donnée. De plus, lorsque différents réseaux sont traversés par une communication, les différents mécanismes de rétablissement y étant déployés ne coopèrent généralement pas. Dans certaines situations d'incident, il peut en résulter une interruption bien trop longue de la délivrance des communications des utilisateurs.

Par conséquent, les mécanismes de rétablissement réseau rencontrés ne sont pas toujours en mesure de satisfaire les demandes de fiabilité des utilisateurs. Ils ne tiennent pas compte de la variété des besoins des différentes communications dans le réseau et leurs performances peuvent ne pas être satisfaisantes. Par conséquent, dans cette thèse, nous nous sommes intéressés à d'autres mécanismes de rétablissement réseau, capables de mieux satisfaire les utilisateurs ayant des besoins de fiabilité importants

Nous nous sommes intéressés aux systèmes de routage P2P. Ces systèmes sont déployés sur un ensemble de noeuds reliés au réseau, qui coopèrent afin de leur permettre d'acheminer des communications. Dans ces systèmes, les noeuds participants forment un réseau logique, superposé au réseau IP et ont un rôle de routeur dans ce réseau : les communications peuvent transiter par eux pour être acheminées d'un noeud à un autre.

Dans un réseau IP, un unique chemin est généralement utilisé pour acheminer des communications entre deux noeuds. L'utilisation du routage P2P permet un plus grand choix dans les chemins à utiliser. En effet, il va être possible de faire transiter une communication par n'importe quelle succession de noeuds participant au système de routage P2P. Le choix du chemin à utiliser dépend normalement de la politique de routage utilisée dans le système P2P.

Nos travaux se concentrent sur l'utilisation des systèmes de routage P2P en tant que mécanisme de rétablissement réseau. En effet, lorsqu'un incident affecte le chemin par lequel une communication est acheminée par le réseau IP, nous proposons d'utiliser la variété de chemin offerte par le routage P2P afin de rétablir l'acheminement de la communication.

Nous avons étudié cette solution et l'avons trouvée pertinente pour répondre aux besoins des utilisateurs dont les communications ont un besoin de fiabilité élevé. En effet, les systèmes de routage P2P sont déployés par les utilisateurs eux-mêmes, et par conséquent, il leur est possible de spécifier leur fonctionnement afin de répondre à un certain besoin de fiabilité, pour une communication donnée.

Afin d'établir la pertinence de cette solution, nous avons soulevé différents problèmes devant être approfondis. Nous nous sommes tout d'abord intéressés à la détection d'incident par envoi de mes-

sages sondes. Cette technique est en effet la principale technique de détection d'incident pouvant être utilisée dans les systèmes de routage P2P. Ils ont un rôle crucial dans les performances du mécanisme de détection d'incident. En effet, ils doivent être en mesure de détecter un incident en un temps limité, en fonction des besoins de l'utilisateur. De plus, il ne doit pas générer trop de faux positifs, et doit consommer le moins de ressources possible.

Nous avons présenté trois types de mécanisme de détection d'incident par envoi de messages sondes. Après avoir étudié leurs performances en détail, nous avons montré qu'ils sont en mesure de détecter des incidents en moins d'une seconde, même dans des réseaux au fonctionnement dégradé. Cependant, ceci est possible au prix d'un coût de fonctionnement élevé et d'un risque d'apparition de faux positifs non négligeable. Par contre, nous avons montré que pour des temps de l'ordre d'une seconde ou plus, l'utilisation de ces mécanismes est tout à fait satisfaisante.

Nous nous sommes ensuite consacrés à la conception d'un système de routage P2P dédié au rétablissement réseau. En fonction des besoins spécifiés par l'utilisateur pour la fiabilité d'une de ces communications, ce système se configure de manière à détecter un éventuel incident affectant la communication et à la réacheminer lorsque cela est nécessaire. Pour cela, nous avons introduit l'utilisation de plusieurs techniques, telles que le routage par la source et le double acheminement. Nous avons de plus réalisé une implémentation de ce système, et l'avons testée dans un réseau virtualisé et sur Internet.

