Comparaison quantivative de différentes techniques de restauration rapide dans les réseaux IP/MPLS

Laurent MELON[1] — **Guy LEDUC**

Université de Liège
Research Unit in Networking
Institut d'Electricite Montefiore (B28)
B-4000 LIEGE 1
BELGIUM

{melon,leduc}@run.montefiore.ulg.ac.be

RÉSUMÉ. *Cet article a pour objet une comparaison quantitative des performances de différentes approches de restauration rapide de chemins dans un réseau Multi-Protocol Label Switching (MPLS). L'objectif poursuivi est d'analyser le comportement général de ces approches (restauration locale, de bout en bout,...) en situation de panne par le biais de simulations. Cette étude est réalisée dans le cadre de trafics ne réclamant aucune qualité de service particulière (Best Effort), le routage des LSPs utilisant un algorithme de plus court chemin. De nombreux articles justifient le choix entre une approche "de bout-en-bout" et/ou locale par la simple intuition et présentent, dans ce cadre, un algorithme de routage des chemins de backup optimisant différents paramètres. Notre approche tente de clarifier la situation en comparant, sur base de résultats concrets, les mérites et défauts respectifs de chaque mécanisme.*

ABSTRACT. *This article is dealing with a quantitative comparison of different fast restoration approaches in the context of a Multi-Protocol Label Switching (MPLS) network. The objective is to analyse the general behaviour of these approaches (local rerouting, edge-to-edge rerouting,...) in case of failure by means of simulations. This study is realised in the context of Best Effort only traffic, routing being done with a shortest paths algorithm. Many papers justifies the choice between an edge-to-edge or a local scheme by means of the intuition and then present a new algorithm for routing backup paths which try to optimize some parameters. Our approach aims to clarify the situation by comparing by means of concrete results the pros and cons of each mechanism.*

MOTS-CLÉS : *MPLS, restauration rapide, simulations*

KEYWORDS: *MPLS, fast restoration, simulations*

1re soumission à *CFIP 2002*, le 4 février 2002.

1. Aspirant du Fonds National de la Recherche Scientifique (FNRS).

1. Introduction

A l'heure de la société de l'information, la fiabilité des réseaux de communication est d'une importance vitale. Bon nombre d'opérateurs Internet commencent à offrir à leurs clients différentes formes de qualité de service. Mais la notion même de qualité de service n'a de sens que si l'on est capable de maintenir celle-ci indépendamment de tout événement se produisant dans le réseau. Or les dispositifs de routage, qu'ils soient optiques ou électroniques, ne peuvent garantir à eux seuls une fiabilité suffisante. Il faut donc construire le réseau avec un certain niveau de redondance et mettre en place les protocoles permettant d'utiliser cette redondance pour fiabiliser le transfert d'information.

Le protocole IP lui même et les protocoles de routage qui lui sont associés sont déjà conçus pour assurer une forme de robustesse au réseau. Cependant, le temps de convergence des algorithmes de routage sur une topologie de taille moderne est largement insuffisant car variant de quelques secondes à plusieurs minutes dans certains cas pathologiques. Si l'on désire pouvoir intégrer les réseaux dédiés à la voix avec les réseaux IP, de bien meilleures performances seront nécessaires.

D'autres solutions ont donc été développées. Dans le monde des réseaux téléphoniques la technologie Synchronous Digital Hierarchy(SDH) ou Synchronous Optical NETwork (SONET) est couramment rencontrée. Ces protocoles autorisent la création de Self-Healing Ring (SHR), topologie en forme de double anneau pouvant être rebouclé sur lui-même en cas de panne afin de restaurer la connectivité. Toutefois, les topologies utilisables sont trop limitées (uniquement l'anneau) et la quantité de ressources inutilisées en l'absence de panne est beaucoup trop importante. De plus, la granularité de restauration est insuffisante : l'ensemble du trafic circulant sur un lien est re-routé alors que souvent un opérateur désirera pouvoir définir plus finement quel type de flux fera l'objet d'une protection.

Des alternatives doivent donc encore être développées. Comme le met en évidence [OWE 01b], pratiquement toutes les couches de la pile de protocoles TCP/IP peuvent se voir adjoindre des fonctions de protection et de restauration. Il y a d'ailleurs fort à parier que, dans le futur, plusieurs couches collaboreront pour atteindre une fiabilité et des performances maximales.

Cependant la technologie MPLS [ROS 01] apparaît comme la position idéale pour un algorithme de restauration rapide. Le protocole MPLS se base comme Asynchronous Transmission Mode (ATM) sur le concept de "circuits virtuels" appelés Labelled Switched Paths (LSPs). Un protocole de signalisation est nécessaire pour établir ces LSPs. Deux protocoles de ce type ont été définis : Label Distribution Protocol (LDP) et une extension de ReSerVation Protocol (RSVP). Une fois les chemins établis, la propagation d'un paquet ne nécessite plus la consultation d'une table de routage. A son entrée dans le réseau le paquet se voit attribuer, sur base d'un ensemble de paramètres (destination, qualité de service, ...) un label (stocké dans l'en-tête MPLS). A chaque nœud c'est ce label qui sera utilisé pour sélectionner le chemin à suivre sans devoir "re-classifier" le paquet à nouveau. De plus les en-têtes MPLS sont empilables

et permettent donc de créer une hiérarchie MPLS, d'agréger plusieurs LSPs en un seul ou de réaliser des tunnels directement au niveau de la couche MPLS.

