Architecture d'une boîte à outils d'algorithmes d'ingénierie de trafic et application au réseau GÉANT

Fabian Skivée — Simon Balon [1] **— Olivier Delcourt — Jean Lepropre — Guy Leduc**

Research Unit in Networking
Université de Liège, Institut Montefiore, B28
4000 Liège, Belgique

{skivee,balon,delcourt,lepropre,leduc}@run.montefiore.ulg.ac.be

RÉSUMÉ. *Cet article présente l'architecture logicielle d'une boîte à outils d'ingénierie de trafic* [2]*.Les méthodes déjà présentes couvrent l'ingénierie de trafic intra-domaine basée sur MPLS, mais devraient bientôt couvrir l'ingénierie de trafic inter-domaine, ainsi que celle basée sur IP. L'objectif de cette boîte à outils est de fournir un logiciel libre permettant à un opérateur de tester des outils issus de la recherche académique et à un chercheur de comparer et promouvoir ses algorithmes d'ingénierie de trafic. La boîte à outils est conçue pour être déployée comme un outil on-line dans un environnement opérationnel, ou comme un outil d'optimisation off-line ou encore comme un simulateur d'ingénierie de trafic. Nous avons réalisé une étude de cas d'optimisation de trafic sur le réseau GÉANT. Cette étude présente une comparaison de différentes méthodes de routage, une évaluation du coût d'une protection de bout en bout ou locale ainsi que l'analyse de la pire panne de lien.*

ABSTRACT. *This paper presents the software architecture of a traffic engineering toolbox. The methods already included cover MPLS-based intra-domain traffic engineering (TE), but will soon cover IP-based TE and inter-domain TE. This toolbox provides an open source software that allows an operator to test methods coming from the academic research. A researcher can also use the toolbox for comparing and promoting his/her TE algorithms. The toolbox is designed to be deployed as an on-line tool in a operational network, or as a traffic engineering simulator. We present a case study on the GÉANT network that compares different routing algorithms, evaluates different protection cost, and analyses the worst-case link failure.*

MOTS-CLÉS : *Boîte à outils, Ingénierie de trafic, TE, MPLS, IP, réseau GÉANT, DAMOTE*

KEYWORDS: *Toolbox, Traffic Engineering, TE, MPLS, IP, GÉANT network, DAMOTE*

1. Aspirant du Fonds National belge de la Recherche Scientifique (FNRS)
2. Disponible à l'adresse http ://totem.info.ucl.ac.be

1re soumission à *CFIP05*, le 21 janvier 2005.

1. Introduction

Encore aujourd'hui, le moyen simple et classique d'offrir un service de qualité pour un réseau d'entreprise ou un fournisseur internet (ISP : Internet Service Provider) est de surdimensionner son réseau. Pour répondre à l'augmentation de la demande et pour atteindre des garanties de service (SLA : Service Level Agreement), cette approche est de moins en moins viable économiquement. L'alternative est de déployer des techniques d'ingénierie de trafic. Cependant, la plupart des problèmes rencontrés dans ce domaine sont combinatoires et de grande taille, ce qui implique de trouver des heuristiques efficaces et (quasi) optimales.

Beaucoup de recherches ont déjà été effectuées par le monde académique et de bonnes solutions sont actuellement disponibles. Cependant, peu d'entre elles sont réellement utilisées par les opérateurs pour gérer leur réseau. Une des raisons est la complexité de ces méthodes qui sont implémentées spécifiquement dans un objectif de recherche et de simulation. Ces méthodes s'intègrent difficilement dans un environnement opérationnel. L'objectif principal de notre boîte à outils est de réconcilier le monde académique et le monde opérationnel en fournissant des interfaces interopérables et conviviales avec d'autres outils existants. Cette boîte à outils servira aussi au chercheur qui désire tester, comparer et promouvoir ses recherches.

La conception de la boîte à outils permet différentes utilisations. Celle-ci peut être déployée comme un outil on-line dans un réseau opérationnel, ou être utilisée comme outil off-line ou encore comme simulateur d'ingénierie de trafic. De plus, une grande variété de méthodes d'ingénierie de trafic peuvent être intégrées. Ces méthodes peuvent être classées selon différents axes : intra-domaine ou inter-domaine, on-line ou off-line, IP ou MPLS [ROS 01] (Multiprotocol Label Switching), centralisée ou distribuée.

Le papier est structuré comme suit. Dans la section 2, nous décrirons les formats de données que nous avons définis. Dans la section 3, nous présenterons l'architecture de la boîte à outils ainsi que les exigences qui ont guidé son développement. Dans la section 4, nous présenterons brièvement les algorithmes déjà intégrés. Enfin, la section 5 présentera une étude de cas réalisée avec la boîte à outils sur le réseau européen de la recherche GÉANT.

2. Standardisation des informations

Cette section présente les trois formats XML que nous avons définis dans le cadre de la boîte à outils. Nous avons choisi d'utiliser la technologie XML car elle nous permet d'être interopérables, mais également car il existe beaucoup d'outils pour cette technologie.

Topologies Un aspect commun aux méthodes d'ingénierie de trafic est qu'elles utilisent des topologies en entrée et/ou en sortie. Tous les algorithmes de la boîte à outils n'utilisent pas les mêmes informations. Ainsi, le format défini est flexible

dans le sens où il peut être étendu et presque tous les attributs et éléments sont optionnels. L'élément racine du schéma est *domain*. Celui-ci contient cinq sous-éléments : *info* (auteur de la topologie, date, etc.), *topology* (informations concernant la topologie vue au niveau de la couche réseau), *mpls* (liste d'éléments *lsp* (Label Switch Path)), *igp* (Interior Gateway Protocol) (informations concernant le protocole de routage intra-domaine) et *bgp* (Border Gateway Protocol) (informations concernant le protocole de routage inter-domaine).