Nous avons montré que notre système est capable de satisfaire les besoins de l'utilisateur lorsqu'une interruption de la délivrance des communications d'une seconde environ est tolérée. Le temps de rétablissement d'une communication offert par notre système peut parfois être plus court, mais ceci n'est pas systématique. Nous avons étudié la consommation de bande passante réseau entrainée par l'utilisation de notre mécanisme et avons montré que celle-ci est fonction du besoin de fiabilité demandé par l'utilisateur pour une communication ainsi que du nombre de communication à protéger. Ainsi, cette consommation peut-être importante si les besoins des utilisateurs sont importants, mais restera modérée sinon.

Afin de compléter notre étude des performances de notre système et des mécanismes basés sur le routage P2P en général, nous avons enfin effectué une simulation de leurs comportements dans un grand réseau interconnecté. Ceci nous permit en particulier d'évaluer la portée de ces mécanismes de rétablissement, en comparaison avec les protocoles de routage usuellement déployés dans les réseaux IP. La portée d'un mécanisme représente sa capacité à rétablir l'acheminement d'une communication affectée par un incident. Nous avons pu vérifier que la portée des mécanismes de routage P2P était proche de celle des protocoles de routage classiques. De plus, nous avons observé que la portée de notre mécanisme était proche de celle de RON, qui est pourtant optimale. Ces expérimentations confirment la capacité des systèmes de routage P2P à rétablir les communications affectées par des incidents, et ainsi améliorer leurs fiabilités.

Tout au long de cette thèse, nous avons étudié différents problèmes afin de proposer un mécanisme de rétablissement réseau centré sur les utilisateurs, mais aussi apportant les meilleures performances possible. Les principaux critères de performances de ce type de mécanismes, évoqués dans la section 2.4.1 sont la portée, le temps de rétablissement, la qualité du chemin alternatif et la consommation de ressource et le passage à l'échelle. Le tableau 6.1 présente les performances de notre mécanisme de rétablissement réseau, tel que nous pouvons les synthétiser après les travaux présentés dans ce document. Nous les comparons aux performances de deux autres types de mécanismes utilisés aujourd'hui : le routage dynamique sur Internet et le système RON, que nous pouvons déduire de l'étude de ces systèmes réalisée dans les sections 2.4.2 et 2.5.3.

Ce tableau met en évidence la principale contribution de nos travaux qui est de proposer un système de rétablissement réseau performant et qui répond aux besoins des utilisateurs. Il permet en effet

TAB. 6.1: Récapitulatif des performances de différents mécanismes de rétablissement réseau

	Mécanisme :		
Critère :	Routage Internet (ex : OSPF + BGP)	RON	Notre système
Temps de rétablissement	Quelques centaines de millisecondes à quelques secondes pour un incident intra-AS, quelques dizaines de secondes pour un incident inter-AS	Quelques secondes	Parfois inférieur à 1 seconde
Portée	Maximale pour une topologie IP donnée en IGP, dépend des accords entre opérateurs de réseau en EGP	Dépends du nombre de noeuds overlay et de leur position dans la topologie du réseau IP	Idem RON + dépend du choix et du nombre de chemins de secours potentiels
Qualité du chemin alternatif	Maximale pour une topologie IP donnée en IGP, dépend des accords entre opérateurs de réseau en EGP	Plusieurs critères possibles et parfois meilleure que le chemin Internet	Idem ci-dessus
Consommation de ressource et passage à l'échelle	Consommation modérée quel . que soit le trafic à protéger. Passage à l'échelle OK	Consommation moyenne quel que soit le trafic à protéger. Pas de passage à l'échelle	Consommation adaptée au trafic à protéger. Passage à l'échelle OK

un compromis entre les performances des mécanismes de rétablissement déployés par les opérateurs et les mécanismes de routage P2P du type de RON : le niveau de fiabilité des communications est similaire à celui des mécanismes des opérateurs, sans toutefois entrainer une consommation de ressource trop importante comme l'aurait fait un mécanisme tel que RON.