MPLS risque de s'imposer comme un protocole de choix pour les opérations de restauration rapide. Tout d'abord, il semble que l'objectif à long terme soit, pour des raisons évidentes de performance et de facilité de configuration, de simplifier au maximum la pile de protocoles pour aboutir à IP/MPLS directement sur fibre optique. Dans cette philosophie, MPLS sera la première couche "intelligente" où il sera possible de placer ces fonctions. De plus, la hiérarchie de labels et la possibilité de réaliser efficacement un routage à la source sont autant d'avantages offerts par MPLS. Ils sont d'un grand attrait pour résoudre tant les problèmes d'ingénierie de trafic que ceux propres à la restauration rapide.

Un certain nombre de travaux se sont déjà penchés sur les problèmes de restauration rapide tant dans le cadre du protocole ATM que des réseaux MPLS. La majorité d'entre eux ont pour objectif de minimiser la quantité de ressources à réserver pour permettre une bonne restauration. [KOD 01, KOD 00, Wei 01] présentent des algorithmes exacts ou probabilistes pour router des demandes d'établissement de chemins et analysent notamment le taux de blocage induit par chaque méthode.

D'autres recherches définissent les protocoles de signalisation indispensables à l'établissement de circuits de backup et à la notification de pannes mais en restant neutres par rapport à une méthode de restauration complète. Ainsi [IWA 01, OWE 01a] présentent des extensions pour différents protocoles permettant de prendre en charge des procédures de restauration rapide.

La justification du choix entre approche locale ou globale nous semble donc un problème insuffisamment traité. Partant de ce constat, nous avons voulu déterminer le comportement dynamique en cas de panne de différents mécanismes de restauration rapide. Ces mécanismes sont présentés dans la section suivante.

2. La restauration rapide

2.1. *Hypothèses et choix de design*

Ce travail suppose qu'une seule panne se produit simultanément dans le réseau, qu'elle soit de lien ou de nœud. Tous les LSPs, primaires et de backup, sont calculés suivant un algorithme de plus courts chemins (Dijkstra) et sont signalés par les protocoles Label Distribution Protocol (LDP, [AND 01]) et Constraint Routed Label Distribution Protocol (CR-LDP, [ABO 01]). Le simulateur utilisé est Network Simulator v2.1b8 modifié par nos soins afin d'inclure les algorithmes présentés ci-dessous.

2.2. *Restauration locale*

Chaque LSR (Label Switching Router) établit, *lorsque le réseau est en parfait état*,

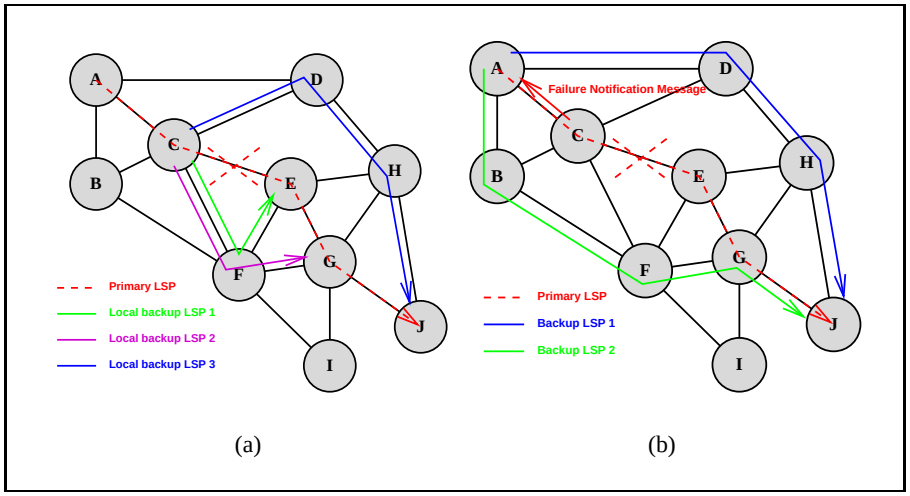

Figure 1. *Restauration locale et "de bout-en-bout"*

un ensemble de LSPs permettant de contourner toute panne d'un lien ou d'un routeur directement adjacent. Deux paramètres déterminent le chemin suivi par le LSP de backup :

– l'hypothèse sur la nature de la panne : le backup peut être établi de manière à prévenir une panne de lien ou de nœud.

– la destination du contournement peut être :

- le nœud situé à l'autre extrémité du lien détecté en panne (cf. figure 1(a), "backup LSP 1"). Un nœud ne peut évidemment détecter efficacement que la panne d'un lien et devra faire une hypothèse sur la vraie nature de la panne.
- le nœud egress pour le flux considéré (cf. figure 1(a), "backup LSP 3")
- n'importe quel LSR intermédiaire (cf. figure 1(a), "backup LSP 2")

Si une panne survient, chacun des LSPs qui empruntait l'interface maintenant défectueuse est redirigé dans un des LSPs qui contourne le problème, établis pendant la phase préliminaire.

2.3. *Restauration globale ou "de bout-en-bout"*

Lorsqu'une panne est détectée par le ou les routeurs immédiatement adjacents à celle-ci, une notification est envoyée à tous les nœuds ingress dont l'un des LSPs passe par le nœud ou le lien défectueux [cf. figure 1(b)].