Matrices de trafic Ces matrices contiennent la quantité de trafic échangée entre chaque paire de nœuds. Le format consiste en une liste de nœuds "source", chacun contenant une liste de nœuds "destination". Enfin, pour chaque nœud destination, on spécifie la quantité de trafic échangé.

Scénarios Les fichiers de scénarios ont pour objet de rassembler les informations concernant des événements devant avoir lieu sur le réseau. Typiquement, le format défini est constitué d'une liste d'événements tels que la création d'un LSP (primaire ou de backup), la suppression d'un LSP, la panne d'un lien, d'un nœud ou d'une interface, la (re)mise en fonction d'un lien, d'un nœud, etc

3. Architecture logicielle

La boîte à outils consiste principalement en un répertoire de méthodes d'ingénierie de trafic. Ce répertoire permet d'effectuer facilement des comparaisons entre algorithmes et aussi de sélectionner un ensemble de méthodes pertinentes à activer dans un environnement réel.

Tout en restant performante, notre architecture logicielle doit satisfaire plusieurs objectifs :

– minimiser l'effort d'intégration d'un nouvel algorithme ;

– proposer des interfaces interopérables pour s'intégrer avec d'autres outils existants comme un simulateur, un outil de gestion de réseaux (NMS : Network Management System) ou un outil de configuration (Provisioning tool) ;

– proposer différents modes d'exécution : on-line/off-line et centralisé/décentralisé ;

– permettre de faire coopérer des algorithmes écrits dans différents langages.

Nous allons commencer par décrire l'intégration de la boîte à outils avec d'autres outils et ensuite décrire l'organisation interne de celle-ci.

3.1. *Intégration avec des outils existants*

Les méthodes d'ingénierie de trafic ont besoin d'une série de données sur le réseau pour pouvoir l'optimiser. Ces informations peuvent être la topologie, la matrice de trafic, la charge des liens, la configuration des LSPs dans le cas d'un réseau MPLS, le

délai des liens, etc. Différents outils sont déployés dans un réseau opérationnel pour collecter ces informations.

Une première catégorie d'outils est appelée système de gestion de réseaux (NMS). Ces outils surveillent et mesurent différentes informations sur le réseau comme la topologie, la matrice de trafic, des informations de routage, etc. Ces outils doivent fournir l'ensemble des données permettant de gérer au mieux un réseau et ceci de manière efficace et précise.

Une deuxième catégorie comprend les outils de configuration qui permettent d'effectuer des actions telles que le changement d'un poids IGP, l'établissement d'un LSP, etc. Ces outils fournissent une interface simple et conviviale pour la configuration d'un réseau.

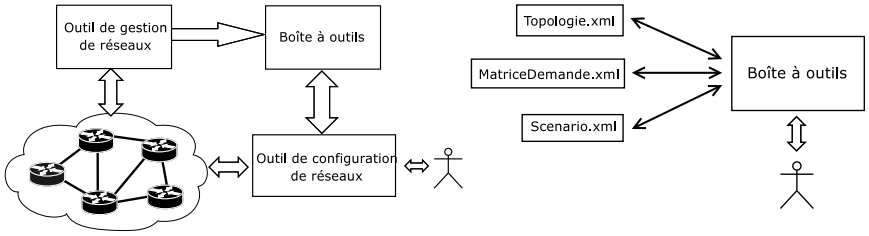

Figure 1. *(a) Intégration dans un environnement réel (b) Intégration dans un environnement de simulation*

Les données nécessaires à l'utilisation des méthodes d'ingénierie de trafic sont communiquées à la boîte à outils différemment selon le mode d'exécution de celle-ci. Nous avons distingué les modes suivants :

– intégration dans un environnement réel (Figure 1a) : la boîte à outils récupère les informations du réseau via un outil de gestion en place, elle exécute différents calculs à la demande de l'outil de configuration de réseaux (p. ex. calculer une route pour un LSP) et répond par une liste d'actions que l'outil de configuration effectuera sur le réseau (p. ex. établir un LSP sur une certaine route et rerouter un autre LSP). Ce mode peut aussi se contenter de fournir les informations à un administrateur de réseau sans modifier la configuration du réseau.

– intégration dans un environnement de simulation (Figure 1b) : la boîte à outils utilise des fichiers (mesurés ou générés) décrivant la topologie, un scénario d'exécution et/ou la matrice de trafic.