Les perspectives pouvant faire suite à ces travaux sont multiples. Tout d'abord, comme nous l'avons vu dans ce document, le rôle joué par les mécanismes de détection d'incident dans le rétablissement réseau est crucial. De plus, ce type de mécanisme est présent dans de nombreux autres systèmes. Par conséquent, il serait probablement bénéfique d'approfondir leurs études. Par exemple, une amélioration du fonctionnement des mécanismes pourrait être recherchée afin d'améliorer leurs performances. De plus, la généralisation de ces mécanismes à d'autres types d'incidents, et non plus uniquement à la rupture de la connectivité, pourrait être envisagée afin de permettre leur utilisation par des systèmes aux objectifs plus larges que le rétablissement réseau.

Nous avons aussi vu que le système de rétablissement présenté dans cette thèse pourrait être complété. Outre les diverses optimisations envisageables pour améliorer ses performances, il serait, pour lui aussi, envisageable d'élargir son domaine d'utilisation. D'autres techniques destinées à augmenter la qualité des communications des utilisateurs peuvent par exemple être utilisées dans les réseaux overlays. Ceci permet d'envisager la conception d'un logiciel destiné à l'amélioration de la qualité des communications des utilisateurs, à leurs demandes, et en fonction de leurs besoins.

Les résultats de nos travaux ont de plus permis de mettre en évidence certains manques dans les systèmes de rétablissement déployés par les opérateurs des réseaux. En effet, ces systèmes fonctionnent comme des boites noires pour les utilisateurs. Par exemple, nous avons vu que la diversité des chemins pour joindre une destination dans un réseau ne peut être exploitée par l'utilisateur sans une intervention de l'opérateur. De même, pour ce qui est de la fiabilité tout du moins, les communications sont traitées de la même façon dans le réseau, alors que les utilisateurs ont des attentes différentes. Il parait bien sûr techniquement difficile de mettre au point un système satisfaisant l'ensemble des utilisateurs et pouvant traiter un très grand nombre de communications en même temps. On peut par conséquent se demander si une certaine ouverture de ces systèmes aux utilisateurs ne permettrait pas d'améliorer leur satisfaction, en particulier lorsque ceux-ci ont des besoins spécifiques pour certaines de leurs communications.

Bibliographie

[1] ANDERSEN D., BALAKRISHNAN H., FRANS KAASHOEK M., *et al.* Resilient Overlay Networks. *Proceedings of the 18th ACM SOSP*, 2001.

[2] ANDERSEN D., SNOEREN A., BALAKRISHNAN H. Best-path vs. multi-path overlay routing. *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, 2003.

[3] AUTORITÉ DE RÉGULATION DES COMMUNICATIONS ÉLECTRONIQUES ET DES POSTES. Recommandations relatives à la définition des prestations d'accès à la boucle locale et à leur mise en oeuvre opérationnelle, 2000.

[4] BARABASI A., ALBERT R. Emergence of scaling in random networks. *Science*, 1999.

[5] CASTRO M., DRUSCHEL P., KERMARREC A., *et al.* SplitStream : High-bandwidth content distribution in cooperative environments. *Proceedings of the nineteenth ACM symposium on Operating systems principles*, 2003.

[6] CHENG C., HUAN Y., KUNG H., *et al.* Path probing relay routing for achieving high end-to-end performance. *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM'04)*, 2004.

[7] CHU Y., RAO S., SESHAN S., *et al.* A Case for End System Multicast. *Proceedings of ACM Sigmetrics*, 2000.

[8] CISCO. Enhanced IGRP. `http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito\_doc/en\_igrp.htm`.

[9] CISCO. Interior Gateway Routing Protocol. `http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito\_doc/igrp.htm`.

[10] CLARKE I., MILLER S., HONG T., *et al.* Protecting free expression online with freenet. *IEEE Internet Computing*, 2002.

[11] CLAUSEN T., JACQUET P. RFC 3626 : Optimized Link State Routing Protocol (OLSR). Technical report, Internet Engineering Task Force, 2003.

[12] COHEN A., RANGARAJAN S. Layer 4/7 switching and other custom IP traffic processing using the NEPPI API. Technical report, Bell Lab Whitepaper, 1999.

[13] COHEN B. Incentives to Build Robustness in BitTorrent. *Proceedings of the 1st Workshop on Economics of Peer-to-Peer Systems*, 2003.