Chaque ingress commence alors par supposer qu'il s'agit d'une panne de lien et calcule un nouveau chemin n'utilisant plus celui-ci. S'il s'agissait en fait d'une panne

de nœud, l'établissement peut échouer et le nœud être amené à changer son hypothèse concernant la panne avant de calculer et d'établir un nouveau chemin.

Dès que ce chemin est établi, l'ensemble du trafic utilisant le LSP pour lequel le nœud a reçu une notification est basculé sur le nouveau LSP.

> REMARQUE. — Même si la totalité des résultats présentés dans ce travail porte sur du trafic de type BE (Best Effort), certains problèmes propres à la gestion de la qualité de service ont déjà été pris en compte. Ainsi, nous n'avons pas évalué d'algorithmes de restauration "de bout-en-bout" avec pré-calcul du chemin de backup ; ce type de solution nous semble en effet incompatible avec une utilisation optimale des ressources. Dans cette situation, le backup doit être complètement disjoint du LSP primaire, ce qui implique un routage nettement sous-optimal. Différents travaux sont néanmoins consacrés à cette approche dont [MUR].

D'autres approches ont également été investiguées dans la littérature. Ainsi [STA 00] présente la notion de p-cycles tandis que [BRE] introduit celle de concaténation de chemins. Cependant, aucun de ces travaux ne réalise une comparaison avec les techniques plus classiques présentées ici, ni ne réalise une étude du comportement dynamique de leur approche.

2.4. *Mécanismes supplémentaires*

En plus de ces techniques de restauration, différents mécanismes ont été implémentés et évalués. Le premier de ceux-ci consiste, en cas de congestion, à utiliser le chemin de backup correspondant à l'interface congestionnée pour essayer de propager le trafic. Cette façon de procéder sera appelée CA (Congestion Avoidance mechanism).

La seconde option explorée consiste à augmenter la priorité des messages LDP dans le but d'accélérer l'établissement des LSPs de backup "de bout-en-bout" lors de la panne. Deux mécanismes ont donc été ajoutés ; l'un d'entre eux garanti que les paquets LDP ne seront jamais détruits tandis que l'autre sert ces paquets avec la plus haute priorité (tant que des paquets LDP sont présents dans la file, ils sont servis en premier).

3. Méthodologie de simulation

3.1. *Algorithmes évalués*

Disposant au sein du simulateur de tous ces mécanismes, nous avons choisi certaines combinaisons que nous désirions évaluer :

– Mécanismes de base :

- LNL (*L*ocal restoration towards *N*ext node under *L*ink failure Hypothesis)
- LNN (*L*ocal restoration towards *N*ext node under *N*ode failure Hypothesis)
- LEN (*L*ocal restoration towards *E*gress node under *N*ode failure Hypothesis)
- E (*E*dge-to-edge restoration)

– Options :

- CA (*C*ongestion *A*voidance mechanism)
- LND (*L*DP messages *N*ever *D*ropped)
- LHP (*L*DP messages have the *H*ighest *P*riority)

– Combinaisons évaluées :

- Méthodes locales : LNL, LNN, LEN, LNN+CA, LEN+CA
- Méthodes de bout-en-bout : E, E+LND, E+LHP
- Méthodes hybrides : E+LEN, E+LEN+CA, E+LEN+LND, E+LEN+LHP

3.2. *Topologies et approvisionnement en ressources*

Deux types de topologies ont été utilisées. Deux topologies ont été générées aléatoirement (une de celles-ci est présentée à la figure 2) tandis qu'une troisième représente le backbone UUnet USA (cf. figure 7).

1) Dans le cas des simulations avec des flux UDP, le choix de la capacité de chaque lien a été réalisé de deux manières :

- Dimensionnement uniforme : la capacité de chaque lien est choisie de sorte qu'avant la panne, le taux d'utilisation soit identique sur tous les liens et égal à 40, 60 ou 80%. Cependant, la capacité minimale d'un lien est fixée à $2\,Mbps$. La moyenne sur tous les liens du taux d'utilisation n'est donc pas exactement de 40, 60 ou 80% mais doit être recalculée. C'est cette nouvelle valeur qui servira d'abscisse pour la majorité des graphes présentés.

- Dimensionnement "orienté-panne" : la capacité de chaque lien est choisie de manière à garantir que, quelle que soit la panne de lien, la restauration "de bout-en-bout" permettra une restauration sans perte. Chaque lien se voit donc attribuer :

$$c_l = \gamma \max_{k=1..m} \sum_{i=1}^{n} \sum_{j=1}^{n} \delta_{i,j,k,l} \omega_{i,j} \quad \forall l = 1..m$$

Avec :

n	le nombre de nœuds
m	le nombre de liens
c_l	la capacité du lien l
$\delta_{i,j,k,l}$	vaut 1 si le trafic émis par le nœud i vers le nœud j emprunte le lien l lors d'une panne du lien k, vaut 0 sinon
$\omega_{i,j}$	la bande passante moyenne consommée par le trafic émis par le nœud i vers le nœud j
γ	un facteur correctif valant $\frac{1}{0.95}$ de sorte que le lien ne soit en moyenne pas utilisé à plus de 95% de sa capacité

La moyenne du taux d'utilisation servira également d'abscisse sur les graphiques présentant les résultats ($\approx 39\%$). Toutefois ce point ne sera pas relié aux autres afin d'insister sur la différence fondamentale entre ces deux types de dimensionnement.

