UNIVERSITÉ DE MONTRÉAL

END-TO-END PATH COMPUTATION SCHEMES FOR TRAFFIC ENGINEERING IN NEXT GENERATION MULTI-DOMAIN NETWORKS

MERAL SHIRAZIPOUR
DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION DU DIPLÔME DE
PHILOSOPHIÆ DOCTOR
(GÉNIE INFORMATIQUE)
JUIN 2010

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

END-TO-END PATH COMPUTATION SCHEMES FOR TRAFFIC ENGINEERING IN NEXT GENERATION MULTI-DOMAIN NETWORKS

présentée par : SHIRAZIPOUR Meral
en vue de l'obtention du diplôme de : Philosophiæ Doctor
a été dûment acceptée par le jury d'examen constitué de :

Mme. NICOLESCU Gabriela, Doct., présidente.
M. PIERRE Samuel, Ph.D., membre et directeur de recherche.
M. QUINTERO Alejandro, Doct., membre.
M. KHENDEK Ferhat, Ph.D., membre.

To Rayan & Shayan...

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Professor Samuel Pierre, my research director, for his guidance and advices that have helped me throughout my research. Without his full support, this work would have never been completed.

I thank the members of the jury for giving me the honour of being part of the examination comity of this thesis. I am also indebted to Professor Antoine Saucier from the Mathematical and Industrial Engineering Department of École Polytechnique de Montréal for accepting in the last minute to be part of the examination comity.

I will be forever grateful for the collaboration of the staff of Ericsson Research Canada. I especially thank Mr. Yves Lemieux for his useful inputs and constant guidance throughout this thesis. I also thank in particular Mr. Frederic Rossi for his comments on the performance evaluation of chapter 3, and Mr. Benoit C. Tremblay for his comments and invaluable inputs regarding the problem treated in chapter 5.

I would also like to thank Professor Steven Chamberland from the Computer and Software Engineering department of École Polytechnique de Montréal for his useful inputs on some last minute optimization clarifications.

I thank as well the PCE Working Group chairs and members for their useful discussions and clarifications regarding PCE standardization.

I also thank Mr. Jim Deleskie and Mr. Nabil Harrabida for sharing their expertise on networking ; I thank as well Dr. Adrian Holzer for his valuable last minute inputs on parts of this work.

To all my colleagues from the department and the associated personnel, I say thank you for the enriching discussions and for the warm atmosphere of the laboratory and university.

Finally, I thank my family, my friends, and Tijs who motivated and encouraged me in one way or another during this long process.

ABSTRACT

With the advent of all-IP Next Generation Networks and the ever increasing Quality of Service (QoS) demands of new real time IP applications, there is a stringent need for mechanisms that allow the end-to-end sustainment of the traffic. QoS requirements are usually a set of network performance indicators that need to be satisfied in order for the IP applications to function properly. Common QoS parameters are the bandwidth, delay, jitter, packet loss and availability. Thus, network operators urgently need to implement solutions enabling them to satisfy the QoS requirements of real time IP applications.

The consensus for QoS provisioning is the application of well defined traffic engineering mechanisms, which consists in optimally routing the traffic using available resources while satisfying QoS and network constraints. This is often achieved by traffic engineered path computation, which is the central focus of this thesis.

Indeed, the QoS performance parameters can be met by carefully choosing a path that has the available bandwidth, offers the acceptable delay and jitter. If bandwidth is reserved along this path, congestion is avoided and the packet loss performance parameter can also be met. Moreover, careful calculation of primary and backup paths allows high availability in case of node or link failure.

Moreover, there is the fact that traffic is usually transported across different administrative networks. Then, there is the detail that networks are multi-layer in nature. Thus, true end-to-end traffic engineering can only be achieved if inter-domain and inter-layer aspects are both considered. To this end, this thesis proposes an overall framework for the end-to-end traffic engineered path computation problem. As discussed below, the framework is subdivided into three separate aspects, all relying on G/MPLS forwarding technology, which enables a controlled routing and the reservation of resources along traffic engineered paths.

The proposals for each aspect are the outcome of extensive literature review which identify existing solutions, if any, and the reasons of their shortcomings or non-existence. This review limits the direction to be taken to find a solution, often by using existing standards and protocols. This is extremely important given the fact that the research topic of this thesis is closely tied to problems of near future generation networks. Thus, it is crucial to reuse existing methods and standards as much as possible in order to get the approval of the research community on the proposed solutions. Moreover, each aspect or sub-problem is carefully studied by defining the actual real world dilemmas surrounding it.

Afterward, the sub-problems are solved by complete proposals consisting of distributed traffic engineering schemes, signalling processes, mathematical programs and algorithms. The proposals are then validated analytically, comparatively and through careful testing and simulations.

Accordingly, the first aspect treated in this thesis consists in the definition of a novel inter-domain scheme that allows for the computation of inter-domain traffic engineered paths in a distributed manner among different administrations. The scheme relies on calculation nodes (PCEs) that can cooperatively compute inter-domain paths. The proposed solution respects both scalability and confidentiality requirements of inter-domain scenarios. Moreover, it establishes a pre-reservation procedure that enhances the effectiveness of the scheme with superior path deployment success rates. This is necessary because the time to compute an inter-domain path is usually longer, allowing fluctuations in networks resources. Accordingly, if resources are pre-reserved at computation time, it prevents their vanishing at path deployment time, and thus avoids blockage. The proposed solution is studied analytically and through rigorous simulations. The results prove that the proposed scheme allows for the optimal computation of inter-domain paths, and that the pre-reservation mechanism is beneficial when compared to the method without this mechanism.

The second aspect is the adaptation of the above mentioned distributed scheme into an inter-layer scheme for the consideration of joint multi-layer/multi-domain scenarios. Such scheme is necessary because most inter-layer traffic engineered path computation problems are also part of an inter-domain setting. The proposed scheme is applied within a complete traffic engineering solution. Moreover, the use of traffic demand forecasts for traffic engineering is evaluated. The proposed ideas are analyzed first by a comparative study and then through simulations on real world networks. The results compare the proposed scheme and its variants to current inter-layer methods. The results show that the proposed scheme performs better in terms of overall utilization and path setup time. Moreover, the results concerning the use of traffic forecasts clearly show that the accuracy of these demand predictions is not a factor in their usefulness within the proposed traffic engineering scheme.

The third aspect is the proposal of a novel constraint based shortest path computation algorithm which, for the first time, considers the adaptation capability of GMPLS nodes. The solution is based on a mathematical program. The constraints taken into consideration are specific GMPLS technological and traffic engineering best practice constraints. Indeed, the nesting/un-nesting capability of GMPLS nodes is considered, and the solution not only prioritized them over costly signal conversion, but also assures their correct ordering along the computed path (i.e., solving the parenthesis problem). The results obtained by solving

the mathematical program validate its correctness. Then, the algorithm is simulated on real world networks for a large set of demands. The results undoubtedly prove its worth compared to existing proposals, in particular to a graph transformation method.

Overall, the proposed solutions in this thesis are both innovative and practical. The three sub-problems treated are closely tied but the schemes and algorithms can be applied, together or separately, in near future generation telecommunication networks in order to optimize their performance.

RÉSUMÉ

Avec la venue des réseaux de prochaine génération basés sur le paradigme tout-IP et la demande croissante en qualité de service (QdS) des nouvelles applications temps réel, il existe un besoin imminent pour des mécanismes capables de soutenir le trafic de bout-en-bout. Les requis de QdS sont souvent décrits par les paramètres de bande passante, délai, gigue, perte de paquets et disponibilité. Ainsi, les opérateurs de réseaux ont un besoin imminent de techniques qui leur permettraient de satisfaire les exigences de QdS des nouvelles applications IP.

Le consensus pour subvenir aux exigences de QdS est la pratique de l'ingénierie de trafic. L'ingénierie de trafic consiste à acheminer le trafic de façon optimale en utilisant les ressources disponibles, tout en satisfaisant les contraintes de QdS et celles du réseau. Cela est souvent réalisé en calculant des chemins optimaux par l'ingénierie de trafic, qui constitue l'aspect central de cette thèse.

En effet, les paramètres de performances de QdS peuvent être satisfaits en choisissant avec soin un chemin qui a assez de bande passante disponible et qui offre un délai et une gigue acceptable. Si la bande passante est réservée le long de ce chemin, la congestion peut être évitée et la perte de paquets peut ainsi être éliminée. En outre, le calcul minutieux des chemins principaux et de recours permet une meilleure disponibilité en cas de panne de lien ou de nœud .

De plus, étant donné que le trafic est habituellement transporté à travers différents réseaux administratifs, l'aspect inter-domaine du problème ne peut être négligé. Puis, il y a le fait que les réseaux sont de nature multi-couches. Donc, l'ingénierie de trafic de bout-en-bout ne peut être atteint que si les aspects inter-domaine et inter-couche sont pris en compte. À cette fin, cette thèse propose un cadre complet pour l'aspect calcul de chemin bout-en-bout de l'ingénierie de trafic, divisé en trois volets. Ces volets suivent tous la technologie G/MPLS pour l'acheminement du trafic et la réservation de ressources sur les chemins optimaux calculés.

Les propositions apportées par cette thèse sont la portée d'une revue de littérature extensive qui a servi à identifier les solutions existantes et leurs lacunes. Chacun des trois sous-problèmes est méticuleusement étudié en définissant d'abord les dilemmes entourant le problème dans des situations réalistes. Cette revue a souvent limité la direction à prendre afin de trouver une solution aux problèmes qui respecte les normes et protocoles existants. Cela est extrêmement important étant donné le sujet de recherche de cette thèse qui est étroitement liée aux problèmes de réseaux d'un futur proche. Ainsi, il est crucial de tenir compte

des standards existants autant que possible afin d'obtenir l'approbation de la communauté scientifique pour les solutions proposées.

Ensuite, chaque sous-problème est résolu par une proposition complète constituée de procédés d'ingénierie de trafic, de signalisation, de formulation de programme mathématique et d'algorithmes. Chaque proposition est ensuite validée analytiquement, comparativement et par des tests et des simulations de rigueur.

Ainsi, le premier volet abordé dans cette thèse définit un nouveau mécanisme inter-domaine qui permet le calcul de chemins dans un contexte d'ingénierie de trafic. Ce mécanisme repose sur un système réparti et la communication des nœuds de calcul (PCE) entre les différentes administrations. La solution proposée tient compte de l'évolutivité et les exigences de confidentialité des environnements inter-domaine. En outre, elle établit une procédure de pré-réservation qui renforce l'efficacité de la solution avec de meilleurs taux de réussite lors du déploiement des chemins. En effet, le temps de calculer un chemin inter-domaine est généralement plus long, ce qui donne le temps à la disponibilité des ressources de fluctuer. Ainsi, en pré-réservant les ressources au moment de calculer le chemin leur disponibilité devient assurée au moment du déploiement. Les mécanismes proposés sont étudiés analytiquement et puis évalués par simulation. Les résultats obtenus, comparés aux procédés existants, montrent l'efficacité du mécanisme en termes d'optimalité inter-domaine ainsi que de taux de blocage réduit lors du déploiement des chemins G/MPLS.

Le deuxième volet adapte le mécanisme distribué inter-domaine au cas inter-couche avec la considération que la plupart du temps les cas inter-couche et inter-domaine surviennent simultanément. Le mécanisme proposé est utilisé dans la proposition d'une solution d'ingénierie de trafic inter-couche/inter-domaine complète. En outre, l'utilisation des prévisions de trafic et leur utilité lors de l'ingénierie de trafic est évalué. Les idées proposées sont analysées d'abord par des études analytiques et comparatives, puis par la simulation, et leurs mérites est démontrés en les comparant aux procédés actuels d'ingénierie de trafic inter-couche. La solution proposée par cette thèse donne de meilleures performances en termes d'utilisation des ressources et le temps de déploiement des chemins GMPLS.