[14] COLLINS A. The Detour framework for packet rerouting. *PhD Qualifying Examination*, 1998.

[15] DABEK F., COX R., KAASHOEK F., *et al.* Vivaldi : A Decentralized Network Coordinate System, 2004.

[16] DEDINSKI I., HOFMANN A., SICK B. Cooperative Keep-Alives : An Efficient Outage Detection Algorithm for P2P Overlay Networks. *Proceedings of the Seventh IEEE International Conference on Peer-to-Peer Computing*, 2007.

[17] DONGHUI G., ZHIYU Z., HANYI Z. A Novel Algorithm for Fast Detection of Network Failure. *Photonic Network Communications*, 2005.

[18] ESTRIN D., LI T., REKHTER Y., *et al.* RFC 1940 : Source demand routing : Packet format and forwarding specification (version 1). Technical report, Internet Engineering Task Force, 1996.

[19] FEAMSTER N., ANDERSEN D., BALAKRISHNAN H., *et al.* Measuring the Effects of Internet Path Faults on Reactive Routing. *Proceedings of ACM SIGMETRICS*, 2003.

[20] FRAGOULI C., LE BOUDEC J.Y., WIDMER J. Network coding : an instant primer. *SIGCOMM Comput. Commun. Rev.*, 2006.

[21] GARBER L. Denial-of-service attacks rip the Internet. *IEEE Computer*, 2000.

[22] GIBSON S. Distributed reflection denial of service, 2002. `http://grc.com/dos/drdos.htm`.

[23] GILLE M., ROHLOFF K., MANGHWANI P., *et al.* Scalable, Adaptive, Time-Bounded Node Failure Detection. *Proceedings of the 10th IEEE High Assurance Systems Engineering Symposium (HASE'07)*, 2007.

[24] GLENN M. A Summary of DoS/DDoS Prevention, Monitoring and Mitigation Techniques in a Service Provider Environment, 2003. `http://www.sans.org/reading-room/whitepapers`.

[25] GOYAL M., RAMAKRISHNAN K., FENG W. Achieving faster failure detection in OSPF networks. *Proceedings of IEEE International Conference on Communications*, 2003.

[26] GROVER W. Availabilty Analysis and APS Systems. *A monograph distributed at DRCN 2007*, 2007.

[27] GUMMADI K., MADHYASTHA H., GRIBBLE S., *et al.* Improving the reliability of internet paths with one-hop source routing. *Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation (OSDI'04)*, 2004.

[28] HUANG C., WANG A., LI J., *et al.* Measuring and evaluating large-scale CDNs. *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement (IMC)*, 2008.

[29] HUBERT B., GRAF T., MAXWELL G., *et al.* Linux Advanced Routing and Traffic Control, 2010. `http://lartc.org/`.

[30] IEEE 802.3 ETHERNET WORKING GROUP. Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications. Technical report, IEEE Computer Society, 2001.

[31] INTERNET WORLD STATS. World Internet Users and Population Stats, 2009. `http://www.internetworldstats.com/stats.htm`.

[32] ITU RECOMMENDATION. X. 200 : Open system interconnection : the basic reference model. Technical report, ITU, 1994.

[33] JACOBSON V. Congestion avoidance and control. *ACM SIGCOMM Computer Communication Review*, 1995.

[34] JOHNSON D., HU Y., MALTZ D. RFC 4728 : The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. Technical report, Internet Engineering Task Force, 2007.

[35] KARIG D., LEE R. Remote denial of service attacks and countermeasures. *Princeton University Department of Electrical Engineering Technical Report CE-L2001-002*, 2001.

[36] KATZ D., WARD D. IETF Draft : Bidirectional Forwarding Detection. Technical report, Internet Engineering Task Force, 2008.

[37] KENT S., SEO K. RFC 4301 : Security architecture for the internet protocol. Technical report, Internet Engineering Task Force, 2005.

[38] KRISHNAN R., MADHYASTHA H., SRINIVASAN S., *et al.* Moving beyond end-to-end path information to optimize cdn performance. *Proceedings of the Internet Measurement Conference (IMC)*, 2009.

[39] KUHN D. Sources of failure in the public switched telephone network. *IEEE Computer*, 1997.