2) Dans le cas des flux TCP, tous les liens se sont vu attribuer une capacité identique, le contrôle de congestion de TCP se chargeant de déterminer le point de fonctionnement du réseau.

4. Résultats

Une énorme quantité de résultats a été générée ; nous présentons ici les plus significatifs et représentatifs d'entre eux.

4.1. *Paramètres mesurés*

Les différents paramètres mesurés grâce aux simulations sont :

– Le taux de pertes après la panne. Il est calculé par la formule :

$$p = \frac{\text{nombre total de paquets arrivés à destination après la panne}}{\text{nombre total de paquets émis après la panne}}$$

– La variation moyenne du délai moyen. La moyenne inclut uniquement les flux subissant une *augmentation* de leur délai moyen supérieure à 1%.

– La jigue moyenne. La jigue est calculée pour chaque flux par différence entre le délai maximum et le délai minimum séparant l'émission d'un paquet de sa réception par son destinataire final.

4.2. *Topologie aléatoire 1 : panne du nœud 1*

Dans ce scénario, le protocole de transport utilisé est UDP. La matrice de charge est pleine, i.e. tous les nœuds terminaux émettent du trafic vers tous les autres, ce qui donne un total de 1600 flux différents. Les sources génèrent leur trafic en suivant une loi de Poisson.

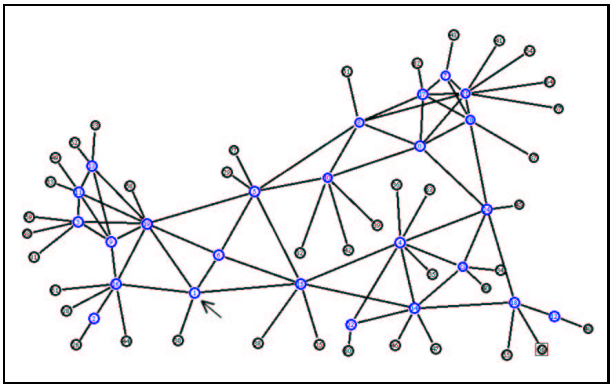

Figure 2. *L'une des topologies générée aléatoirement*

Nous simulons la panne du nœud 1 (indiqué sur la figure 2 par une flèche).

La figure 3 montre le taux de pertes observé après la panne. L'algorithme LNL se basant sur l'hypothèse d'une panne de lien n'est d'aucune utilité dans cette situation particulière. Il fournit une indication des performances "sans" restauration.

Les algorithmes locaux affichent des performances médiocres. Toutefois, lorsqu'ils sont combinés avec l'option CA le taux de pertes observé diminue fortement (près de 6% à forte charge).

Les algorithmes incluant une phase de restauration de bout-en-bout affichent des performances nettement meilleures. Étonnamment, la phase locale semble handicaper la restauration de bout-en-bout. Cela est dû au fait que dans cette série de simulations, aucune forme de priorité n'est utilisée pour les messages de contrôle. Ils perdent donc beaucoup de temps pendant la période de congestion amplifiée par la restauration locale, certains des messages LDP étant perdus et retransmis. Il faut donc beaucoup plus de temps pour établir les LSPs de backup globaux. A long terme, il va de soit que les taux de pertes des méthodes E et E+LEN convergeraient.

Le dimensionnement "orienté-panne" affiche des performances excellentes toujours dans le cas de la restauration de bout-en-bout.

La figure 4 présente la variation de délai moyenne sur tous les flux subissant une augmentation de délai supérieur à 1%. C'est sur ce plan que le réseau "orienté-panne" montre tout son intérêt. Toutes les méthodes de restauration affichent des performances supérieures à celles observées dans le cas du réseau dimensionné uniformément pourtant de charge inférieure.

La figure 5 détaille la répartition de la variation de délai sur les différents flux. On constate que, dans le cas de l'algorithme E, plus de 90% des flux subissent une

variation de délai inférieure à 60%. Dans le cas des méthodes LNN et E+LEN, la variation de délai qui borne 90% des flux s'élève à près de 80%.