Le troisième volet définit un algorithme pionnier de calcul de chemin inter-couche avec contraintes d'adaptation GMPLS et contraintes de bonnes pratiques d'ingénierie de trafic inter-couche. La solution proposée est basée sur un programme mathématique qui est résolu de manière exacte. Cette solution est innovatrice dû au fait qu'elle traite les contraintes d'adaptations des nœuds GMPLS. La solution proposée considère en plus de la conversion, l'encapsulation et désencapsulation des LSPs. Comme bonne pratique d'ingénierie de trafic inter-couche, la solution proposée donne priorité à l'encapsulation versus la conversion qui

est plus exigeante pour le nœud en termes de ressources, et qui cause des pertes de bande passante. Les résultats obtenus par la résolution du programme mathématique proposé valident son exactitude. Puis, les résultats de simulation prouvent les bénéfices de l'algorithme proposé comparé à une solution existante qui repose sur les méthodes de transformation de graphes.

Globalement, les solutions proposées dans cette thèse sont à la fois innovatrices et pratiques. Les trois sous-problèmes traités sont étroitement liées, mais les solutions proposées peuvent être appliquées, en combinaison ou de façon individuelle, dans les réseaux de télécommunication d'un proche avenir.

CONDENSÉ EN FRANÇAIS

MÉCANISMES DE CALCUL DE CHEMINS DE BOUT-EN-BOUT POUR L'INGÉNIERIE DE TRAFIC DANS LES RÉSEAUX MULTI-DOMAINES DE PROCHAINE GÉNÉRATION

Dans le but d'augmenter la rentabilité des infrastructures de réseaux de télécommunications à implémenter ou celles déjà existantes, les opérateurs et les fournisseurs de services se sont donnés comme objectif commun d'offrir différents services et applications à l'échelle mondiale. Parmi les services envisagés figurent la migration des applications téléphoniques sur des accès IP à des coûts compétitifs, l'offre de nouveaux services de communication IP tels que la vidéo téléphonie, la télévision en temps réel, la vidéo à la demande, etc. Ces dernières diffèrent totalement des types d'applications supportées par les réseaux IP actuels. Elles impliquent à la fois son, données, images et animations. Donc, dans un futur rapproché les réseaux IP et l'Internet actuel devront être adaptés de manière à pouvoir supporter en grande partie du trafic multimédia et du trafic à caractère mission critique, en plus du trafic de données qu'ils supportent actuellement. Ce type de trafic en temps réel nécessite une qualité de service (QdS) soutenue de bout-en-bout pendant la durée de la connexion. Or, l'Internet actuel est de nature "meilleur effort" et n'offre aucune garantie sur la QdS.

La tendance actuelle pour soutenir la QdS consiste à faire de l'ingénierie de trafic. L'ingénierie de trafic est une solution de plus en plus populaire pour obtenir un bon rendement du réseau en termes de QdS, tout en optimisant l'utilisation du réseau. Le concept a été introduit pour la première fois par Nakagome et Mori (1973) et, depuis, l'intérêt de la communauté scientifique a été de l'adapter à différents scénarios ou technologies pour des résultats optimaux concernant tant la QdS que l'utilisation des ressources. Les technologies G/MPLS (Rosen *et al.*, 2001 ; Mannie, 2004) ont fait leurs preuves comme outil de base pour les techniques d'ingénierie de trafic, et donc seront considérés dans les solutions proposées par cette thèse.

D'autre part, le trafic sur Internet traverse généralement deux à huit domaines, *Autonomous System* (AS), avant d'atteindre sa destination (Pan, 2002). Or, la plupart des méthodes d'ingénierie de trafic proposées dans la littérature traitent du problème au niveau intra-domaine, c.à.d. au niveau d'un domaine à administration unique. Donc, pour soutenir la QdS de bout-en-bout, l'ingénierie de trafic doit être considérée aussi bien à l'intérieur des

AS qu'à travers les AS. Ainsi, cette thèse traite du problème d'ingénierie de trafic inter-domaine [1]. Elle considère que les techniques d'ingénierie de trafic intra-domaine sont déjà en place. Les défis liés à cette problématique viennent surtout de l'hétérogénéité des opérateurs de réseaux en termes technologique et politique. Pour relever ce défi, il est nécessaire que toutes solutions proposées reposent sur des processus standardisés ou en voie de l'être, entre autre l'architecture *Path Computation Element* (PCE) défini par Farell *et al.* (2006).

D'un autre part, les réseaux de télécommunications sont de nature multi-couches, tant sur un plan technologique que sur un plan service où une couche supérieure est client d'une couche inférieure. De plus, très souvent la relation multi-couches de type client servi par une couche inférieure engendre aussi un aspect inter-domaine, c'est-à-dire que la couche inférieure peut appartenir à une autre administration. Donc, un processus qui considère l'ingénierie de trafic pour le calcul de chemin inter-couche et inter-domaine est requis. Ainsi, cette thèse traite pour la première fois du problème commun d'ingénierie de trafic inter-couche et inter-domaine. La solution est placée dans un cadre complet d'ingénierie de trafic où elle est testée en considérant l'utilisation de la prédiction de trafic.

Les deux volets mentionnés ci-haut considèrent que les algorithmes de calcul de chemin intra-domaine sont en place. En effet, à l'échelle intra-domaine, un chemin optimal doit tenir compte de l'aspect inter-couche des réseaux ainsi que de la technologie en place. Donc, il est nécessaire de s'assurer qu'un tel algorithme existe dans les contextes technologiques considérés par les deux autres volets. Ce défi est levé par le troisième volet de cette thèse.

Cette thèse est organisée comme suit. Le chapitre 1 introduit le contexte, les éléments de la problématique ainsi que les objectifs de recherche. Le chapitre 2 présente la revue de littérature et les travaux de standardisation sur lesquels repose cette thèse. Les chapitres 3 à 5 présentent respectivement les trois volets traités dans cette thèse. Finalement, le chapitre 6 conclut cette thèse en rappelant ses contributions majeures, en présentant ses limitations et en proposant des travaux futurs pertinents.

CHAPITRE 1

INTRODUCTION

Ce chapitre présente les concepts de base nécessaires pour comprendre le sujet et les objectifs de cette thèse.

1. Les termes inter-domaine et multi-domaines ainsi que inter-couche et multi-couches sont utilisés de manière interchangeable dans cette thèse.

Concepts de bases et éléments de la problématique

La QdS est un terme utilisé dans l'Internet pour désigner la capacité du réseau à fournir aux usagers un service en tenant compte de leurs besoins en terme de débit, délai, gigue, perte de paquets, et disponibilité. Ces indices de performance sont détériorés en particulier par la congestion dans le réseau. La congestion provient d'un manque de ressources physiques, causé par une mauvaise planification du réseau, par une panne dans le réseau, ou par un mauvais partage des ressources. Donc, pour offrir la QdS désirée, la congestion doit être détectée et contrôlée. Or, les réseaux IP actuels ne supportent pas la QdS car les protocoles de routage de base qu'ils utilisent ne tiennent pas compte de la congestion.

Clairement, pour soutenir la QdS dans les réseaux de télécommunications, il est nécessaire de recourir à l'ingénierie de trafic. L'ingénierie de trafic est souvent un problème d'optimisation mathématique qui consiste à déterminer comment allouer des ressources du réseau à un ensemble de demandes connues. Ce problème est considéré par les opérateurs à différentes échelles de temps. À long terme, l'ingénierie de trafic sert à l'optimisation du réseau par son dimensionnement, et à court terme elle sert à répondre à des congestions temporaires ou à l'utilisation optimale du réseau en temps réel par le contrôle du trafic.

Objectifs de recherche

L'objectif principal de cette thèse est de proposer un cadre pour soutenir la QdS du trafic de bout-en-bout dans les réseaux de prochaine génération. En particulier, de considérer le problème de calcul de chemin dans un contexte d'ingénierie de trafic, et de proposer les algorithmes et la signalisation nécessaires. Les solutions proposées doivent respecter les protocoles et standards déjà en place. Plus spécifiquement, cette thèse vise à :

1. analyser les solutions existantes de calcul de chemin dans un contexte d'ingénierie de trafic inter-domaine afin d'identifier leurs lacunes ;
2. proposer et évaluer un processus de calcul de chemins optimaux dans un contexte d'ingénierie de trafic inter-domaine ;
3. analyser les solutions existantes de calcul de chemin dans un contexte d'ingénierie de trafic inter-couche avec GMPLS afin d'identifier leurs lacunes ;
4. proposer et évaluer un processus de calcul de chemins optimaux dans un contexte commun d'ingénierie de trafic inter-couche et inter-domaine ;
5. proposer et évaluer un algorithme efficace de calcul de chemins optimaux sous des contraintes multi-couches GMPLS.

CHAPITRE 2

REVUE DE LITTÉRATURE

Ce chapitre présente les principaux travaux reliés à cette thèse.

Le volet inter-domaine

Ingénierie de trafic inter-domaine est étudiée depuis plus d'une décennie, mais dû au problème d'hétérogénéité des réseaux et au manque de consensus, aucune solution qui pourrait soutenir le trafic de bout-en-bout n'existe encore. Les seules techniques d'ingénierie de trafic inter-domaine en pratique sont basées sur le protocole de routage BGP déployé sur l'Internet (Sangli *et al.*, 2006). Les techniques à base de BGP manipulent les attributs de chemins de BGP pour obtenir un certain contrôle sur le trafic inter-domaine. Presque toutes les nouvelles techniques d'ingénierie de trafic actuellement utilisées ou acceptées par la communauté scientifique traitent du niveau intra-domaine. Elles peuvent néanmoins servir d'inspiration pour les techniques d'ingénierie de trafic inter-domaine à venir. Encore mieux, elles peuvent être rehaussées pour permettre leur utilisation dans un environnement inter-domaine. Parmi ces techniques intra-domaine figurent l'ingénierie de trafic basée sur la technologie MPLS et plus récemment GMPLS.

Les réseaux de prochaine génération doivent, par définition, offrir une interopérabilité ainsi qu'une QdS soutenues à l'échelle inter-domaine. Dans ce cadre, l'IETF a proposé l'architecture PCE qui consiste en des nœuds de calcul de chemin et d'un protocole de communication inter-PCE permettant la coopération entre PCE. Les deux propositions majeures de méthodes de calcul de chemin inter-domaine sont le par-domaine (Vasseur *et al.*, 2008) et la procédure *Backward Recursive PCE-based Computation* ou BRPC (Vasseur *et al.*, 2009). La méthode par-domaine ne permet pas de trouver un chemin inter-domaine optimale. Le BRPC, quant à lui, permet ultimement de trouver un chemin inter-domaine optimal. Par contre, la norme précise que la séquence de domaines à traverser et de PCEs est connue d'avance. Aussi, le BRPC souffre d'un temps de réponse considérable, ce qui peut nuire au taux de succès du déploiement des chemins LSPs.

Le volet commun inter-couche/inter-domaine

Pour ce deuxième volet, il faut souligner qu'aucun travail n'a traité du problème commun inter-couche et inter-domaine. Par contre, pour tout mécanisme automatique et dynamique d'ingénierie de trafic de bout-en-bout, l'aspect commun inter-couche/inter-domaine, qui est

un fait très courant, doit être pris en considération. De plus, parmi les travaux cités qui traitent du problème d'ingénierie de trafic inter-couche, la couche inférieure n'est sollicitée que quand la couche supérieure n'a plus de ressource. Aussi, aucun autre travail n'a étudié l'utilisation d'une solution inter-couche dans un contexte complète d'ingénierie de trafic. De plus, la prédiction de trafic est souvent mentionnée pour la pratique de l'ingénierie de trafic. Or, les travaux cités proposent des mécanismes de prédiction sans toutefois étudier leur efficacité et leur utilité au sein d'un mécanisme d'ingénierie de trafic.