Certains outils commerciaux combinent plusieurs modules de la Figure 1a (p. ex . TSOM (Alcatel)[1] Tunnel Builder Pro (Cisco) [2] ou MATE (Cariden) [3]) mais nous avons

1. www.alcatel.com
2. http ://www.cisco.com/en/US/products/sw/netmgtsw/ps4731/
3. www.cariden.com

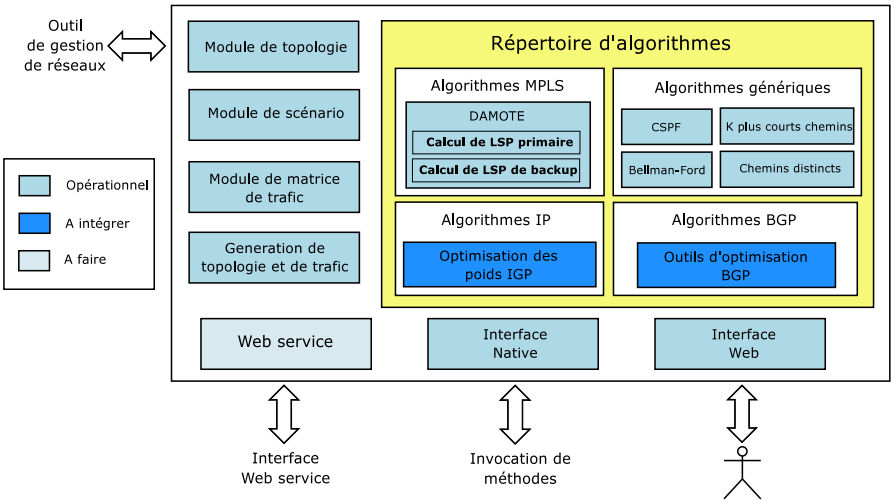

Figure 2. *Architecture de la boîte à outils*

opté pour une séparation claire des objectifs et nous nous concentrons principalement sur la partie ingénierie de trafic.

3.2. *Cœur de l'architecture*

Le cœur de l'architecture de la boîte à outils est le répertoire d'algorithmes d'ingénierie de trafic (Figure 2) que nous avons divisé en plusieurs catégories :

– algorithmes IP : optimisation des poids IGP

– algorithmes MPLS : calcul de LSPs primaires ou de backup

– algorithmes BGP : redistribution de trafic ou simulation de comportement BGP ;

– algorithmes génériques : algorithmes classiques (p. ex. calcul des K plus courts chemins disjoints) et algorithmes génériques (p. ex. tabu search ou algorithme génétique).

Nous avons choisi de développer la boîte à outils en Java pour la facilité de développement et le grand nombre d'outils de traitement du XML fournis autour de ce langage comme : JAXB [4], Services Web, etc. De plus, la technologie JNI[5] (Java Native Interface) nous permet d'intégrer dans la boîte à outils des algorithmes écrits en C ou en C++.

4. http ://java.sun.com/xml/jaxb/
5. http ://java.sun.com/j2se/1.4.2/docs/guide/jni

Afin de rendre le développement de la boîte à outils plus aisé, nous l'avons divisée en modules :

Module de topologies : regroupe l'ensemble des classes relatives à la topologie du réseau qui permettent entre autres d'ajouter/retirer des liens, d'ajouter/retirer des LSPs, de vérifier certaines conditions comme la connexité, d'obtenir des statistiques comme l'utilisation, le débit ou l'équité ;

Module de scénarios : regroupe l'ensemble des classes relatives aux scénarios de simulation permettant de lire, exécuter et générer des scénarios ;

Module de matrices de trafic : regroupe les fonctionalités liées aux matrices de trafic comme la lecture, la vérification de la cohérence avec la capacité des liens et la génération de matrices.

3.2.1. *Intégration d'un algorithme*

Pour être intégré dans la boîte à outils, tout algorithme doit implémenter deux méthodes appelées *start* et *stop*. La première méthode est utilisée pour instancier l'algorithme, le configurer et lui envoyer toutes les informations relatives à l'état courant de la topologie. La seconde méthode est utilisée pour clôturer l'exécution de l'algorithme. Selon le type d'algorithmes, des méthodes additionnelles doivent être implémentées. Par exemple, pour des algorithmes de routage de LSPs MPLS, une méthode *route* doit être implémentée, qui correspond au routage d'un LSP.

Cette méthode est susceptible de renvoyer des informations (typiquement le chemin du LSP calculé). Nous avons décidé d'implémenter le concept générique d'actions. Tout retour d'informations se fait sous la forme d'une liste d'actions. Ce procédé a l'avantage d'être fort générique et flexible. Par exemple, la méthode *route* est susceptible de renvoyer une action *addLSP*. La classe correspondant à cette action comporte une méthode *execute* qui a pour impact d'ajouter ce LSP à la topologie. Dès lors, pour intégrer un nouvel algorithme de routage de LSPs écrit en Java, seules ces trois méthodes (*start*, *stop* et *route*) et les actions susceptibles d'être retournées par la méthode *route* doivent être implémentées.

Notre algorithme de routage de LSPs primaires et de backup DAMOTE (décrit dans la section 4.2 et développé avant d'avoir conçu la boîte à outils) a été intégré via JNI et nous avons mesuré une très faible dégradation (2%) des performances de l'algorithme ainsi intégré.

3.2.2. *Interface web*

Pour permettre une utilisation aisée et conviviale de la boîte à outils, nous l'avons pourvue d'une interface web réalisée grâce à la technologie JSP et au serveur Tomcat[6].

L'interface web est organisée comme suit : l'utilisateur commence par choisir une topologie. Les nœuds, liens et LSPs présents s'affichent alors ainsi qu'une représentation graphique de la topologie. Il peut alors ajouter des LSPs à l'aide de DAMOTE

6. http ://jakarta.apache.org/tomcat/index.html

[BLA 03b] ou à l'aide de l'algorithme CSPF. L'interface web permet aussi, en utilisant Topgen, de générer des topologies et du trafic sur ces topologies (cf. section 4.3).