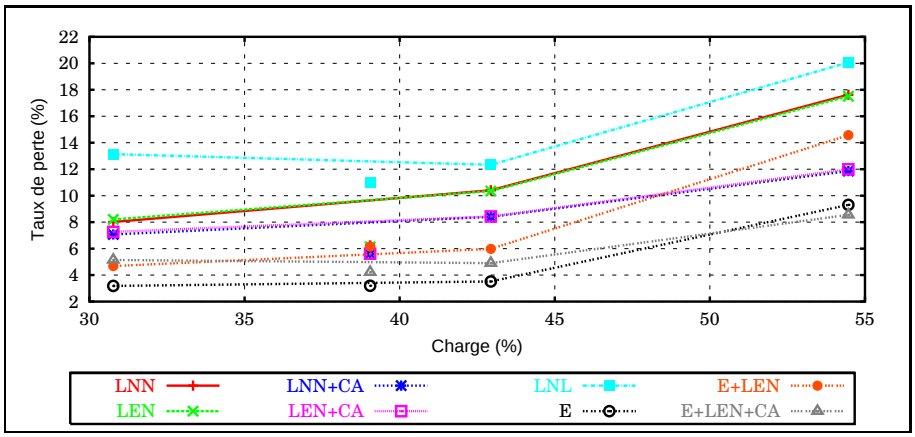

Figure 3. *Topologie aléatoire 1, panne du nœud 1, taux de pertes global*

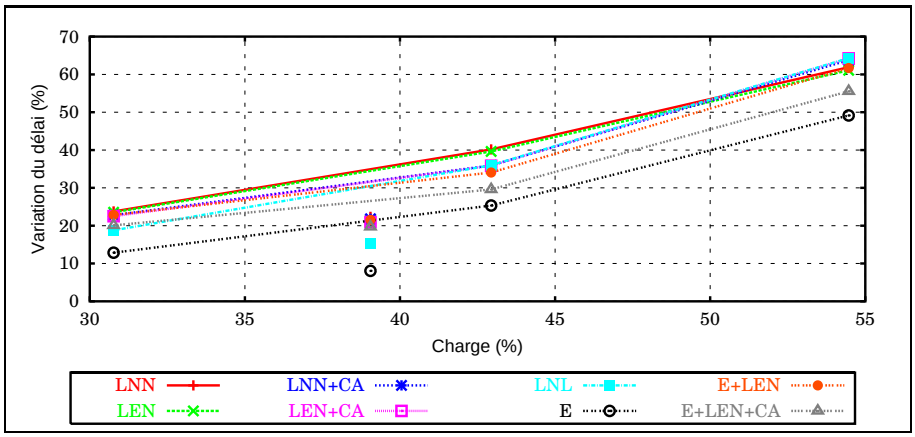

Figure 4. *Topologie aléatoire 1, panne du nœud 1, variation moyenne du délai moyen*

La figure 6 présente la jigue moyenne après la panne. Avant la panne, elle était inférieure à $6\,ms$ dans tous les dimensionnements. On observe le problème posé par l'option CA : la jigue est multipliée par un facteur deux ou trois pour les méthodes l'utilisant. La jigue sur le délai reste très basse pour toutes les méthodes dans le cas du dimensionnement "orienté-panne". La diminution de la jigue observée à forte charge peut être expliquée par l'important taux de pertes ; les paquets détruits n'influençant évidemment pas la valeur de la jigue.

La tableau 1 montre un des autres inconvénients de l'utilisation de l'option CA : le nombre important de paquets hors séquence. Il faut noter que toutes les méthodes

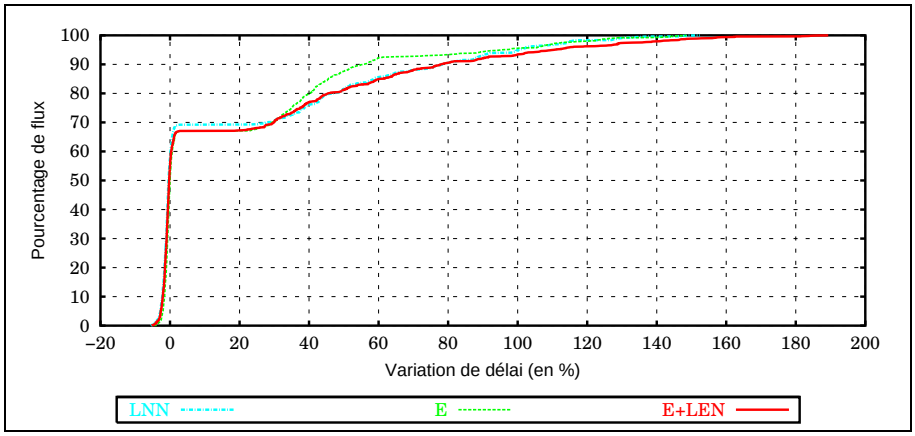

Figure 5. *Topologie aléatoire 1, panne du nœud 1, dimensionnement uniforme à 80%, répartition de la variation de délai*

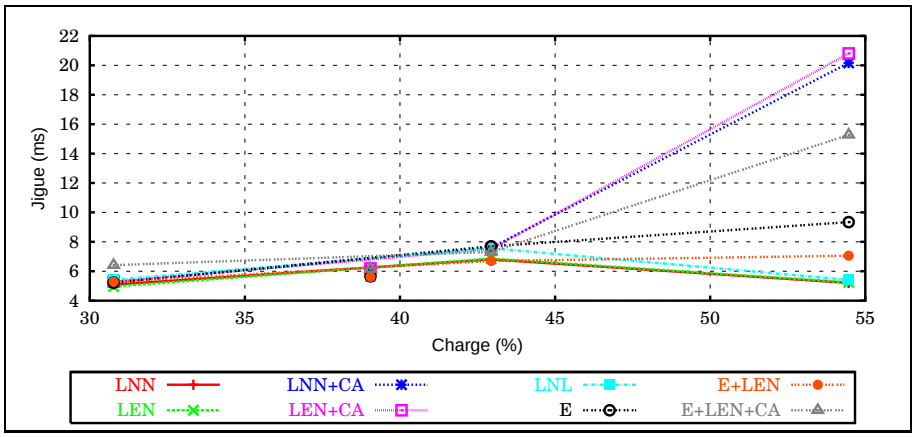

Figure 6. *Topologie aléatoire 1, panne du nœud 1, jigue moyenne sur le délai*

possédant une phase globale précédée d'une restauration locale sont susceptibles d'introduire un certain nombre, généralement très faible, de permutations de l'ordre des paquets.