Le volet algorithme de calcul de chemin multi-couches

Le calcul de chemin inter-couche sous contraintes est un problème NP-difficile qui n'est souvent pas possible de résoudre de façon exacte (Huang *et al.*, 2006). Dans le cadre de l'ingénierie de trafic inter-couche dans un réseau GMPLS, le calcul de chemin optimal sous contraintes d'adaptation est un problème non résolu dans la littérature. Les travaux qui le considèrent font de nombreuses abstractions des aspects technologiques importants comme la notion d'encapsulation des LSPs d'une couche supérieure dans une couche inférieure. Une des solutions existantes repose sur la méthode de transformation de graphes pour calculer un chemin inter-couche en ne considérant que la conversion pour passer d'une couche technologique à une autre. Or, cette pratique n'est pas bonne d'un point de vue d'ingénierie de trafic car la conversion entre deux couches engendre souvent la perte de bande passante. Par exemple, pour aller de la couche TDM à LSC, une conversion d'un OC3 (TDM) va prendre un OC48 complet (lambda complet). De plus, la conversion de signal est très demandant en terme de ressources matériels des nœuds GMPLS.

CHAPITRE 3

PROCÉDURE DE CALCUL DE CHEMIN INTER-DOMAINE

La technique proposée dans ce chapitre repose sur l'architecture PCE et consiste en deux parties, une qui permet le calcul de chemin optimal de bout-en-bout dans un contexte inter-domaine, et l'autre qui permet un taux de réussite élevé de déploiement des LSPs inter-domaine. Pour cela, les ressources sont pré-réservées au moment du calcul de chemin. Ainsi, quand vient le temps de déploiement du LSP, la disponibilité des ressources est presque garantie.

Méthode inter-domaine proposée

La méthode proposée consiste en l'envoi d'un message *[Path/QoS request]* par un nœud PCC d'un premier domaine, vers son PCE. Si ce PCE voit que la destination du chemin demandé pour le LSP n'est pas sous son administration, il achemine la demande vers les autres PCEs qui le connectent vers l'extérieur. En même temps, il regarde ses ressources internes jusqu'à la limite des autres PCEs et pré-réserve ce qui est nécessaire pour le chemin optimal. Le message *[Path/QoS request]* se propage ainsi jusqu'à atteindre le PCE du réseau du nœud destination. Ce dernier fait ses calculs et pré-réserve les ressources. Ensuite, il répond au PCE qui l'a sollicité par un message *[Path/QoS reply]*. Ces messages réponses retournent au premier PCE qui aura finalement une vue en arbre de tous les meilleurs chemins inter-domaine possibles. Il peut alors choisir l'optimal en envoyant un *[Path/QoS request-confirm]* au PCE destination qui lui répond par un message *[Path/QoS reply-confirm]*. En réalité, les messages *[Path/QoS request]* et *[Path/QoS reply]* peuvent correspondre respectivement aux messages *PCReq* and *PCRep* du protocole de communication inter-PCE, PCEP. Les messages *[Path/QoS request-confirm]* et *[Path/QoS reply-confirm]* peuvent correspondre respectivement aux messages *Path* et *Resv* de RSVP-TE qui déploie le LSP.

Résultats d'analyse et de simulation

La méthode proposée peut garantir de trouver la solution optimale par sa façon d'explorer une grande partie des possibilités de chemins inter-domaine. Aussi, les résultats de simulation avec la pré-réservation montrent que cette technique permet d'avoir un meilleur taux de réussite lors des tentatives de déploiement des LSPs inter-domaine ainsi calculés. Le temps de pré-réservation a été étudié et il s'avère que la meilleure solution est d'utiliser des réservations permanentes avec possibilité d'annulation précoce avec un message de signalisation. Ainsi, le taux d'utilisation des ressources n'est pas affecté, ce qui été attendu a priori.

CHAPITRE 4

INGÉNIERIE DE TRAFIC INTER-COUCHE/INETR-DOMAINE DANS UN CONTEXTE BOUT-EN-BOUT

La technique proposée ici est une extension de celle du chapitre 3 adaptée à un nombre limité de couches. Cette technique repose sur le modèle de recouvrement du plan de contrôle

GMPLS. Chaque couche technologique a son propre PCE qui peut communiquer avec les couches adjacentes (couches client ou couche de service). Ainsi, si une couche appartient à une autre organisation, le même processus inter-domaine répondra aux exigences de confidentialité, de manque de visibilité et d'évolutivité.

Méthode commune inter-couche/inter-domaine

La méthode proposée consiste en l'envoi d'un message *[Path/QoS request]* par un nœud PCC vers son PCE. À des fins de simplification, l'explication suit le cas où le PCE en question, ou d'autres PCEs interrogés à la même couche technologique que la demande, appartiennent à la même organisation. Les couches inférieures peuvent appartenir à différentes organisations. Aussi, la destination pour la demande peut appartenir à une autre organisation, dans ce cas il faut utiliser en plus, la méthode proposée au chapitre 3. Le PCE fait le calcul de chemin au niveau de sa couche, mais avant d'attendre la réponse, il envoie la requête au PCE de la couche inférieure. Ce dernier fait de même jusqu'à la première couche ou jusqu'à une couche administrativement choisie. Chaque couche, de la même manière, fait le calcul en parallèle au niveau de sa propre couche et attend la réponse de la couche inférieure. Quand cette réponse est obtenue par un message *[Path/QoS reply]*, le PCE décide du meilleur chemin et répond au PCE de la couche supérieure de la même manière par un message *[Path/QoS reply]*. La couche initiale aura ainsi la possibilité de comparer différentes possibilités, soit d'utiliser ses ressources disponibles à la même couche, soit de faire déployer une nouvelle connexion à une couche inférieure. Cette technique permet une meilleure utilisation des ressources à long terme, comme démontrée par les simulations. Aussi, elle permet à la requête initiale de demander des chemins physiquement disjoints à des fins de résistance aux pannes (chemin de secours) ou pour des raisons d'administration à base de règles. Aussi, ce chapitre présente un modèle analytique d'estimation de temps de calcul de chemin et de temps de déploiement de LSP inter-couche.

Résultats d'analyse et de simulation

Les résultats d'analyse et de simulations montrent que la proposition de déclencher la couche inférieure même si les ressources sont disponibles permet d'obtenir de meilleurs résultats comparés aux méthodes actuelles. Entre autres, le gain est dans l'utilisation du réseau et surtout dans le temps de déploiement des LSPs. Aussi, l'utilisation des prédictions de trafic est étudiée et les résultats montrent clairement que l'exactitude de ces prédictions ne joue pas un grand rôle dans le résultat final et que des prédictions exactes à 50% donnent d'aussi bons résultats que celles exactes à 100%.

CHAPITRE 5

ALGORITHME DE CALCUL DE CHEMIN INTER-COUCHE AVEC CONSTRAINTES D'ADAPTATION

Le chapitre 2 a montré qu'aucun travail n'a traité du problème de calcul de chemins inter-couche sous contraintes d'adaptation de GMPLS. Les quelques travaux existants ne touchent qu'à une seule de ces contraintes, la conversion. La conversion est le fait de transformer le signal d'une couche technologique en une autre par un nœud hybride GMPLS. Ensuite le signal/trafic est acheminé en utilisant la nouvelle technologie. Bien sûr, il faut assurer qu'avant d'atteindre la destination, le signal/trafic soit reconverti au même type que la demande arrivée au premier nœud. Or, les nœuds GMPLS peuvent aussi, et ont surtout été conçus pour, l'encapsulation de LSP d'une couche dans une autre. L'encapsulation/désencapsulation, contrairement à la conversion consomme beaucoup moins de ressource de traitement dans le nœud. De plus, elle ne subit pas des pertes de bande passante en allant d'une couche technologique à une autre. C'est pourquoi une bonne pratique d'ingénierie de trafic devrait donner priorité à l'encapsulation/désencapsulation sur la conversion. Par contre, l'encapsulation/désencapsulation apporte un nouveau défi, c'est la détermination de l'ordre dans lequel l'encapsulation/désencapsulation se fait. Comme solution, cette thèse a conçu un modèle de programmation mathématique qui peut être résolu de manière exacte.

Algorithme inter-couche pour réseau GMPLS

L'algorithme de calcul de chemin inter-couche avec contraintes GMPLS est divisé en trois parties :

1. obtenir un nombre donné de chemins les plus courts en utilisant une modification de l'algorithme K-plus court chemins sur le graphe normalisé du réseau ;
2. pour chaque chemin parmi les K, appliquer le modèle de programmation mathématique proposé ;
3. comparer la valeur de la fonction objective de chaque chemin et choisir le minimum.

Le modèle de programmation mathématique proposé est un programme en nombres entiers binaires. Normalement, ce genre de problèmes est NP-dur, par contre les cas traités dans cet algorithme sont assez petits pour obtenir des résultats exact.

Résultats numériques et de simulation

Les résultats obtenus d'une part valident le modèle de programmation mathématique proposé qui permet en effet de trouver le bon ordre d'encapsulation et de désencapsulation des LSPs ainsi que d'autres contraintes. D'autre part, quand utilisé dans la simulation de vrais réseaux avec un grand nombre de demandes à traiter, l'algorithme proposé performe beaucoup mieux que l'algorithme existant qui repose sur la transformation de graphes. En effet, l'algorithme proposé performe mieux en terme de blocage et d'utilisation de réseau. De plus, les analyses de la solution proposée ont montré que, pour de bons résultats, il n'est pas nécessaire de vérifier tous les K chemins possibles.

CHAPITRE 6

CONCLUSION

Ce chapitre de conclusion présente les majeures contributions de chaque volet, leurs limitations et les travaux futurs possibles.

Le volet inter-domaine

La contribution majeure est la proposition d'une méthode distribuée qui peut garantir l'optimalité du chemin inter-domaine. De plus, par la pré-réservation de ressources, elle garantit un bon taux de succès lors du déploiement des LSPs. La limitation est que cette méthode souffre des problèmes d'évolutivité. Aussi, elle ne tient pas compte des contraintes reliées aux chemins inter-domaine.

Le volet commun inter-couche/inter-domaine

La contribution majeure est de traiter pour la première fois le problème commun d'ingénierie de trafic inter-couche/inter-domaine. La solution proposée permet d'obtenir le chemin bout-en-bout optimal. Aussi, la technique d'ingénierie de trafic complète proposée utilise cette méthode, en plus de l'utilisation de la prédiction de trafic. La limitation repose sur la validation faite en grande partie avec des simulations. Il serait intéressant de valider cette solution sur un banc d'essai pour obtenir des valeurs plus réaliste sur le temps de déploiement des LSPs. Aussi, les simulations ne considèrent que deux couches. Il aurait été intéressant

d'utiliser plus de couches en validant la méthode. Finalement, comme travail futur il serait intéressant d'utiliser la technique proposée dans un contexte infonuagique.