4. Algorithmes de TE intégrés

4.1. *Algorithmes classiques*

Nous avons intégré plusieurs algorithmes couramment utilisés dans les réseaux. Ces algorithmes peuvent être utilisés comme point de comparaison ou pour développer des méthodes plus complexes.

Une fonction de base fréquemment utilisée dans les réseaux est de trouver le chemin le plus court, de coût le plus faible ou de poids minimum entre des nœuds d'un réseau. Nous avons implémenté différents algorithmes de calcul de plus courts chemins (pour plus d'informations [GRO 03] [BHA 99]) :

– Dijkstra : algorithme le plus couramment utilisé par les routeurs IP pour calculer le plus court chemin vers une ou toutes les destinations. Cet algorithme est implémenté en utilisant une file de priorité sous forme de tas binaire (Binary heap) ce qui permet d'obtenir une complexité de l'ordre de $\mathcal{O}((V + E) \log V)$ où V est le nombre de nœuds et E le nombre de liens.

– CSPF (Constraint Shortest Path First) : cet algorithme est un Dijkstra qui vérifie que chaque lien a une capacité suffisante pour router une certaine demande. Cet algorithme est utilisé pour calculer des LSPs qui réservent une certaine bande passante sur leur chemin.

– Bellman-Ford : cet algorithme de plus courts chemins autorise des poids de liens négatifs mais implique le calcul des plus courts chemins vers toutes les destinations. Il possède une complexité $\mathcal{O}(VE)$ ce qui le rend beaucoup moins efficace que l'algorithme de Dijkstra.

– K plus courts chemins disjoints : [BHA 99] propose un algorithme très rapide pour calculer K plus courts chemins dans un graphe. Cet algorithme peut par exemple être utilisé pour calculer une paire de chemins disjoints et ainsi obtenir un chemin primaire et un chemin de backup disjoints.

– tous les chemins distincts : cet algorithme calcule tous les chemins distincts entre deux nœuds. On peut aussi donner une limite sur la taille des chemins. Par exemple, calculer tous les chemins distincts de moins de 5 nœuds.

4.2. *DAMOTE*

DAMOTE [BLA 03b, BLA 03a] (Decentralized Agent for MPLS Online Traffic Engineering) est un algorithme de routage de LSPs sous contraintes. DAMOTE est bien plus évolué qu'un simple algorithme CSPF. Alors qu'un CSPF est un SPF sur une topologie "élaguée" (obtenue en écartant les liens qui n'ont pas suffisamment de

ressources pour accepter le nouvel LSP), DAMOTE peut réaliser des optimisations plus intelligentes basées sur la minimisation d'une fonction de score. Des exemples de telles fonctions sont : l'utilisation de ressources (ce qui mène à un plus court chemin classique), l'équilibrage de charge, l'équilibrage de charge hybride (où les longs détours sont pénalisés), le routage avec prise en compte des préemptions (avec pénalisation des reroutages).

DAMOTE est générique pour plusieurs raisons. Premièrement, la fonction de score est un paramètre de l'algorithme. Deuxièment, des contraintes peuvent être combinées assez librement. Des contraintes typiques sont par exemple la bande passante disponible sur les liens par classe (CT, Class Type) ou les niveaux de préemption. Par exemple, il est possible de spécifier qu'un LSP d'un certain CT peut uniquement être accepté sur un lien s'il y a suffisamment de bande passante non réservée pour ce CT en ne prenant en compte que les ressources réservées par des LSPs de niveaux de préemption inférieurs (LSPs plus prioritaires). Ceci permet de préempter d'autres LSPs moins prioritaires si besoin est. DAMOTE est dans ce cas en mesure de calculer le "meilleur" ensemble de LSPs à préempter.

DAMOTE calcule efficacement une solution quasi optimale, peut travailler avec différentes fonctions de score et types de contraintes et est compatible avec le modèle MAM [Le 04] (Maximum Allocation Model) proposé par le groupe de travail MPLS DiffServ de l'IETF.

DAMOTE permet également le calcul de LSPs de backup ([MéL 03]). Tout LSP primaire est protégé par une série de LSPs de détour, chacun d'eux partant du nœud immédiatement en amont de chaque lien du LSP primaire. Ces LSPs de détour protègent donc le nœud en aval (si possible) ou le lien en aval et fusionnent avec le LSP primaire quelque part entre la ressource protégée et le nœud egress (inclus). Ces LSPs doivent être préétablis pour du reroutage rapide en cas de panne et approvisionnés avec de la bande passante. En termes de consommation de bande passante, ce schéma est uniquement viable si les LSPs de détour peuvent partager la bande passante entre eux (voir Figure 3) ou avec les LSPs primaires (voir Figure 4). Ceci est possible en faisant l'hypothèse d'une panne unique dans le réseau.