Paramètre	Réseau	Algorithmes							
		LNN	LEN	LNN +CA	LEN +CA	LNL	E	E+ LEN	E+ LEN +CA
Nombre de sources	40 %	40	40	40	40	40	40	40	40
subissant	60 %	40	40	40	40	40	40	40	40
des pertes	80 %	40	40	40	40	40	40	40	40
	OP	40	40	40	40	40	40	40	40
Pourcentage de	40 %	28%	28%	29%	29%	21%	13%	24%	28%
flux subissant	60 %	32%	33%	36%	35%	26%	26%	35%	39%
une variation	80 %	31%	32%	58%	58%	25%	36%	34%	55%
de délai $> 1\%$	OP	33%	34%	34%	34%	22%	22%	30%	32%
Nombre de	40 %	0	0	116	162	0	0	6	381
paquets	60 %	0	0	487	669	0	0	6	605
hors séquence	80 %	0	0	5445	6066	0	0	8	3603
	OP	0	0	127	106	0	0	2	403

Tableau 1. *Topologie aléatoire 1, panne du nœud 1, mesures diverses*

4.3. *Topologie UUnet : panne du nœud "New-York"*

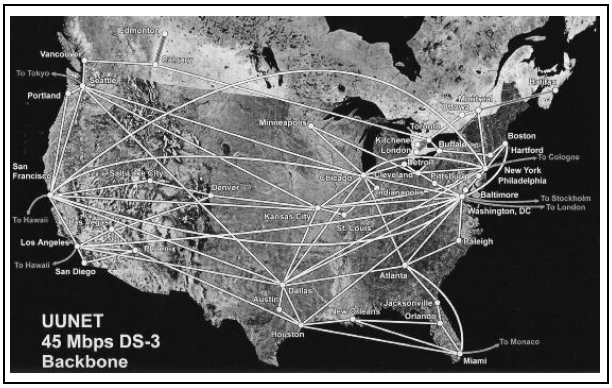

Figure 7. *La topologie du réseau UUnet USA*

Dans ce scénario, les sources de trafic sont de type TCP. Contrairement au cas des sources Poissoniennes utilisées dans le cas du protocole UDP, un seul flux TCP ne peut modéliser à lui seul le comportement de plusieurs connexions TCP. Nous avons donc simulé dix connexions TCP entre toute paire de nœuds terminaux. Chaque ville représentée sur la figure 7 s'étant vue attachée un tel nœud, plus de 16000 connexions TCP simultanées ont été simulées. Si ce nombre reste relativement petit en comparaison de ce que l'on peut rencontrer dans un réseau réel de cette ampleur, des simulations ont montré qu'augmenter ce nombre ne modifiait pas significativement les résultats.

La figure 8 présente l'évolution temporelle, après une panne, du nombre de LSP de backup établis. L'abscisse 0 représente le moment de la panne. Les temps de détection de la panne et de calcul des nouveaux chemins ne sont pas pris en compte. Les deux algorithmes E et E+LEN combinés avec l'option LHP, qui accorde une priorité maximum aux paquets LDP, affichent les meilleurs résultats : plus de 95% des LSPs sont établis après $60\,ms$. On notera que les deux courbes représentant ces deux algorithmes sont pratiquement superposées sur la figure.

Le phénomène déjà observé précédemment, à savoir la pénalité qu'implique l'utilisation de backup locaux sur l'établissement des chemins de backups "de bout-en-bout", est à nouveau présent dans le cas LND. Bien que les messages LDP ne soient plus jamais détruits, ils perdent du temps dans les files d'attente et la signalisation des nouveaux LSP est donc ralentie. Ceci explique le fait que la courbe figurant l'algorithme "E+LND" soit au dessus de celle représentant le même algorithme mais incluant une phase locale (LEN). L'écart entre les deux méthodes est de l'ordre de $10\,ms$.

Dans tous les cas, la restauration est totalement en place après $90\,ms$. Ce temps pourrait encore être réduit en modifiant l'hypothèse choisie par les nœuds ingress lors de leur première tentative d'établissement d'un chemin alternatif. L'option choisie (panne de lien) impose en effet, dans le contexte d'une panne de nœud, deux tentatives d'établissement pour certains LSP.

Un éclairage différent sur l'impact du choix entre méthode de restauration locale et/ou globale est apporté par la figure 9. Celle-ci montre l'évolution du taux de pertes instantané juste après la panne pour différents algorithmes.

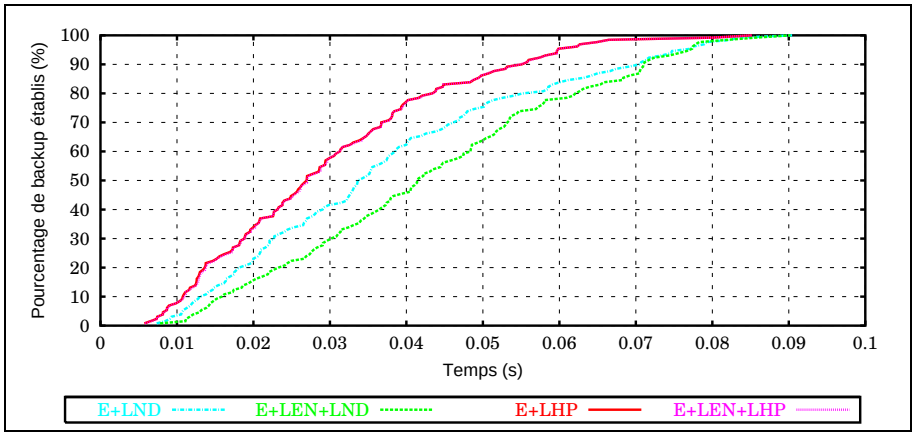

Figure 8. *Topologie UUnet USA, panne du nœud "New-York", temps d'établissement des LSPs de backup*

L'instant de la panne (en $t = 2\,s$) se marque évidemment par un pic de perte à cause des paquets présents sur les liens et dans les files d'attentes du nœud défectueux.