Le volet algorithme de calcul de chemin multi-couches

La contribution majeure de ce volet est de considérer pour la première fois toutes les contraintes d'adaptation dans un réseau GMPLS. L'algorithme proposé et le modèle mathématique pour l'assignation des actions d'adaptation par nœud sur un chemin est une contribution majeure. La limitation du modèle est qu'elle ne permet pas par exemple l'encapsulation du type STa dans STb, puis la conversion de STb à STc et ensuite la désencapsulation de STc en STa. Mais, d'après les standards et recommandations de l'IETF, ce scénario n'est pas un cas commun. Aussi, comme travail futur, il faudrait considérer l'assignation optimale des valeurs des matrices de coûts.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS AND SYMBOLS

AS	Autonomous System
ASBR	AS Boundary Router
ASON	Automatic Switched Optical Network
BE	Best Effort
BGP	Border Gateway Protocol
BRPC	Backward Recursive PCE-based Computation
BW	Bandwidth
CAC	Connection Admission Control
CAPEX	Capital Expenditures
CSPF	Constrained Shortest Path First
DiffServ	Differentiated Services
EGP	Exterior Gateway Protocol
EID	Endpoint Identifier
EoS	Ethernet over SONET
FEC	Forwarding Equivalence Class
FSC	Fiber Switch Capable
GMPLS	Generalized MultiProtocol Label Switching
H-LSP	Hierarchical LSP
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IGRP	Interior Gateway Routing Protocol
ILM	Incoming Label Map
IntServ	Integrated Services
IP	Internet Protocol
IPTV	Internet Protocol Television
ISC	Interface Switching Capability
ISCD	Interface Switching Capability Descriptor
IS-IS	Intermediate System to Intermediate System
ISP	Internet Service Provider
ITU-T	International Telecommunication Union, Telecommunications Sector
L2SC	Layer 2 Switch Capable
LDP	Label Distribution Protocol
LER	Label Edge Router

LFIB	Label Forwarding Information Base
LISP	Locator/Identification Separation Protocol
LSC	Lambda Switch Capable
LSP	Label Switched Path
LSR	Label Switch Router
MPLS	MultiProtocol Label Switching
NGN	Next Generation Network
NHLFE	Next Hop Label Forwarding Entry
NSFNET	National Science Foundation Network
OAM	Operation And Management
OPEX	Operational Expenditures
OSI	Open System Interconnection
OSPF	Open Shortest Path First
PCC	Path Computation Client
PCE	Path Computation Element
PCEP	PCE Communication Protocol
PCPSO	PCE Path Sequence Object
PDU	Protocol Data Unit
PHB	Per Hop Behavior
PSC	Packet Switch Capable
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RIP	Routing Information Protocol
RLOC	Routing Locator
RSVP	Resource ReserVation Protocol
RSVP-TE	Resource ReserVation Protocol - Traffic Engineering
SDH	Synchronous Digital Hierarchy
SLA	Service Level Agreement
SLS	Service Level Specification
SONET	Synchronous Optical NETworking
SRLG	Shared Link Risk Group
TCP	Transmission Control Protocol
TDM	Time Division Multiplexing
TED	Traffic Engineering Database
TE-LSP	Traffic Engineered LSP
VoIP	Voice over the Internet Protocol

VNT	Virtual Network Topology
VPN	Virtual Private Network
VSPT	Virtual Shortest Path Tree

CHAPTER 1

INTRODUCTION

In order to increase the profitability of existing and future telecommunication network infrastructures, operators and service providers are addressing the common goal of providing ever innovative ubiquitous communication services worldwide. The current development in the telecommunication industry is to evolve towards all Internet Protocol (all-IP) Next Generation Networks (NGNs). An all-IP environment will allow better resource management and interoperability along with a reduction in CAPEX/OPEX costs. Among the foreseen services, one can enumerate the complete migration of fixed and mobile voice services over the Internet Protocol (VoIP), as well as other cost effective all-IP communication services like video telephony, real-time television and video on demand (IPTV), etc. These are totally different from the types of applications currently supported by IP networks. They involve sound, data, images and animations, making them very demanding in bandwidth and very sensitive to network conditions. In the near future, current IP networks and the Internet must be adapted to withstand to a large part multimedia and mission critical traffic, on top of the usual data traffic it currently supports. This type of real-time traffic requires end-to-end supported Quality of Service (QoS), making it vulnerable to network conditions. Irrespective of the attention given by the research communities to QoS problems, today's IP networks remain of "best effort" nature and do not guarantee QoS.

The trend for the support of QoS is to make use of traffic engineering techniques. Traffic engineering principles are subject of ongoing studies since the first time the concept was introduced by Nakagome et Mori (1973). Since, traffic engineering is an ever increasing in popularity solution that promises good network performances in terms of QoS and resource utilization. It consists in the optimal allotment of traffic to network resources. In other words, it consists in assigning the right amount of traffic to the right network resources while satisfying the basic QoS needs in terms of bandwidth, delay, jitter, loss and availability.

One of the main challenges with traffic engineering is to optimally route the traffic to obtain the best allotment of traffic to network resources, while respecting a set of constraints. This process is referred to as constraint shortest path first (CSPF) routing. Constraints are necessary because an optimal route must usually satisfy many technological or policy based requirements. Over the time, the challenge with traffic engineered path computation has been to develop and apply its principles to specific network architectures and technologies.

However, due to the complexities of these problems, existing solutions often make abstraction of the encountered challenges and neglect to consider the end-to-end nature of the problem.

To this end, this thesis proposes an end-to-end solution for the specific problem of traffic engineered path computation in next generation Generalized MultiProtocol Label Switching (GMPLS) networks. It considers the multi-domain and multi-layer nature of the problem. First, the multi-domain aspect is considered due to the fact that IP traffic usually crosses more than one administrative domain before reaching its destination. Second, the multi-layer aspect is considered due to the fact that IP traffic relies on transport networks that are composed of more than one technological layer. More specifically, this thesis first proposes a distributed solution to the inter-domain traffic engineered path computation problem. This solution respects existing inter-domain routing norms and is practical as it builds on top of recently defined standards. The proposed solution is analyzed analytically and then validated through rigorous simulation. In a second part, this thesis addresses the overall end-to-end traffic engineering problem which is both multi-domain and multi-layer in nature. The distributed multi-domain method of part one is adapted to the multi-layer case and applied within a full traffic engineering framework. Moreover, traffic engineering paradigms like prediction are analyzed in this part. The proposed method is analyzed both qualitatively and through rigorous simulation. Finally in a third part, this thesis completes the work by proposing a novel multi-layer path computation algorithm that respects specific constraints of next generation multi-layer networks, which has been neglected by existing works. The proposed algorithm is validated with mathematical results and through simulation.

This first chapter presents the basic concepts related to next generation multi-domain and multi-layer networks. It gives a broad review of the subject and problem statements. Detailed description of existing solutions and proposals are deferred to the next chapter. The present chapter describes the specific challenges tackled in this thesis. Then, it presents the research objectives of the thesis, followed by the methodology plan used to achieve them. This introductory chapter ends by presenting a detailed outline of the remaining chapters.

1.1 Basic Notions and Important Aspects

End-to-end IP QoS challenges in NGN networks are in big part caused by the architectural nature of telecommunication networks. Figure 1.1 gives an abstract view of the global telecommunication ecosystem, which is both multi-domain and multi-layer in nature[1]. An Autonomous System (AS) designates any IP network that connects to another administra-

1. The terms inter-domain and multi-domain as well as inter-layer and multi-layer are used interchangeably in this thesis.

tion's IP network through Border Gateway Protocol (BGP) version 4. Today's global Internet is composed of approximately 45000 ASes (Huston, 2009). ASes connect to each other by what is called peering agreements. Depending on their size, ASes are classified as Tier-1, Tier-2 or Tier-3 networks. Tier-1 networks are the ASes that usually have global connectivity. They peer with other Tier-1s and sell their services to Tier-2 ASes. Tier-2 ASes are large enough to peer with some other Tier-2 ASes and may need to purchase IP connectivity services from Tier-1 and other Tier-2 networks. Tier-3 networks usually purchase IP connectivity from Tier-2s. Tier-3 ASes are often referred to as stub networks. That is all traffic that enters them is destined to them. Tier-1 ASes are usually referred to as transit networks. That is they serve as transit for the traffic destined to Tier-2s and Tier-3s. Tier-2 networks can serve both as stub and transit. Nevertheless, IP traffic crosses in average between two to eight of these ASes before reaching its destination (Pan, 2002). This imposes the need for inter-domain traffic engineering in the quest of end-to-end QoS provisioning.

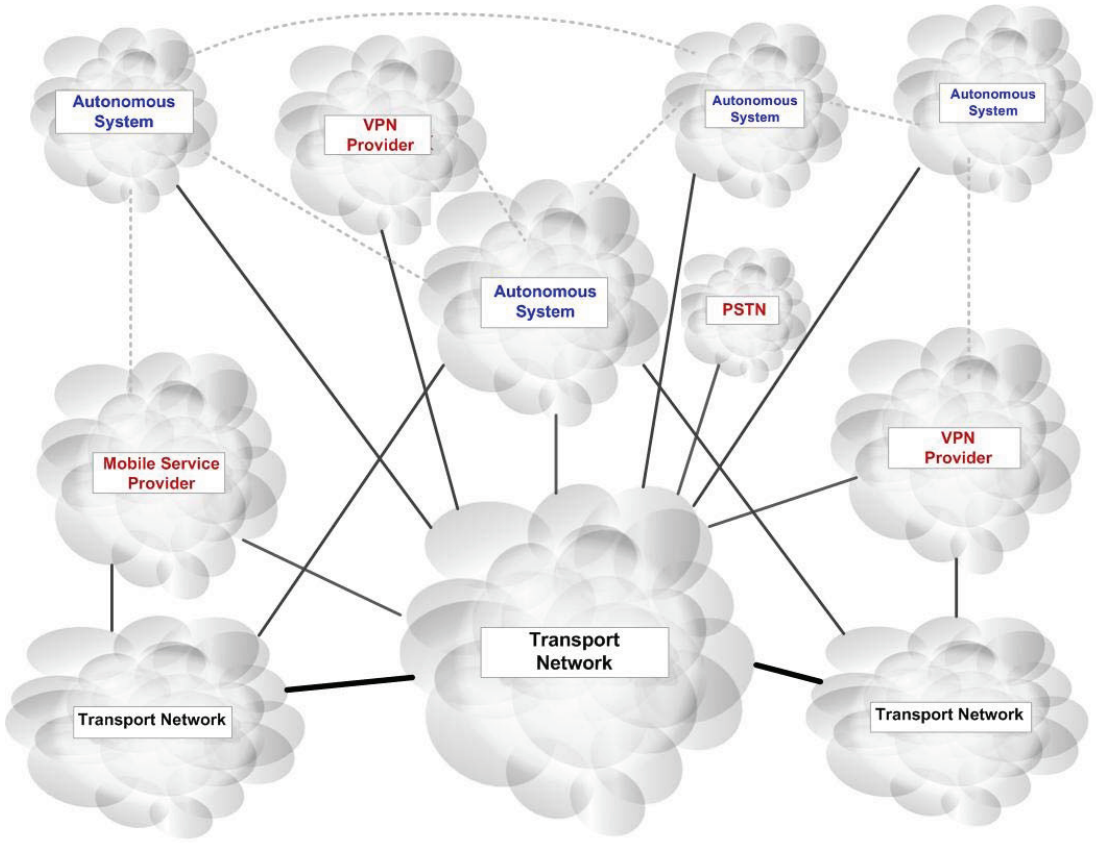


Figure 1.1 Global Telecommunication Ecosystem

Then, all upper layer networks rely on transport networks for connectivity; upper layers being ASes and Internet Service Providers (ISPs), Virtual Private Network providers (VPNs), backhaul for mobile telephony networks, and public switched telephone networks (PSTNs). Transport networks are mainly composed of optical switches and they still mainly use Time Division Multiplexing (TDM) within the Synchronous Optical Networking (SONET) or Synchronous Digital Hierarchy (SDH) standards. Another way to view this is to consider that upper layer networks are clients of the lower layer transport networks. A single transport network can serve multiple higher layer networks; it can also connect and serve as connectivity to other transport networks. The reverse is also possible, that is the higher layer networks can be served by more than one transport networks. Each network layer imposes its traffic to the layer below. Similarly, in the transport network itself, different layers can be identified. These are usually differentiated depending on the 'data rates' or on the technology. The multi-layer data rate problem is often referred to as traffic grooming, which consists in filling lower rate signals into higher rate signals. The multi-layer 'technology' case is often referred to as a multi-region problem where each region corresponds to a different switching technology in GMPLS networks.