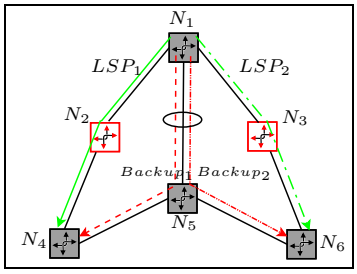

Figure 3. *$Backup_1$ protège LSP_1 de la panne du noeud N_2. $Backup_2$ protège LSP_2 de la panne du noeud N_3. Etant donné que $Backup_1$ et $Backup_2$ ne seront jamais utilisés simultanément, ils peuvent partager la bande passante sur le lien $N_1 - N_5$.*

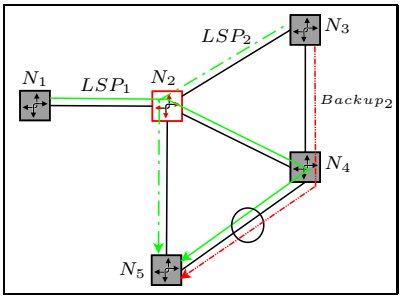

Figure 4. *Les deux LSPs primaires (LSP_1 et LSP_2) tomberont en panne simultanément si N_2 tombe en panne. $Backup_2$, qui protège LSP_2, peut partager de la bande passante avec LSP_1 sur le lien $N4 - N5$, étant donné que $Backup_2$ et LSP_1 n'utiliseront jamais ce lien simultanément. $Backup_1$, qui protège LSP_1, n'est pas présent sur la figure.*

DAMOTE parvient à une protection complète de tous les LSPs primaires contre les pannes de liens et de nœuds avec une surconsommation de ressources de l'ordre de 30 à 70 % des ressources réservées pour les LSPs primaires (selon la topologie). La protection sur SDH/SONET mène quant à elle à 100% de surconsommation sans protéger les nœuds. En ce qui concerne la résistance au facteur d'échelle, si l'ingress de chaque LSP primaire doit calculer tous les LSPs de détour, il a besoin d'une quantité substantielle d'informations à propos de l'état des liens dans le réseau. La solution consistant à inonder le réseau avec cette information grâce à OSPF-TE est possible mais résiste mal au facteur d'échelle. Nous avons proposé une solution ([BAL]) consistant à distribuer le calcul des LSPs de détour entre les nœuds du chemin primaire. L'idée est que tout nœud du chemin primaire calcule le LSP de détour qui le protège (ou qui protège son lien en amont) car il dispose de l'information nécessaire pour effectuer ces calculs. Cette manière de procéder réduit de manière drastique la quantité d'information à faire circuler dans le réseau.

4.3. *Topgen*

Topgen est une application intégrée dans la boîte à outils qui permet de générer des topologies et des matrices de trafic sur ces topologies. Elle permet donc à un chercheur de générer des cas de test pour les algorithmes qu'il développe, et ainsi de valider ou d'invalider une approche d'ingénierie de trafic.

Pour ce qui est de la génération de topologies, Topgen utilise le générateur BRITE [MED 01]. Bien que l'approche prise dans ce dernier est de plus en plus contestée [LI 04], BRITE est actuellement un des meilleurs outils *fonctionnels*. Nous prévoyons cependant d'intégrer de meilleures méthodes de génération de topologies.

En ce qui concerne la génération de trafic, Topgen fournit deux modèles de trafic : un premier basé sur le modèle de gravité et un second qui utilise des distributions de probabilité (distributions uniforme à valeurs entières ou réelles, normale, de Poisson, constante et bimodale). Nous prévoyons également d'intégrer de nouvelles méthodes de génération de matrices de trafic comme celles issues de la tomographie réseau [MED 04, ZHA 03].

5. Etude de cas : le réseau GÉANT

Pour illustrer l'utilisation de la boîte à outils, nous avons réalisé une étude de cas sur le réseau européen de la recherche GÉANT [7]. Le réseau GÉANT est un réseau multi-gigabit, représente 30 pays et connecte 26 réseaux de recherche et d'éducation. Ce réseau contient 23 nœuds et 38 liens.

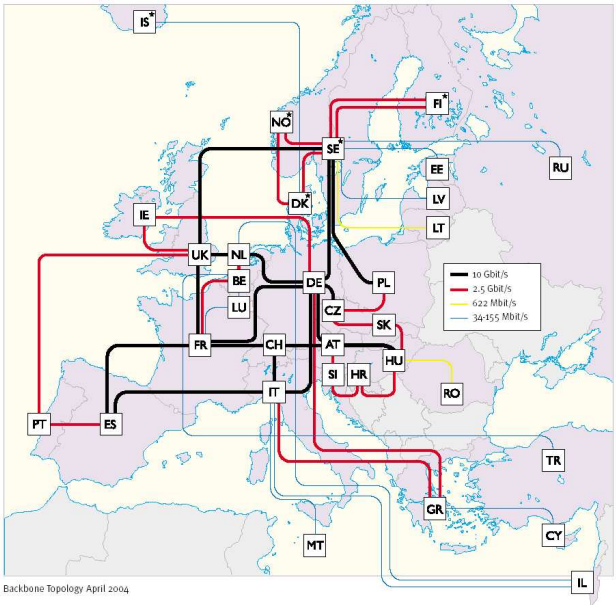

Figure 5. *Réseau GÉANT*

Dans cette section, nous décrirons d'abord la manière d'inférer la matrice de trafic sur le réseau. Nous comparerons le routage proposé par différents algorithmes. Ensuite, nous présenterons le coût d'une protection du réseau en utilisant des LSPs de backup, et enfin nous étudierons l'impact des pannes les plus graves.

7. www.geant.net

5.1. *Inférence de la matrice de trafic*

Pour cette étude de cas, nous avons mis au point une nouvelle méthode de génération de trafic en plus de celles déjà présentes dans Topgen. Nous avons tout d'abord obtenu la charge de chaque interface extérieure du réseau GÉANT (de et vers chaque POP). Ensuite, pour les nœuds qui ont plusieurs interfaces extérieures, nous avons agrégé le trafic entrant (resp. sortant) par nœud. Puis, nous avons construit une matrice de trafic *plausible* en suivant le modèle de gravité décrit dans la section 3.2.1 de [MED 04] : le trafic qui va du nœud i au nœud j est donné par

$$X_{ij} = O_i \frac{T_j^{out}}{\sum_k T_k^{out}} \quad [1]$$

où O_i représente la quantité de trafic injecté dans le réseau par le nœud i et T_j^{out} la quantité de trafic quittant le réseau par le nœud j. Enfin, nous avons vérifié que cette matrice de trafic pouvait être supportée par le réseau.