Les algorithmes possédant une phase locale (LEN et E+LEN+LHP) voient ce taux redescendre instantanément après la panne. L'algorithme E+LHP continue à subir des pertes importantes puisqu'aucune action n'a encore été prise. S'agissant de flux TCP, ceux-ci ralentissent en réaction aux pertes ce qui justifie la décroissance. Les phases locales impliquent par contre une congestion importante en $t = 2.015\,s$ puisque les émetteurs ne sont pas informés de celle-ci.

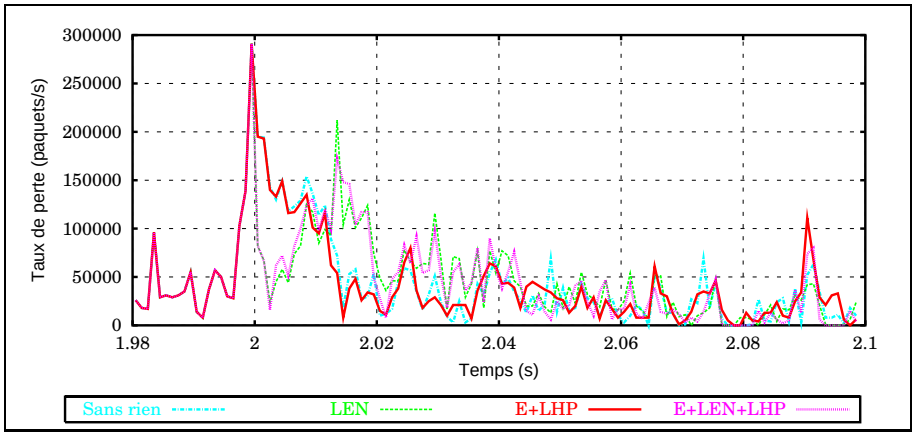

Figure 9. *Topologie UUnet USA, panne du nœud "New-York", évolution du taux de perte après la panne*

Le tableau 2 présente les valeurs mesurées de différents paramètres pendant cette série de simulations.

La haute connectivité du réseau UUnet et le contrôle de congestion de TCP égalise quelque peu les résultats. Cependant les tendances déjà observées sont toujours présentes : les méthodes locales affichent une variation du délai moyen plus importante et la jigue est également à l'avantage des méthodes "de bout-en-bout".

Paramètre	Algorithmes					
	LNN	LEN	E+LND	E+LEN +LND	E+LHP	E+LEN +LHP
Taux de perte	2,621%	2,681%	2,615%	2,625%	2,641%	2,639%
Variation moyenne du délai	7,705%	6,892%	5,785%	5,590%	5,682%	5,778%
Jigue	8,620 ms	8,462 ms	8,319 ms	8,385 ms	8,284 ms	8,339 ms
Nombre de paquets hors séquence	0	0	0	5	0	19
Nombre de paquets transmis	2 991 707	2 990 491	3 000 111	2 994 943	2 993 907	2 994 399

Tableau 2. *Topologie UUnet USA, panne du nœud "New-York", mesures diverses*

5. Conclusions

L'objectif de ce travail était de justifier le choix intuitif que l'on est inévitablement amené à réaliser lorsqu'on désire concevoir un algorithme de restauration rapide. Toute l'étude a été réalisée en optant pour un certain nombre d'hypothèses simplicatrices : routage selon le plus court chemin, trafic sans qualité de service, une seule panne simultanée dans le réseau.

Cependant, un certain nombre de résultats intéressants ont déjà été mis en évidence :

– En fonction de la topologie du réseau, les performances, en terme de taux de perte et de variation du délai, des méthodes globales et locales peuvent être très similaires. Toutefois les algorithmes de bout-en-bout conservent souvent un léger avantage.

– En ce qui concerne le temps de restauration, les algorithmes locaux sont évidemment imbattables. Cependant les résultats montrent qu'il est relativement aisé de rester en dessous de $90\,ms$ de temps restauration dans le cas de méthodes globales. Si on ajoute environs $20 - 25\,ms$ de temps de détection et environ $100\,ms$ de temps de calcul des nouveaux chemins (temps facilement atteignable sur un processeur moderne), l'ensemble du processus de restauration peut être achevé en $200 - 250\,ms$. Pour la majorité des applications, en particulier best-effort, ce résultat sera largement suffisant. Un trafic de type "voix" pourrait presque s'en satisfaire dans beaucoup de

situations même s'il serait plus prudent de traiter cette application par une méthode locale.