Moreover, the overall problem is often a mixture of the multi-domain and multi-layer scenarios described above. This is because the vast majority of higher layer networks do not own at all or completely their transport network. Therefore, they must rely on another organization for lower layer connectivity. This means that for the case of higher layer networks not owning their transport network, the multi-layer solution must also answer inter-domain constraints. This is in fact the case of Tier-3 and most Tier-2 ASes. Therefore, this ecosystem of various networks results in a multi-layer setting where each layer has its own technology with its own types of nodes, links, traffic and perhaps even administration. The complete end-to-end IP QoS problem cannot be discussed without considering the important role of transport networks underneath the IP networks that are the basis of the global telecommunication ecosystem.

The environment crossed by the traffic has been described, but it is important to point out how QoS parameters are affected by the routing, and how path computation can overcome the QoS challenge. The first important QoS parameter, bandwidth can be guaranteed if the traffic takes a route which has enough bandwidth. Better yet, the traffic engineered path can reserve this bandwidth for the traffic, thus guaranteeing this parameter. Then, the delay QoS parameter can be guaranteed if the path takes non-congested links and is not too long in terms of distance. Again, a correctly traffic engineered path can accommodate such criteria. The jitter QoS parameter can be guaranteed by a path that takes non-congested links. Moreover, if all packets take the same path, this will prevent jitter caused by routes having different

delays. The packet loss QoS parameter can be guaranteed in the same manner, by a path that contains highly available non-congested links and nodes. The availability QoS criterion often refers to resiliency issues and the ability to route or re-route the traffic in case of network link or node failure. This is also often achieved by the careful routing of disjoint primary and backup paths.

It is worth mentioning that IP routing is connectionless, that is no prior end-to-end determination of a path is made before routing packets. IP traffic is usually routed using the well known Open Shortest Path First (OSPF), Intermediate System to Intermediate System (IS-IS), and BGP routing protocols. These protocols are implemented in a distributed manner in router nodes and contribute to the best effort nature of IP networks. Therefore, network engineers had to resort to other means in order to control the route taken by the IP packets. This started the success of MultiProtocol Label Switching (MPLS) technology (Rosen *et al.*, 2001). MPLS is a packet forwarding technology that performs label switching between layer 2 and layer 3 protocols in the OSI model. The original purpose of MPLS was faster packet forwarding, which nowadays is achievable by more advanced hardware. Today, MPLS is the technology of choice for traffic engineering and the routing of packets on CSPF paths. MPLS relies on Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) defined by Awduche *et al.* (2001) in order to deploy each Label Switched Path (LSP).

Given the worldwide success of MPLS for QoS provisioning with the routing of IP packets on traffic engineered LSPs [2], this technology was extended to its general form known as GMPLS (Mannie, 2004), and is the CSPF forwarding technology of choice for multi-layer scenarios. Moreover, given this success, G/MPLS has been extended to allow for the routing of CSPF inter-domain paths (Farrel *et al.*, 2008). GMPLS in particular introduces an automated and distributed control plane that is applicable to a variety of technologies called switching layers [3]. The GMPLS control plane allows for automatic resource management, automatic resource discovery, as well as dynamic resource provisioning and recovery. These functionalities of GMPLS matched with the potentials of inter-domain route optimization form a promising duo for obtaining end-to-end QoS guaranties.

The notion of end-to-end in this thesis refers to the routing from the first AS to the destination AS. The traffic considered refers to the aggregation of multiple end-user flows. For user flows' QoS assurance inside the end networks, usually other traffic engineering architectures such as Integrated Services (IntServ) defined by Braden *et al.* (1994) or Differentiated

2. The terms traffic engineered path and traffic engineered LSP (TE-LSP) are used interchangeably in this thesis.

3. In this introductory chapter the terms GMPLS layer is used to designate both multi-layer and multi-region scenarios. These terms will be defined in detail in chapter 2.

Services (DiffServ) defined by Blake *et al.* (1998) are proposed. Thus, this thesis considers path computation from the first network to the destination network, for aggregations of similar traffic flows. Conceptually, for example an ISP could trigger the proposed mechanisms for the deployment of a new LSP from one of its border routers to a border router in the destination network. The path request could specify that the LSP should have a certain guaranteed bandwidth, should not exceed a certain number of transport nodes (for delay assurance), and should be node disjoint from another existing LSP. The proposed mechanisms in this thesis allow for the dynamic computation of such end-to-end LSP route and its successful deployment.

Finally, the proposed solutions fall under the umbrella of the Path Computation Element (PCE) architecture defined by Farell *et al.* (2006). This is a standard proposed by the Internet Engineering Task Force (IETF) that defines PCE nodes, and a TCP based protocol that allows them to communicate with other PCEs in the same or in different domains. The goal is to compute an end-to-end optimal path for the deployment of an inter-domain LSP. A PCE can reside in a router or constitute a separate network node. PCEs receive path computation requests from Path Computation Clients (PCCs). A PCE is considered as a PCC when it requests a path computation from another PCE. The PCE communication protocol (PCEP) defined by Vasseur et LeRoux (2009) is standardized and ensures an efficient interaction between connected PCC and PCE nodes. Thus, in the above example, the ISP's PCC can request the path computation of an LSP from the PCE of its network provider. At this level of granularity, the proposed mechanisms will aim at offering low path request blockage, fast response times and LSP setup times, optimal utilization of resources as well as satisfaction of the constraints present in the request as well as those imposed by the technologies used.

1.2 Motivations and Research Challenges

The discussion above outlined the trend in the industry for end-to-end traffic engineering covering both multi-domain and multi-layer scenarios. However, as it will be reviewed in chapter 2, due to the complexity of the problem, no work has addressed it in a practical and complete manner. The inter-domain traffic engineered path computation problem is challenged both by technical issues and by various policy enforcements. For confidentiality reasons, network operators are not willing to collaborate for a centralized solution. This leaves only the possibility of a distributed solution. However, the optimality of the computed path becomes an issue when the problem is addressed in a distributed manner. The challenges of considering the confidentiality requirements, the scalability issues involved with an inter-domain environment, as well as the optimality of the computed path, are all raised by this

thesis.

Moreover, the issue of joint multi-layer/multi-domain problem has been neglected by existing proposals. This thesis brings light to this difficulty which is imminent in today's telecommunication ecosystem. The multi-layer/multi-domain problem differs from the multi-domain problem as the number of layers crossed by the multi-layer LSP is usually pre-determined. Moreover, even though the PCE architecture is said to be adaptable to inter-domain and to inter-layer path computation, there has been no proposal yet as how to satisfy both situations simultaneously. Thus, the challenge of finding a path computation scheme for the joint multi-layer/multi-domain problem has been raised by this thesis. Moreover, the application of multi-layer traffic engineering is a real world scenario needs to be considered. Path computation schemes are necessary, but they need to be analyzed within realistic settings. The overall effect of using traffic engineering can only be measured in this way. Furthermore, it is interesting to apply and test the path computation scheme alongside other traffic engineering techniques like traffic prediction for example. This is the only way that the true benefits of end-to-end traffic engineered path computation can be measured.

The above described challenges are about defining schemes that will consider the restrictions imposed by multi-domain and multi-layer environments. The end-to-end problem shall be divided and a cooperative scheme shall be proposed. In addition, each subdivision of the problem will need to use a CSPF algorithm. Figure 1.2 presents a general classification of path computation. Non-prunable CSPF path computation is often NP-Hard and requires specific algorithms and heuristics to be solved. As it will be pointed out in chapter 2, the GMPLS multi-layer path computation problem is quite complex and existing works do not cover the actual problem correctly.

The real challenge is to satisfy technological constraints as well as good traffic engineering practices when computing multi-layer paths. For this, a better understanding of the GMPLS technology is required. Section 2.3 will present the problem in details and describe how existing works fail to address the actual issues. Thus, overcoming the inadequacy of existing works on CSPF for GMPLS networks is another challenge raised by this thesis.

1.3 Research Objectives and Scope

Before presenting the research objectives pursued by this thesis, it is essential to clearly define its scope. As previously discussed, this thesis addresses backbone and core networks, as well as traffic aggregates. The traffic engineering procedures discussed are in a time scale of hours and days. This thesis does not consider per flow real-time traffic engineering problems

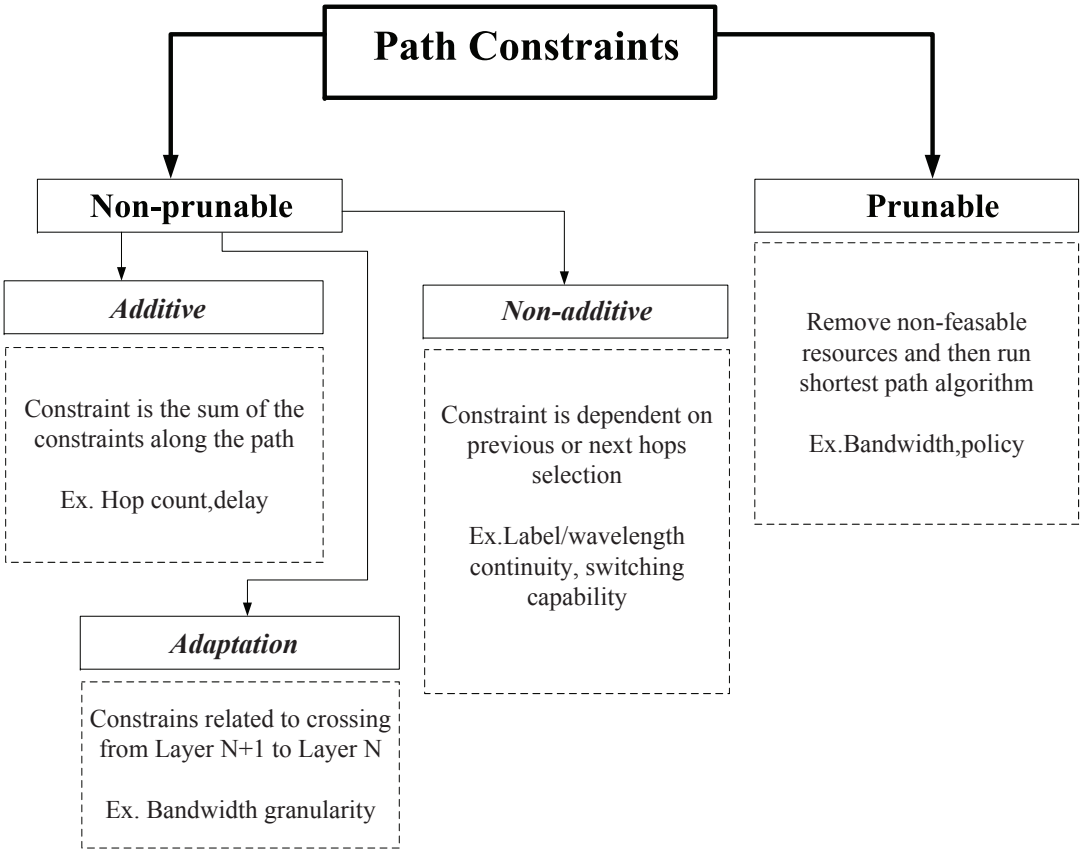


Figure 1.2 Constraint based path computation

(time scale of minutes to hours), which are usually addressed within the access network using traffic engineering principles like IntServ or DiffServ and queuing theory principles.

Thus, this thesis considers optimal path computation for aggregated traffic demands. The path computations are generally initiated for client networks requiring QoS aware connectivity from a source node (e.g. their access router) to a destination node. Given a cost objective function, an optimal path designates the least cost path that respects a given set of constraints. Most importantly, the work in this thesis is aimed at achieving inter-domain/inter-layer reachability in the context of traffic engineering and path computation.