[40] KUZMANOVIC A., KNIGHTLY E. Low-rate TCP-targeted denial of service attacks : the shrew vs. the mice and elephants. *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communications*, 2003.

[41] KVALBEIN A., HANSEN A., CICIC T., *et al.* Fast recovery from link failures using resilient routing layers. *10th IEEE Symposium on Computers and Communications*, 2005.

[42] KWOK T. Residential broadband Internet services and applications requirements. *Communications Magazine, IEEE*, 1997.

[43] LAPRIE J.C. Sûreté de fonctionnement des systèmes informatiques et tolérance aux fautes. *Techniques de l'ingénieur. Informatique industrielle.*, 1989.

[44] LAPRIE J.C., KANOUN K. *Software reliability and system reliability*, McGraw-Hill, Inc., pages 27–69. 1996.

[45] LAU F., RUBIN S., SMITH M., *et al.* Distributed denial of service attacks. *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, 2000.

[46] LI Z., MOHAPATRA P. The Impact of Topology on Overlay Routing Service. *Proceedings of INFOCOM : Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, 2004.

[47] LICHTWALD G., ZITTERBART M., WALTER U. Improving Convergence Time of Routing Protocols. *Proceeding of the 3rd International Conference on Networking*, 2004.

[48] LINUX MAN-PAGES PROJECT. raw, SOCK_RAW - Sockets brutes (raw) IPv4 sous Linux., 2010. Manuel du programmeur Linux.

[49] LIU Y., REDDY A. A fast rerouting scheme for OSPF/ISIS Networks. *Proceedings of Computer Communications and Networks (ICCCN 2004)*, 2004.

[50] LUMEZANU C., BADEN R., LEVIN D., *et al.* Symbiotic relationships in Internet routing overlays. *Proceeding of USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2009.

[51] MA W., SHEN B., BRASSIL J. Content services network : the architecture and protocols. *Proceedings of the 6th IWCW*, 2001.

[52] MALKIN G. RFC 2453 : RIP Version 2. Technical report, Internet Engineering Task Force, 1998.

[53] MARKOFF J. Before the Gunfire, Cyberattacks. *The New York Times*, 2008.

[54] MARKOPOULOU A., IANNACCONE G., BHATTACHARYYA S., *et al.* Characterization of failures in an IP backbone. *Proceedings of INFOCOM : Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, 2004.

[55] MEDARD M., FINN S., BARRY R., *et al.* Redundant trees for preplanned recovery in arbitrary vertex-redundant or edge-redundant graphs. *IEEE/ACM Transactions on Networking*, 1999.

[56] MEDINA A., LAKHINA A., MATTA I., *et al.* BRITE : An Approach to Universal Topology Generation. *Proceedings of the International Symposium on Modeling, Analysis, and Simulation of Computer Systems*, 2001.

[57] MIRKOVIC J., REIHER P. A taxonomy of DDoS attack and DDoS defense mechanisms. *SIGCOMM Comput. Commun. Rev.*, 2004.

[58] MOLNÁR M., TEZEGHDANTI M. Reroutage dans OSPF avec des chemins de secours. *Projet ARMOR, Rapport de recherche 4340*, 2001.

[59] MOORE D., SHANNON C., BROWN D., *et al.* Inferring Internet denial-of-service activity. *ACM Trans. Comput. Syst.*, 2006.

[60] MORRIS R., YIP A. NATRON : overlay routing to oblivious destinations. Technical report, Massachusetts Institute of Technology, 2002.

[61] MOVSICHOFF A., LAGOA C., CHE H. End-to-end optimal algorithms for integrated QoS, traffic engineering, and failure recovery. *IEEE/ACM Trans. Netw.*, 2007.

[62] MOY J. RFC 2328 : OSPF Version 2. Technical report, Internet Engineering Task Force, 1998.

[63] MUELLER M. Digital convergence and its consequences. *Javnost—The Public*, 1999.

[64] NAVEH B. JGraphT, 2010. `http://jgrapht.sourceforge.net`.

[65] OPPENHEIMER D. Why Do Internet Services Fail, and What Can Be Done About It ? Technical report, University of California at Berkeley, 2002.