Pour exécuter nos simulations, nous avons produit une matrice de trafic (nommée TM-max) calculée durant un jour de semaine typique du mois d'octobre 2004 au moment où la somme du trafic entrant et sortant par toutes les interfaces d'accès est maximale.

5.2. *Analyse du trafic*

Dans cette section, nous allons comparer différents algorithmes de routage de LSP grâce à la boîte à outils. Pour comparer ces méthodes, nous allons analyser l'utilisation des liens et plus particulièrement l'utilisation maximale (i.e. le lien le plus utilisé). Cette métrique donne une information sur le goulet du réseau. Nous allons aussi analyser l'utilisation moyenne et son écart type qui reflète l'équilibrage de la charge. Sur une topologie réelle comme celle de GÉANT, les algorithmes donnent des résultats assez différents.

Il faut analyser ces résultats avec précaution pour plusieurs raisons. Premièrement, la matrice de trafic n'est pas réelle mais inférée. L'erreur liée au modèle de gravité est assez grande. Deuxièmement, la configuration actuelle des métriques IGP sur le réseau GÉANT permet de combiner différents objectifs : utilisation préférentielle des liens de grande capacité et contraintes de délai. Dans ces simulations, nous nous intéressons uniquement à réduire la charge du lien le plus chargé et à répartir la charge dans le réseau.

Nous avons comparé les algorithmes de routage suivants :

– MCF (Multi Commodity Flow) dont l'objectif est de minimiser l'utilisation maximale des liens ;

– CSPF qui utilise les métriques réelles du réseau GÉANT ;

– CSPFHopCount qui utilise une métrique de 1 pour chaque lien ;

– CSPFInvCap qui utilise comme métrique l'inverse de la capacité (méthode proposée par Cisco) ;

– CSPFInvFreeBW qui utilise comme métrique l'inverse de la capacité résiduelle ;

– DAMOTE paramétré pour équilibrer la charge.

Dans chaque cas, nous avons établi un maillage complet de LSP (full mesh). Toutes les méthodes, sauf MCF, dépendent de l'ordre d'établissement des différentes demandes. Pour chaque algorithme, après le calcul d'un chemin (LSP), on réserve la demande au niveau de chaque lien. Les résultats pour la matrice TM-max et un ordre aléatoire sont présentés dans le Tableau 1.

Algorithmes	Utilisation maximale	Percentile 10 [8]	Utilisation moyenne	Ecart type
MCF	29,9%	26,9 %	10,4%	9,6 %
CSPF	50,5 %	18,1 %	7,4 %	8,4 %
CSPFHopCount	83,5 %	19,8 %	10,2 %	15,0 %
CSPFInvCap	43,5 %	16,5 %	7,1 %	7,2 %
CSPFInvFreeBW	30,3 %	13,7 %	7,1 %	6,2 %
DAMOTE	30,3 %	14,9 %	8,9 %	5,3 %

Tableau 1. *Comparaison de différents algorithmes de routage sur la matrice de trafic TM-max*

L'utilisation maximale fournie par le multi commodity flow donne la borne minimale que l'on peut atteindre. Cet optimum ne concerne que le lien le plus utilisé car le MCF n'essaie pas de répartir la charge sur les autres liens. On peut constater que les méthodes CSPFInvFreeBW et DAMOTE donnent de très bons résultats, assez proches de l'optimum au niveau de l'utilisation maximale tout en permettant d'obtenir une utilisation moyenne très faible.

L'ingénierie de trafic sur un réseau de petite taille comme GÉANT atteint rapidement ses limites à cause de la configuration de la topologie et du trafic. Vu les coûts importants de connectivité vers certains nœuds (Israël, Grèce, New York, etc.), ceux-ci sont connectés par deux liens de plus faible capacité, ce qui limite considérablement les possibilités de routage. De plus, ces nœuds envoient un trafic important par rapport à la bande passante disponible. Ces liens constituent le goulet inhérent à la configuration du réseau. Les méthodes d'ingénierie de trafic n'ont donc pas énormément de possibilités et doivent se contenter de répartir au mieux la charge sur ces liens. On peut remarquer que DAMOTE et CSPFInvFreeBW excellent sur ce plan.

8. Le percentile N donne l'utilisation du lien choisi de telle sorte que N % des liens du réseau soient plus chargés que lui.

Comme nous l'avons déjà signalé, toutes les méthodes, à l'exception du MCF, dépendent de l'ordre d'établissement. Nous avons évalué les ordres d'établissement suivants : aléatoire, croissant et décroissant. Avec la matrice de trafic utilisée, toutes ces méthodes se montrent relativement insensibles à l'ordre d'établissement. Tout au plus constate-t-on pour CSPFInvFreeBw une légère contre-performance quand les LSPs sont établis en ordre croissant (en commençant donc par les LSPs de plus petites demandes).