Objectives of the thesis

The main objective of this thesis is to propose a framework for the end-to-end support of the traffic in next generation networks. In particular, to consider the path computation aspect of traffic engineering, along with the necessary algorithms and the corresponding signalling. The proposed framework and algorithms shall respect the protocols already in place and the

PCE architecture standard.

More specifically, the aim of this thesis is to:

1. analyze existing inter-domain traffic engineered path computation solutions and identify their shortcomings;
2. propose and evaluate an inter-domain traffic engineered path computation scheme that allows the computation of optimal inter-domain paths and their successful deployment;
3. analyze existing multi-layer path computation algorithms and identify their shortcomings;
4. propose and evaluate an end-to-end traffic engineering procedure that considers the joint inter-layer/inter-domain nature of the path computation problem;
5. propose and evaluate an effective algorithm for the constraint based multi-layer path computation problem in next generation GMPLS networks.

1.4 Methodological Approach

Research objectives 1 and 3 are achieved by a through and ongoing literature review in all stages of this thesis. After a first complete literature review, scientific research papers are periodically seek in order to keep an up to date perspective on the state of the art. Moreover, given the great interest this research topic has among equipment manufacturers, operators and thus standardization bodies, their work has to also be followed closely through IETF mailing lists, by getting in contact with standardization authors and by bringing small contributions in the forms of inputs or error corrections in the draft documents.

Then, research objective 2 is achieved by addressing the shortcomings identified by objective 1 and proposing a practical solution for the computation of inter-domain traffic engineered paths. The proposed solution shall be based on the PCE architecture standard. The final solution shall be viable and consider other challenges caused by the inter-domain environment, such as the longer path computation time. The proposed solution shall be validated by analysis and simulation on a real world network.

Then, based on the outcomes of objectives 1 and 3, research objective 4 shall propose a complete multi-layer/multi-domain traffic engineering solution. The solution shall be analyzed and then validated through simulation while placed alongside a prediction based traffic engineering mechanism.

Finally, based on the outcome of objective 3, research objective 5 shall propose a multi-layer GMPLS path computation algorithm that relies on solving a mathematical program-

ming model of the path constraints. The proposed mathematical program shall be solved by the optimization toolbox of MATLAB. The performance of the algorithm shall be evaluated through simulations.

1.5 Contributions and Originalities

This thesis makes original and major contributions to the field of network traffic engineering. The findings of this thesis are not only innovative, but also realistic, in the sense that they build on existing standards or on proposals on their way to standardization (i.e., G/MPLS technology, PCE architecture, RSVP-TE and PCEP signalling, etc.). This means that the solutions in this thesis are valuable to both the research community as well as to the telecommunication industry composed of network operators and equipment manufacturers.

As previously mentioned, the problem consists in computing end-to-end QoS aware traffic engineered G/MPLS paths. This problem is multi-domain as well as multi-layer in nature. The solution to this problem lead to three major contributions in this thesis.

1. Inter-domain contributions:

The initial contribution is the definition of a novel distributed inter-domain optimal path computation scheme and the use of pre-reservations to overcome the risks of deployment blockage. The solution finds optimal inter-domain paths by receiving the complete list of possible paths first. However, the longer inter-domain path computation delays cause inevitable delays in the path computation process. This in turn worsens the risks of resource fluctuation and the probability of blockage at LSP deployment time. The proposed distributed inter-domain path computation scheme not only finds the optimal inter-domain path, but also guarantees unblocked LSP deployment. The proposed scheme relies on the PCE architecture and requires little or no change in existing standards. The findings in this part of the thesis clearly prove the need for a non-blocking inter-domain solution. Thus, the proposed scheme allows the finding of optimal inter-domain paths with a reduced LSP deployment blockage.

2. Joint inter-layer/inter-domain contributions:

Subsequently, this thesis is a pioneer in the consideration of the joint inter-layer/inter-domain problem by the proposal of a novel end-to-end traffic engineering scheme. The proposed ap-

proach consists in adapting the distributed solution of the inter-domain part and by defining specific traffic engineering guidelines that can reduce LSP setup delay and the path request blockage (i.e. increase the throughput). This second part is also original in its consideration of the benefits of traffic prediction, as opposed to existing works which only focus on precise prediction algorithms. The findings in this part are that lower layers should be triggered even when bandwidth resources are still available at the higher layer. Moreover, it is established that there is an interest for traffic predictions in the proposed traffic engineering method. However, they do not need to be accurate in order to obtain the desired results. In fact, an accuracy of 50% proved to be beneficial. This is of major importance when considering the vast number of research works on traffic prediction.

3. Inter-layer algorithmic contributions:

The last, but extremely valuable contribution brought by this thesis is the definition of a novel GMPLS multi-layer/multi-region CSPF algorithm that considers complex constraints overlooked by exiting works. The proposed algorithm relies on one part on the computation of K shortest paths, which allows the definition of a mathematical program that considers tricky GMPLS adaptation constraints. In fact, GMPLS inter-layer LSPs are more suitable for nesting/un-nesting as opposed to conversion. This is known as good inter-layer traffic engineering practice. However, the nesting/un-nesting adaptation functions raise many constraints, mainly one which is analogous to the parenthesis problem, which is solved by the proposed mathematical program. Despite the fact that these GMPLS constraints are being considered for the first time in a CSPF algorithm, the proposed solution was still be validated by comparing it to an existing graph transformation method. At last, the proposed CSPF algorithm can be implemented within any standard PCE node and allows, for the first time, the dynamic GMPLS inter-region path computation and deployment.

Additionally, general but very significant, outcome of the works of chapters 4 and 5 is that traffic forecasts are always useful to determine in advance which route and resource assignation scheme will result in overall best results (e.g. in terms of resource utilization). However, given the specific application where the predictions are to be used, the required prediction accuracy should be determined prior to investigating on the actual prediction algorithm to be used. This is important because, very often, the inability of obtaining very precise predictions has discouraged their use in traffic engineering.

Thus, the general contribution of this thesis is the definition of a complete framework

for end-to-end traffic engineering and path computation. The findings are discussed following their respective separation into chapters 3, 4 and 5. It should be mentioned that, throughout the thesis, notwithstanding this clear separation of the work into three themes, the relation between them remains obvious; that is, end-to-end traffic engineering and QoS is only achieved by considering all three proposals simultaneously. Nonetheless, the proposed techniques can be used separately or along other existing traffic engineering solutions.

Finally, as outlined in chapter 6, the findings in this thesis contributed to various published and submitted scientific articles and patents. Moreover, since the work was conducted in parallel with standardization efforts at the IETF, on many occasions the findings and conducted studies lead to direct intervention and participation in the standardization processes; work that has been acknowledged in some draft and RFC documents.

1.6 Organization of the Thesis

The remaining of this thesis is organized as in the following. Chapter 2 presents technology and standard details as well as prior art relevant to this thesis. Chapter 3 presents the pre-reservation based procedure for the elaborate task of inter-domain path computation of LSPs within a PCE based architecture. This chapter also presents the simulation results that bring light to the usefulness of such technique as well as possible drawbacks. Chapter 4 follows by presenting the joint multi-layer/multi-domain problem. It analyzes the adaptation of the inter-domain scheme to the inter-layer scenario. It also presents the validation of the overall solution which is performed through simulations, bringing light to the benefits of inter-layer traffic engineering and the use of prediction. Chapter 5 presents the novel GMPLS inter-layer/inter-region path computation algorithm along with mathematical and simulation results. Chapter 6 concludes this thesis by discussing its major contributions, its main limitations, and well as a selection of future research directions.

CHAPTER 2

RELATED WORK

The next shift in the telecommunication industry is to use a common all-IP infrastructure for the delivery of all types of services. This requires serious consideration of QoS for the traffic, which can only be achieved by the practice of traffic engineering. One important aspect of traffic engineering is the computation of optimal end-to-end paths. However, the telecommunication ecosystem is both multi-domain and multi-layer in nature, which imposes great challenges to the traffic engineering problem. This thesis treats the problem of end-to-end traffic engineering and path computation under three separate themes, notably the inter-domain path computation scheme, the joint inter-layer/inter-domain traffic engineering scheme and the inter-layer CSPF algorithm. This chapter overviews the related standards and scientific research works with respect to these three themes.

2.1 Related Work for the Inter-Domain Path Computation Scheme

The inter-domain traffic engineering difficulty in the current Internet architecture is caused by the various QoS policies enforced with often a different definition or implementation from one domain to the other. Moreover, topology and link state information is essential for any effective traffic engineering mechanism; however, for scalability and privacy reasons, BGP which is the only inter-domain routing protocol does not propagate this information.

2.1.1 Inter-domain traffic engineering with BGP and its shortcomings

The literature has proposed different methods for performing basic inter-domain traffic engineering using BGP. Notably, Bonaventure *et al.* (2003b) present the limited possibilities to control IP traffic at the inter-domain scale using BGP. The work of Fu (2009) also considers inter-domain traffic engineering; but, it is based on BGP and faces the same limitations, i.e., they are too general, based on trial and error, and don't offer guarantees on the QoS.

Then, in Bonaventure *et al.* (2003a) and in Bonaventure et Quoitin (2003), some standardization attempts were made for extending BGP to allow for more control over the traffic. Only the work presented by Sangli *et al.* (2006) has been standardized and consists of a new BGP attribute that can be used to label information carried by BGP. Some of these BGP

traffic engineering techniques are already in use in the Internet. BGP traffic engineering is performed by tuning route advertisements. Again, tuning mechanisms have their limitations; they are trial and error based, give little control over the end-to-end path taken, lack optimality and have no notion of QoS.

Due to these shortcomings, and given the current state of inter-domain routing techniques, the possibility of using other technologies for the control of inter-domain traffic has been contemplated. The main technique that has been considered is MPLS and necessary extensions for its inter-domain deployment .

2.1.2 Multiprotocol Label Switching

Inter-domain MPLS promised to be more useful in controlling the inter-domain traffic, but it was not fully standardized until recently. The works of Okumus *et al.* (2001) and Pelsser et Bonaventure (2003) gave early solution to the deployment of MPLS in inter-domain settings. Then, Farrel *et al.* (2008) standardized the MPLS technology for inter-domain reachability. Before introducing these extensions, a recapitulation of basic MPLS is necessary.

The functionality of MPLS can be explained better with the help of Figure 2.1 which shows a typical MPLS network. The IP packet is only routed once at the ingress Label Edge Router (LER) where it gets assigned to a forwarding group and receives a label. It is then forwarded through the network following the LSP assigned to its label. At each Label Switched Router (LSR), the label is swapped with another label of local significance, according to the Label Forwarding Information Base (LFIB) table of the LSR. When the packet emerges at the egress LER, the last label is removed and the packet is forwarded to its destination using IP or any other layer three protocols.

In each node, packets assigned to a given label belong to the same Forwarding Equivalence Class (FEC). A FEC is a logical entity that designates a group of packets undergoing equivalent forwarding in a given node. During normal IP operation, for each possible next hop, a router usually creates a different FEC. With MPLS, other more advanced criteria can be used to designate a FEC. This is very useful for traffic engineering purposes.