[66] ORAN D. RFC 1142 : OSI IS-IS Intra-domain Routing Protocol. Technical report, Internet Engineering Task Force, 1990.

[67] PAN P., SWALLOW G., ATLAS A. RFC 4090 : Fast Reroute Extensions to RSVP-TE for LSP Tunnels, 2005.

[68] PAPADAKIS H., ROUSSOPOULOS M., FRAGOPOULOU P., *et al.* Imbuing Unstructured P2P Systems with Non-intrusive Topology Awareness. *Proceedings of the Ninth International Conference on Peer-to-Peer Computing (P2P 2009)*, 2009.

[69] PARALLELS. OpenVZ, 2010. `http://wiki.openvz.org/`.

[70] PATHAN A., BUYYA R. A taxonomy and survey of content delivery networks. Technical report, Grid Computing and Distributed Systems (GRIDS) Laboratory, University of Melbourne, Parkville, Australia, 2006.

[71] PEI G., GERLA M., CHEN T., *et al.* IETF Draft : Fisheye State Routing Protocol (FSR) for Ad Hoc Networks. Technical report, Internet Engineering Task Force, 2000.

[72] PERKINS C., BELDING-ROYER E., DAS S. RFC 3561 : Ad hoc On-Demand Distance Vector (AODV) Routing. Technical report, Internet Engineering Task Force, 2003.

[73] PETHIA R., CROCKER S., FRASER B. RFC 1281 : Guidelines for the secure operation of the Internet. Technical report, Internet Engineering Task Force, 1991.

[74] POSTEL J. RFC 791 : Internet protocol. Technical report, Internet Engineering Task Force, 1981.

[75] POSTEL J. RFC 792 : Internet Control Message Protocol. Technical report, Internet Engineering Task Force, 1981.

[76] POSTEL J. RFC 793 : Transmission control protocol. Technical report, Internet Engineering Task Force, 1981.

[77] POSTEL J., AL. RFC 768 : User datagram protocol. Technical report, Internet Engineering Task Force, 1980.

[78] POULSEN K. FBI busts alleged DDoS Mafia, 2004. Security Focus : http://www.securityfocus.com/news/9411.

[79] PUJOLLE G., SALVATORI O., NOZICK J. *Les réseaux*, Eyrolles, pages 77–108. 2000.

[80] PUTTASWAMY K., SALA R., WILSON C., *et al.* Protecting anonymity in dynamic peer-to-peer networks. *Proceedings of the IEEE International Conference on Network Protocols (ICNP)*, 2008.

[81] RAMASUBRAMANIAN V., MALKHI D., KUHN F., *et al.* On the Treeness of Internet Latency and Bandwidth. *Procedings of SIGMETRICS/Performance*, 2009.

[82] RATNASAMY S., HANDLEY M., KARP R., *et al.* Topologically-aware overlay construction and server selection. *Proceeding of INFOCOM*, 2002.

[83] REKHTER Y., LI T., HARES S. RFC 4271 : A Border Gateway Protocol 4 (BGP-4). Technical report, Internet Engineering Task Force, 2006.

[84] ROSEN E., VISWANATHAN A., CALLON R. RFC 3031 Multiprotocol Label Switching Architecture. Technical report, Internet Engineering Task Force, 2001.

[85] ROWSTRON A., DRUSCHEL P. Pastry : Scalable, distributed object location and routing for large-scale peer-to-peer systems. *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, 2001.

[86] ROZYCKI P., KORNIAK J., JAJSZCZYK A. Failure Detection and Notification in GMPLS Control Plane. *GMPLS Performance : Control Plane Resilience, 2007 Workshop on*, 2007.

[87] RUDIGER S. A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. *the IEEE 2001 International Conference on Peer-to-Peer Computing (P2P2001)*, 2001.

[88] RUSSELL R., WELTE H. Linux netfilter Hacking HOWTO, 2010. http://www.iptables.org/documentation/HOWTO/netfilter-hacking-HOWTO.html.

[89] SAROLAHTI P., FLOYD S., M. K. IETF Draft : Transport-layer Considerations for Explicit Cross-layer Indications. Technical report, Internet Engineering Task Force, 2007.