Nous avons également évalué l'influence de l'utilisation de plusieurs LSPs pour router une demande. Les méthodes suivantes ont été testées : toute demande est routée par un seul LSP ; toutes les petites demandes (bornées par une valeur x kbit/s) sont routées en un seul LSP, toutes les grosses demandes (dépassant x kbit/s) sont routées en plusieurs LSPs (de taille x sauf pour le dernier LSP résiduel) ; toutes les demandes sont réparties aléatoirement en plusieurs LSPs (de telle sorte que la somme de leur capacité égale la demande).

5.3. *Analyse du coût des backups*

Dans cette section, nous nous attaquons à la protection des flux en cas de panne d'un nœud ou d'un lien du réseau. Nous faisons l'hypothèse qu'une seule panne peut se produire à la fois. Cette hypothèse nous permet de partager la bande passante entre différents LSPs de backup qui protègent des ressources différentes. De plus, nous permettons aux backups de partager également de la bande passante avec des LSPs primaires dans certaines conditions (cfr section 4.2). S'il est impossible de protéger un LSP contre toute panne de nœud, nous protégeons le LSP contre toute panne de lien.

Nous avons adopté la manière de procéder suivante. Nous établissons un LSP primaire unique par demande, et ce de manière séquentielle. L'ordre d'établissement des LSPs peut donc avoir une influence sur la qualité de ceux-ci. Le calcul du LSP primaire s'effectue avec l'algorithme DAMOTE. Signalons qu'un autre choix d'algorithme de routage pour les LSPs primaires pourrait changer les résultats. Ensuite, le calcul des chemins de backup se fait également avec DAMOTE en prenant le chemin primaire comme une contrainte à respecter. Nous avons établi deux sortes de protection : des backups de bout-en-bout (un LSP de backup par LSP primaire) ou des backups locaux (un LSP est établi par nœud ou lien à protéger).

La table 2 nous donne les résultats correspondant à la matrice de trafic TM-max. Les LSPs sont établis dans un ordre aléatoire. Le surcoût en bande passante des backups représente la bande passante totale réservée pour les backups divisée par la bande passante totale réservée pour les primaires. Un surcoût de 100 % signifie donc qu'il y a dans le réseau autant de bande passante réservée pour les backups que pour les primaires. Le partage entre LSPs primaires et LSPs de backups nous permet de réserver respectivement 3 % et 8 % de bande passante en moins dans le cas de la

	protection locale	protection de bout en bout
Surcoût en bande passante	86.8 %	71.7 %
Utilisation maximale	64.8 %	85.7 %
Utilisation moyenne	13.2 %	10.7 %
Nombre de LSP primaires	506	506
Nombre de LSP de backups	1638	506

Tableau 2. *Protection locale vs protection de bout-en-bout. Algorithme de calcul de chemin primaire : DAMOTE.*

restauration locale et de bout en bout. Nous remarquons que les valeurs de surcoût observées ne sont pas très bonnes. En effet, dans l'article [MéL 03], nous obtenions des valeurs de surcoût de l'ordre de 50 %. Les méthodes de restauration offrent de bien meilleures performances sur des réseaux plus larges et plus denses. Effectivement, si on a plus de choix pour le routage des chemins de backup, on pourra choisir un chemin qui partage un maximum de bande passante avec des LSPs déjà établis.

Signalons aussi que pour certains LSPs primaires, les LSPs de bout-en-bout protègent bien contre toute panne de lien, mais pas contre toute panne de nœud, car il était impossible d'éviter un nœud commun avec le LSP primaire.

Analysons maintenant ces résultats. Nous observons que les backups locaux ont de sérieux avantages : ils offrent une restauration plus rapide avec une réservation maximale des liens plus faible. Par contre, les LSPs locaux induisent une réservation moyenne des liens sensiblement plus élevée et évidemment un plus grand nombre de LSPs de backup.

5.4. *Etude de la panne la plus grave*

Nous définissons la panne la plus grave de la façon suivante. Soit S_i l'état du réseau dans lequel le lien i est en panne (S_0 désigne l'état du réseau sans panne). Soit $\mu(S_i)$ l'utilisation maximale dans l'état S_i. On dit que le lien i provoque la panne la plus grave si $\mu(S_i) \geq \mu(S_j), \forall j$.

Comme expliqué ci-dessus, le réseau GÉANT possède une capacité moindre vers le nœud d'Israël (connecté au réseau par deux liens de faible capacité). Il s'ensuit que si un de ces deux liens devient défaillant, tout le trafic à destination d'Israël devra être routé par l'autre lien engendrant ainsi la panne la plus grave. En effet, une quantitié de trafic non négligeable doit atteindre ce nœud et il n'y aura plus qu'un seul lien d'accès à ce nœud. Nous avons dès lors observé que, pour le MCF et les algorithmes de routage CSPF, CSPFInvFreeBW, CSPFInvCap et Damote, la panne la plus grave était due au lien Israël-Italie, et que ces algorithmes de routage donnaient tous les mêmes performances avec une utilisation maximale égale à celle donnée par le MCF (59,8%). Quant à CSPFHopCount, il ne parvient pas à router toutes les demandes. En

effet, lors de la panne de certains liens, l'algorithme fait transiter certaines demandes via des liens de faible capacité (comme les liens connectés à Israël) provoquant une surcharge de ces liens lorsqu'il doit router des demandes destinées aux nœuds que ces liens connectent. Pour cette raison, CSPFHopCount n'est plus repris dans les résultats ci-après.