Resource ReSerVation Protocol-Traffic Engineering (RSVP-TE)

The deployment of LSPs are signaled using Resource Reservation Protocol-Traffic Engineering (RSVP-TE) as defined by Awduche *et al.* (2001). In RSVP-TE, the well known Resource ReSerVation Protocol (RSVP) defined by Braden *et al.* (1997) is enhanced to enable routers supporting both RSVP and MPLS to associate labels with RSVP flows. To support

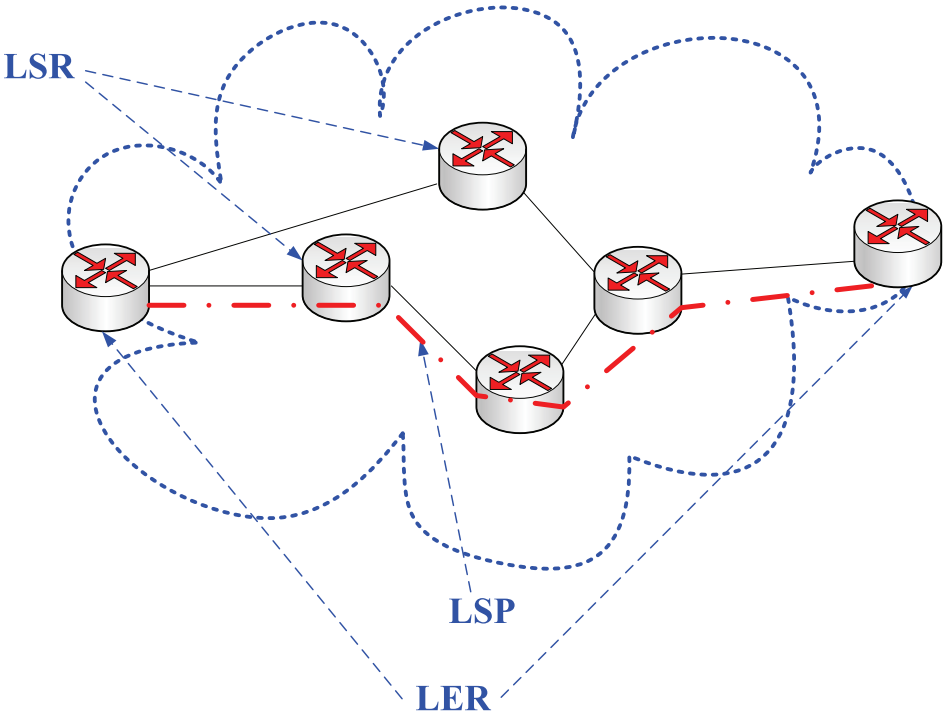


Figure 2.1 Multiprotocol Label Switching (MPLS)

MPLS, RSVP-TE introduces new objects that will be carried inside RSVP *Path* and *Resv* messages.

The *LABEL_REQUEST* object is carried inside a *Path* message initiated by the ingress LER. Its purpose is to request the egress LER to initiate a reservation and establish an LSP along the path followed by the *Path* message. The egress LER assigns a label to the LSP that is being created, puts that label in the *LABEL* object of a *Resv* message and sends it to the next node upstream. At each node, a local label is assigned to the LSP, the *LABEL* object is updated and sent to the next node upstream. This procedure ends at the ingress LER, creating this way the LSP.

RSVP-TE also introduces two other important objects for traffic engineering purposes. The Explicit Route Object (*ERO*) and the Record Route Object (*RRO*). These objects are used to allow the LSP to be established along a predefined route rather than the one obtained by the IP routing protocols. The predefined route can be calculated by different means, e.g. using manual configuration or by a PCE using the schemes proposed by this thesis for example. Thus, optimal explicitly routed LSPs could be used to avoid congested routes, to take disjoint routes during fault recovery mechanisms, and simply to obtain the required QoS.

2.1.3 Inter-domain extensions for MPLS

The above standards have been extended to support MPLS on a multi-domain scale. In Farrel *et al.* (2006b), a framework for the deployment of inter-domain LSPs is given. Based on this framework, Farrel *et al.* (2008) proposed the necessary extensions to RSVP-TE and definitions for the deployment of inter-domain LSPs.

Just like intra-domain LSPs, the inter-domain LSPs can be signalled in three different ways. The first signalling approach is the *contiguous* traffic engineered LSP. This type of LSP is setup across the domains with a single RSVP-TE session and with the same LSP identification (ID) at every LSR along the path.

The second approach is the *nested* traffic engineered LSP where, as described by Kompella et Rekhter (2005b), more than one LSPs can be carried inside another LSP, in a nested fashion. This allows the nesting of inter-domain LSPs inside the intra-domain LSPs in the traversed domains.

The third signalling approach is the *stitched* traffic engineered LSP where, as described by Ayyangar *et al.* (2008), smaller LSP segments are connected together to create a single end-to-end LSP. Thus, intra-domain segments can be stitched together to form an inter-domain LSP. Thus, from a data plane perspective, the end result will be a contiguous LSP; from a control plane perspective, each segment has its own RSVP-TE session and the stitched LSP has its own session, similar to the nesting case. The RSVP-TE extension for signaling the type of LSP to use across domains consist in using the *LSP_Attributes* object defined in Farrel *et al.* (2006a).

Moreover, when an error occurs during LSP setup (e.g. unavailable resources), a *PathErr* message is sent back to the LSP's ingress node to report the error. If the failed LSP traverses multiple domains, this *PathErr* is successively returned by each domain's border node. Only the border nodes can modify the information carried in the *PathErr* message.

2.1.4 Path Computation Element (PCE) architecture

The technological challenge of signalling an inter-domain LSP has been answered by the above described standardization works. However, the actual traffic engineering challenge is the computation of the optimal end-to-end route for the inter-domain LSP.

The process of CSPF path computation is often resource hungry in terms of CPU power and memory. Moreover, as in the inter-domain case, it is often impossible for one entity to have visibility on all the required resource information to compute the end-to-end path. Due

to these restrictions, the PCE architecture has been proposed by the IETF for the context of MPLS and GMPLS traffic engineering (Farell *et al.*, 2006).

As briefly introduced in the previous chapter, a PCE node, shown in Figure 2.2, is an entity that can reside inside a router or on a separate entity. The PCE has resource visibility

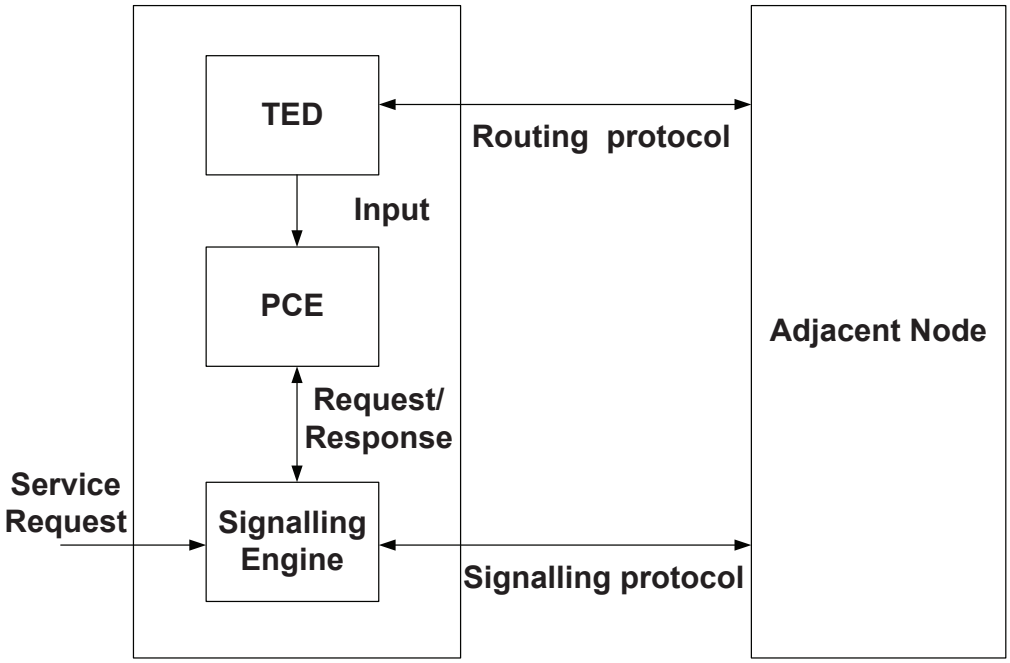


Figure 2.2 Path Computation Element node

through the traffic engineering database (TED). It has its own signalling protocol called PCE Communication Protocol (PCEP) as defined by Vasseur et LeRoux (2009).

The service of a PCE is usually triggered by a Path Computation Client (PCC) which could be another PCE. The standard sets the number of supported PCCs to 1000 per domain and the number of PCEs to 100 per domain. Each PCE can have up to 1000 PCCs that could send requests to it. A PCC can have up to 100 PCEs to which it can send requests. The maximum number of domains considered by the standard is only 20. The standard also recommends in average not more than 10 request messages per second sent to the same PCE. It also sets a maximum burst of 100 requests per second per PCE within a 10 second interval.

PCE Communication Protocol (PCEP)

The PCEP protocol relies on TCP for communication between PCEs or between a PCC and a PCE. It defines a number of messages described next. The *Open* message is defined for the initiation of the session. A *Keepalive* message serves for keeping the connection alive. A *Close* message serves for closing the session. The Path Computation Request ($PCReq$) and the Path Computation Reply ($PCRep$) messages carry a path computation demand and its reply. A $PCErr$ message serves for communication error messages and a $PCNtf$ message serves for notification of special circumstances between PCEs.

A Request Parameters RP object which carries a $Request\text{-}ID\text{-}Number$ object is carried by each $PCReq$ message. The RP object has a variable length and may contain additional *Type Length Value* (TLV) fields. The corresponding $PCRep$ message carries the same RP object.

In the $PCReq$ message, the $END\text{-}POINTS$ object is carried to specify the source IP address and the destination IP address of the path for which a path computation is requested. A $BANDWIDTH$ object is used to specify the requested bandwidth for a traffic engineered path, and is carried inside the $PCReq$ message. Similarly, a $METRIC$ object can be carried inside the $PCReq$ message to specify other traffic engineered metrics (e.g. hop count) for the requested path. Moreover, a LSP Attributes $LSPA$ object is optionally carried in the $PCReq$ message to specify various constraints to be considered when computing the path.

If a path is found by the PCE, an Explicit Route Object ERO is carried within the $PCRep$ message to return the computed traffic engineered path. To preserve confidentiality in inter-domain path computation, instead of explicitly expressing the computed route, Path-Key Subobjects (PKSs) are carried in the ERO of $PCRep$ Messages (Bradford *et al.*, 2009).

If a path is not be found, a $NO-PATH$ object is returned, which can contain an optional $NO\text{-}PATH\text{-}VECTOR$ TLV to give the details of why the path computation failed.

2.1.5 Existing inter-domain path computation schemes

Several inter-domain path computation procedures have been proposed, but, only recent inter-domain path computation schemes rely on the given PCE architecture. Before the PCE architecture was developed, many works proposed methods based on bandwidth broker nodes.

Among these works, Okumus *et al.* (2001) proposed to establish inter-domain LSPs by using bandwidth management points (BMP) and a certain SIBBS protocol (Simple Inter-

domain Bandwidth Broker Signalling). In the proposed architecture, the BMPs may receive requests for allocation of resources as Resource Allocation Requests (RAR) from different sources. The RAR message can reach a node within an AS, another BMP, or a third agent representing an application or node. The BMP responds to the RAR by a Resource Allocation Answer (RAA).

This straight forward proposal was used in other inter-domain solutions and finally by the PCE architecture. This is not a coincidence because this is the only reasonable way to compute inter-domain paths in a distributed inter-domain environment, where centralized solutions have been excluded by the telecommunication communities.

The most important and recent path computation schemes are the Per-Domain Method defined by Vasseur *et al.* (2008) and the Backward Recursive PCE-based Computation (BRPC) procedure proposed by Vasseur *et al.* (2009). The Per-Domain method consists in having each domain compute the local portion of the end-to-end LSP, probably by using a PCE. This determines the possible exit point and therefore the next domain, which will be triggered in the same manner until the end-to-end path is computed. The work of Aslam *et al.* (2007) studies the inter-domain path computation scenarios related to the per-domain scheme. Their results show that paths computed this way are not guaranteed to be optimal.