– Dans le cas de trafic adaptatifs comme TCP, tenter d'assurer le minimum de pertes possible immédiatement après la panne n'est peut être pas la meilleure stratégie à appliquer car elle se paie par une congestion accrue quelques instants plus tard. Une approche pourrait donc être de détruire "préventivement" un certain nombre de paquets afin de permettre une décongestion et un établissement rapide des chemins de backup.

– Il est important d'accorder une plus haute priorité aux messages de signalisation afin de permettre une stabilisation rapide des routes dans le réseau. Dans un modèle comme Diff-Serv cela signifie qu'il faut prévoir une classe de trafic avec une haute priorité, par exemple EF, pour propager ces messages.

– L'option CA, utilisant les backups inutilisés pour propager des paquets qui auraient été détruits par la congestion en l'absence de ce mécanisme, est un mauvais choix. Elle induit une jigue importante et un grand nombre de paquets arrivent dans le désordre. Toutefois, pour un trafic best-effort dont les contraintes sont extrêmement faibles, cette option pourrait permettre d'utiliser les derniers pourcents de ressources disponibles un peu partout dans le réseau.

– Le dimensionnement "orienté-panne" offre dans la majorité des simulations des performances supérieures dans tous les registres. L'utilisation moyenne du réseau n'est toutefois pas trop détériorée par cette approche.

En conclusion, ce travail se veut un éclairage différent sur une étape souvent négligée du design d'un algorithme de protection rapide. Cet objectif nous semble être atteint. Les résultats sont cependant loin de mettre en évidence une technique surclassant largement les autres et il s'agira d'étudier la technique de restauration la plus adaptée au réseau et à la nature du trafic à protéger.

6. Travaux futurs

Se basant sur l'éclairage fourni par ce premier travail, de nombreuses pistes restent à explorer en intégrant cette fois la notion de réservation de ressources en vue de satisfaire des contraintes de qualité de service.

Les performances du réseau orienté-panne sont obtenues en réalisant une ingénierie du réseau une fois la matrice de trafic et les routes connues. Le problème réel possède comme données la topologie, la capacité de chaque lien et une matrice de charge tandis que la solution est un ensemble de routes minimisant les pertes en cas de panne. Les travaux futurs consisteront donc à rechercher des algorithmes permettant d'atteindre les performances du dimensionnement orienté-panne mais dans les conditions du problème réel.

7. Remerciements

Cette recherche a été partiellement financée par la Commission Européenne dans le cadre du projet "ATRIUM" (IST-1999-20675).

8. Bibliographie

[ABO 01] ABOUL-MAGD O., ANDERSON L., ASHWOOD-SMITH P., HELDSTRAND F., SUNDELL K., CALLON R., DANTU R., DOOLAN P., WORSTER T., FELDMAN N., FREDETTE A., GIRISH M., GRAY E., HALPERN J., HEINANEN J., KILTY T., MALIS A., VAANANEN P., WU L., « Constraint-Based LSP Setup using LDP, draft-ietf-mpls-cr-ldp-06.txt », Internet-Draft, November 2001.

[AND 01] ANDERSON L., DOOLAN P., FELDMAN N., FREDETTE A., THOMAS B., « LDP specification, RFC3036 », www.ietf.org, January 2001.

[BRE] BREMLER-BARR A., AFEK Y., KAPLAN H., COHEN E., MERRITT M., « Restoration by Path Concatenation : Fast Recovery of MPLS Paths », citeseer.nj.nec.com/374608.html.

[IWA 01] IWATA A., FUJITA N., NISHIDA T., « MPLS Signaling Extensions for Shared Fast Rerouting, draft-iwata-mpls-shared-fastreroute-00.txt », Internet-Draft, July 2001.

[KOD 00] KODIALAM M. S., LAKSHMAN T. V., « Dynamic Routing of Bandwidth Guaranteed Tunnels with Restoration », *INFOCOM (2)*, 2000, p. 902-911.

[KOD 01] KODIALAM M., LAKSHMAN T., « Dynamic Routing of Locally Restorable Bandwidth Guaranteed Tunnels using Aggregated Link Usage Information », *INFOCOM 2001. Proceedings. IEEE*, vol. 1, 2001.

[MUR] MURALI K. K., « Routing Restorable Bandwidth Guaranteed Connections using Maximum 2-Route Flows », citeseer.nj.nec.com/463198.html.

[OWE 01a] OWENS K., SHARMA V., MAKAM S., MACK-CRANE B., HUANG C., AKYOL B., « Extensions to CRLDP for MPLS Path Protection, draft-owens-crldp-path-protection-ext-01.txt », Internet-Draft, July 2001.

[OWE 01b] OWENS K., SHARMA V., OOMMEN M., « Network Survivability Considerations for Traffic Engineered IP Networks, draft-owens-te-network-survivability-01.txt », Internet-Draft, July 2001.

[ROS 01] ROSEN E. C., VISWANATHAN A., CALLON R., « Multiprotocol Label Switching Architecture, RFC3031 », www.ietf.org, January 2001.

[STA 00] STAMATELAKIS D., GROVER W. D., « IP Layer Restoration and Network Planning Based on Virtual Protection Cycles », *IEEE Journal on selected areas in communications*, vol. 18, n° 10, 2000.

[Wei 01] WEIDONG CUI, « Efficient Bandwidth Allocation for Backup Paths », citeseer.nj.nec.com/473274.html, 2001.