Pour obtenir des résultats plus intéressants, nous avons considéré la deuxième panne la plus grave (ou la troisième panne la plus grave si la deuxième était causée par la défaillance d'un lien connecté à New-York). Cela mène aux résultats du Tableau 3 pour la matrice TM-max et un ordre d'établissement des LSPs aléatoire.

Algorithmes	Util. maximale	Percentile 10	Util. moyenne	Ecart-type	Lien resp.
MCF	32,9 %	32,4 %	11,6 %	10,5 %	AT-HU
CSPF	55,4 %	18,2 %	7,7 %	8,9 %	NL-UK
CSPFInvFreeBW	37 %	24,4 %	9,3 %	8,4 %	AT-HU
Damote	36,2 %	24,1 %	11,3 %	7,6 %	AT-HU
CSPFInvCap	50,2 %	17,1 %	7,3 %	7,6 %	DE-NL

Tableau 3. *Comparaison de différents algorithmes de routage pour la deuxième panne la plus grave sur la matrice de trafic TM-max*

On remarque que la deuxième panne la plus grave n'est pas toujours causée par le même lien. Enfin, Damote offre un meilleur résultat au niveau de l'utilisation maximale que les autres algorithmes et fait même légèrement mieux que le CSPFInvFreeBW (contrairement au cas sans panne).

6. Conclusion

La boîte à outils proposée fournit une architecture unique pour comparer et valoriser de nouveaux algorithmes. De plus, l'interopérabilité des technologies utilisées permet à un opérateur de rapidement intégrer cette boîte à outils dans son environnement et ainsi tester les méthodes d'ingénierie de trafic les plus modernes. L'étude de cas réalisée présente un aperçu des résultats que l'on peut aisément obtenir avec la boîte à outils actuelle . Grâce au projet TOTEM financé par la région Wallonne et au réseau d'excellence européen E-Next, nous avons déjà beaucoup de collaborations en vue pour augmenter le nombre d'algorithmes, citons notamment le simulateur C-BGP [QUO 03] et l'optimisation de poids IGP [FOR 02].

Remerciements

Cette recherche a été partiellement financée par le programme WIST de la Région Wallonne dans le cadre du projet TOTEM, et s'intègre dans une des thématiques

du réseau d'excellence européen E-NEXT. Nous tenons en outre à remercier Nicolas Simar (DANTE, UK) pour les données qu'il nous a fournies sur le réseau GÉANT.

7. Bibliographie

[BAL] BALON S., MÉLON L., LEDUC G., « A scalable and decentralized fast-rerouting scheme with efficient bandwidth sharing », submitted to Computer Networks.

[BHA 99] BHANDARI R., *Survivable Networks : Algorithms for Diverse Routing*, Kluwer Academic Publishers, 1999.

[BLA 03a] BLANCHY F., MÉLON L., LEDUC G., « An efficient decentralized on-line traffic engineering algorithm for MPLS networks », *In proceedings of 18th International TELETRAFFIC CONGRESS*, Berlin, Germany, September 2003, p. 451-460.

[BLA 03b] BLANCHY F., MÉLON L., LEDUC G., « Routing in a MPLS network featuring preemption mechanisms », *In proceedings of 10th IEEE International Conference on Telecommunications (ICT'2003)*, Papeete - Tahiti, February 2003, p. 253-260.

[FOR 02] FORTZ B., THORUP M., « Optimizing OSPF/IS-IS weights in a changing world », *IEEE Journal in Selected Areas in Communications*, vol. 20, n° 4, 2002, p. 756-767.

[GRO 03] GROVER W., *Mesh-Based Survivable Networks, Options and Strategies for Optical, MPLS, SONET, and ATM Networking*, Prentice Hall PTR, 2003.

[Le 04] LE FAUCHEUR F., LAI W., « Maximum Allocation Bandwidth Constraints Model for Diff-Serv-aware MPLS Traffic Engineering », Internet draft, March 2004, Internet Engineering Task Force.

[LI 04] LI L., ALDERSON D., WILLINGER W., DOYLE J. C., « A First-principles Approach to Understanding the Internet's Router-level Topology », *In Proceedings of ACM Sigcomm'04*, Portland, OR, August 2004.

[MED 01] MEDINA A., LAKHINA A., MATTA I., BYERS J., « BRITE : An Approach to Universal Topology Generation », *In Proceedings of the International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems - MASCOTS '01*, Cincinnati, Ohio, Août 2001.

[MED 04] MEDINA A., SALAMATIAN K., N.TAFT, MATTA I., DIOT C., « A Two-step Statistical Approach for Inferring Network Traffic Demands », rapport n° BUCS-2004-011, March 2004, Boston University.

[MéL 03] MÉLON L., BLANCHY F., LEDUC G., « Decentralized local backup LSP calculation with efficient bandwidth sharing », *In proceedings of 10th IEEE International Conference on Telecommunications (ICT'2003)*, Papeete - Tahiti, February 2003, p. 929-937.

[QUO 03] QUOITIN B., « C-BGP, an efficient BGP simulator. », September 2003, http ://cbgp.info.ucl.ac.be.

[ROS 01] ROSEN E., VISWANATHAN A., CALLON R., « Multiprotocol label switching architecture », RFC n° 3031, January 2001, Internet Engineering Task Force.

[ZHA 03] ZHANG Y., ROUGHAN M., DUFFIELD N., GREENBERG A., « Fast accurate computation of large-scale IP traffic matrices from link loads », *In proceedings of the 2003 ACM SIGMETRICS Conference*, San Diego, USA, Juin 2003.