[90] SCHUBA C., KRSUL I., KUHN M., *et al.* Analysis of a Denial of Service Attack on TCP. *Proceedings of IEEE Symposium on Security and Privacy*, 1997.

[91] SEETHARAMAN S., AMMAR M. On the Interaction Between Dynamic Routing in Native and Overlay Layers. *Proceedings of the 25th IEEE International Conference on Computer Communications*, 2006.

[92] SHIELDS C. What do we mean by network denial-of-service. *Proceedings of the IEEE Workshop on Information Assurance and Security*, 2002.

[93] SOCOLOFSKY T., KALE C. RFC 1180 : TCP/IP Tutorial. Technical report, Internet Engineering Task Force, 1991.

[94] SONTAG D., ZHANG Y., PHANISHAYEE A., *et al.* Scaling all-pairs overlay routing. *Proceedings of the 2009 ACM Conference on Emerging Network Experiment and Technology (CoNEXT)*, 2009.

[95] SRISURESH P., EGEVANG K. RFC 3022 : Traditional IP Network Address Translator (Traditional NAT). Technical report, Internet Engineering Task Force, 2001.

[96] STAMATELAKIS D., GROVER W. IP Layer Restoration and Network Planning Based on Virtual Protection Cycles. *IEEE Journal on Selected Areas in Communications (JSAC)*, 2000.

[97] STOICA I., ADKINS D., ZHUANG S., *et al.* Internet indirection infrastructure. *Proceedings of the 2002 SIGCOMM conference*, 2002.

[98] STOICA I., MORRIS R., KARGER D., *et al.* Chord : A Scalable Peer-to-peer Lookup Service for Internet Applications. *Proceedings of the ACM SIGCOMM'01 Conference*, 2001.

[99] VASSEUR J., PICKAVET M., DEMEESTER P. *Network recovery*, Elsevier, chapter 1.2.3. 2004.

[100] VASSEUR J., PICKAVET M., DEMEESTER P. *Network recovery*, Elsevier, chapter 5. 2004.

[101] VASUDEVAN V., ANDERSEN D., ZHANG H. On Internet Availability : Where Does Path Choice Matter ? Technical report, Carnegie Mellon University, 2009.

[102] VAUGHN R., EVRON G. DNS amplification attacks (Preliminary Release), 2006.

[103] VELEZ F., CORREIA L. Mobile broadband services : classification, characterization, and deployment scenarios. *Communications Magazine, IEEE*, 2002.

[104] VERMA D. Service level agreements on IP networks. *Proceedings of the IEEE*, 2004.

[105] XING J.Y., CHAN W., YAJUN WANG S. Network topology inference based on end-to-end measurements. *IEEE JSAC*, 2006.

[106] XUE G., CHEN L., THULASIRAMAN K. Delay reduction in redundant trees for pre-planned protection against singlelink/node failure in 2-connected graphs. *IEEE GLOBECOM*, 2002.

[107] ZHANG M., ZHANG C., PAI V., *et al.* PlanetSeer : Internet Path Failure Monitoring and Characterization in Wide-Area Services. *Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation*, 2004.

[108] ZHANG W., HE J. Modeling End-to-End Delay Using Pareto Distribution. *Proceedings of the Second International Conference on Internet Monitoring and Protection (ICIMP '07)*, 2007.

[109] ZHAO B., HUANG L., STRIBLING J., *et al.* Exploiting Routing Redundancy via Structured Peer-to-Peer Overlays. *Proceedings of ICNP*, 2003.

[110] ZHAO B., KUBIATOWICZ J., JOSEPH A. Tapestry : An infrastructure for fault-tolerant wide-area location and routing. *IEEE Computer*, 2001.

[111] ZHENG H., LUA E., PIAS M., *et al.* Internet routing policies and round-trip-times. *Passive and Active Network Measurement*, 2005.

[112] ZHI L., LIHUA Y., PRASANT M., *et al.* On the analysis of overlay failure detection and recovery. *Comput. Netw.*, 2007.

[113] ZHUANG S., GEELS D., STOICA I., *et al.* On failure detection algorithms in overlay networks. *Proceedings IEEE of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)*, 2005.