On the other hand, the BRPC method allows the computation of an optimal inter-domain path by exploring all end-to-end paths and by letting the initiating domain choose the best possible one. It guarantees that the optimal inter-domain path is always found. As defined by Vasseur *et al.* (2009) and analyzed by Dasgupta *et al.* (2007) as well as by Paolucci *et al.* (2008), BRPC consists in having a PCC send a path request message to a first PCE in its domain. The inter-domain path computation request is forwarded all the way from PCE_1 to PCE_N in the destination domain. In each domain, all the AS Boundary Routers (ASBRs) are considered. The replies to this request consist of each domain/PCEs computing all possible paths and adding the results as a Virtual Shortest Path Tree (VSPT) in the reply. The replies are returned upstream, backward recursively, to PCE_1 (or the PCC). PCE_1 (or the PCC) receives all the replies and uses the VSPT information to choose the optimal path to use for the inter-domain LSP. Like other inter-domain path computation procedures, BRPC is exposed to significant PCE response times that could result in LSP blockage during deployment.

This response time is a drawback of current distributed inter-domain path computation procedures. Other than the need for a distributed solution, the main idea that resulted in the conception of the PCE architecture is based on the need of a separate network entity to compute optimal paths, since this task is often very demanding in terms of processing

resources. Thus, it is important that any scheme relying on the PCE architecture uses these resources efficiently.

As it will be discussed in chapter 3, the overall response time in the inter-domain environment may impose a substantially significant lapse of time between the path computation request and the reception of a reply. Then, by the time the signalling for the LSP deployment reaches the corresponding domains, resources once available during the path computation process may not be available anymore. Therefore, blocking errors could occur at LSP establishment time, requiring more signalling, e.g. Crankback as defnied by Farrel *et al.* (2007), and thus resulting in an increased setup time and a possible sub-optimal inter-domain path. This is not to mention the waste of the PCEs processing resources.

Moreover, in an inter-domain environment, blockage in one domain could require LSP tear downs and redeployments in previous domains, which complicates the situation even more. This problem is not unique to the PCE architecture, as seen in other works.

Among these works, Mantar *et al.* (2004) among others propose a general admission control method to prevent blockage by adjusting the deployment/reservation rate to the incoming request rates. This method lowers the blocking probability of reservations between path negotiations, but does not prevent blockage.

Then, the work of Sanchez-Lopez *et al.* (2007) addresses a similar issue in an ATM environment, but their proposed solution is very different as they try to resolve the problem by proposing new LSP deployment signalling for a faster LSP deployment through an ATM network.

Nevertheless, the path computation time is more significant in the inter-domain case due to the number of domains and PCEs to be crossed. A new solution is needed because the above proposed methods do not apply to a PCE based inter-domain path computation. Indeed, the work of Mantar *et al.* (2004) is centralized whereas most PCE based schemes tend to be cooperative and distributed. The work of Sanchez-Lopez *et al.* (2007) does not apply because the inter-domain blockage risk is mainly due to the path computation time as opposed to the deployment signalling time.

2.2 Related Work for the Joint Inter-Layer/Inter-Domain Traffic Engineering Scheme

The traffic on the Internet (IP traffic) has to traverse a certain number of nodes or routers from source to destination. These routers in turn have to be connected to other routers in order to connect the end-to-end path. This router connectivity is often offered by an optical

transport network built from optical switches, as shown in Figure 1.1 of the previous chapter. The whole picture can easily be viewed as traffic demands routed on the IP network, which in turn is connected via the transport network. This is the simplest example of an inter-layer setting that is often considered for describing inter-layer routing issues.

2.2.1 Importance of an inter-layer path computation scheme

The importance of inter-layer traffic engineered path computation can be highlighted with the example of Figure 2.3 where the IP demand layer is considered as a higher layer, on top of the MPLS layer which is supported by the optical layer. As already mentioned, one basic QoS criterion is high service availability. This criterion often requires diversity of the links at the physical layer to preserve reachability in the presence of link or node failures. The example shown below highlights the importance to have a complete view of the layers in order to provide path diversity for recovery mechanisms.

Path diversity consists of computing a primary path for the traffic as well as a diverse backup path in case a failure occurs. Recovery mechanisms are defined by Pan *et al.* (2005). It consists of a way to establish backup paths for the restoration of LSPs in case of failure. The actual difficulty is not in the signalling involved, but in the calculation of such paths.

In fact, if each layer treats the path diversity problem separately, inconsistencies may occur. One problem is that two disjoint paths at a higher layer may share the same lower layer link. This means that the failure of such apparent diverse links at the higher layer will happen simultaneously with the failure of their lower layer common link. This is more commonly called in the literature as the Shared Risk Link Group (SRLG). Thus, to provide route diversity the higher layer backup path must be on a different SRLG than the primary path.

The example of Figure 2.3 considers the IP layer and the route between routers R2 and R3. It is assumed that path diversity is required for better performance in case of failure of the R2-R3 link. There are two routes between R2-R3 routers. However, by looking at the layer below (MPLS), it is interesting to note that these two routes share the same MPLS links (LSPs), $R_{mpls}2$-$R_{mpls}1$-$R_{mpls}3$, that is the IP layer routes R2-R3 and R2-R1-R3 both pass by these same MPLS layer links.

Now, it is interesting to note that this MPLS layer path (LSP), has diverse routes at the optical layer, where there is a direct link between the $Switch_{optical}2$ and $Switch_{optical}3$ and an indirect path passing by $Switch_{optical}1$. This path diversity example clearly shows the importance of inter-layer traffic engineering. By considering all layers, it is possible to obtain

true physical path diversity as well as avoid duplication which could occur if the problem is considered separately at each layer. Moreover, it is only by having an inter-layer view that specific redundancy measures can be applied at path computation time.

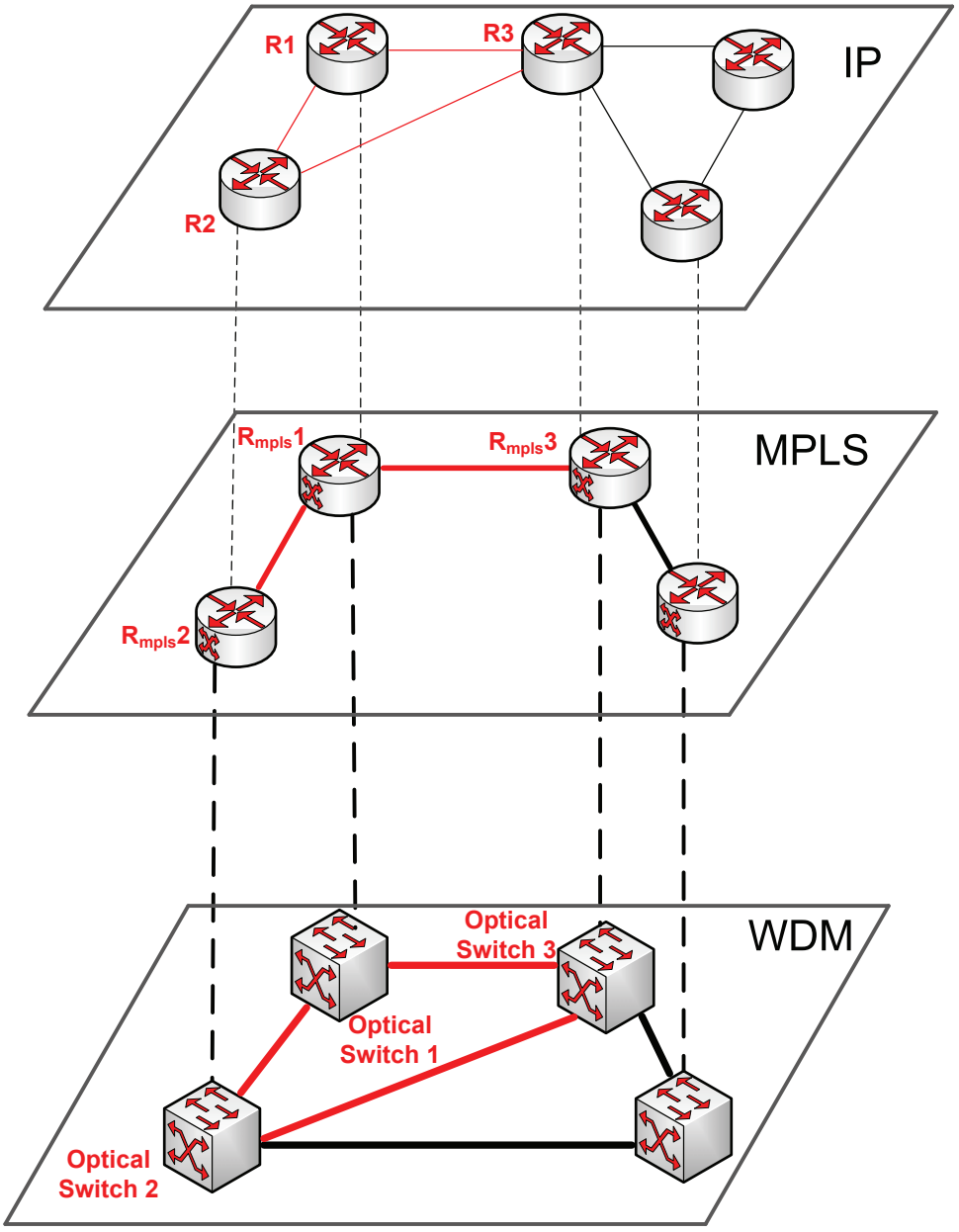


Figure 2.3 An inter-layer scenario with an IP demand layer on a MPLS layer routed on a WDM optical transport layer

Furthermore, following the last comment about duplication at different layers, it is intuitive that considering network resources at all layers when performing routing and traffic engineering is more efficient. This is not to mention that by considering all layer, true global optimization is obtained, often lowering the overall CAPEX and OPEX costs of the network. However, this gain is often obtained with an additional computational complexity attached to multi-layer traffic engineering problems.

2.2.2 Generalized MPLS (GMPLS) for inter-layer path deployment

In terms of technology, GMPLS (Mannie, 2004) is used for the deployment of multi-layer traffic engineered paths (LSPs). GMPLS or Automatic Switched Optical Network (ASON) standards are a promising solution for an ultimately unified control plane.

Automatic provisioning in the control plane is standardized within Study Group 15 (SG15) of the International Telecommunication Union, Telecommunications Standardization Sector (ITU-T), under the ASON umbrella of recommendations, and in the IETF, under the umbrella of the Common Control and Measurements Plane Working Group (CCAMP WG).

In Berger (2003), the RSVP-TE protocol is extended to support GMPLS. It generalizes existing RSVP-TE messages and objects. GMPLS will be discussed in section 2.3. For this part of the thesis, it is important to point out that the concepts of multi-layer and multi-region are different in GMPLS.

A region refers to switching technologies (e.g. packet switch or TDM). A layer refers to granularities inside a switching region. For example in TDM, an OC3, a VC4 or V11 are examples of GMPLS layers. This part of the thesis refers to a layer to designate both GMPLS regions and layers. Moreover, the same MPLS LSP signalling types are available in GMPLS, that is contiguous, stitched and nested.

Furthermore, as presented in Figure 2.4, the GMPLS control plane supports the overlay model, the hybrid or augmented model, and the peer model. The solution in chapter 4 is suitable for the overlay and hybrid models, but the solution presented in chapter 5 is suitable for the peer model.

A framework for PCE based multi-layer G/MPLS traffic engineered path computation is defined by Oki *et al.* (2009). It too, proposes to trigger lower layers only when capacity is no longer available at higher layers. However, this technique will be shown as not optimal when a considerable number of requests are to be routed.

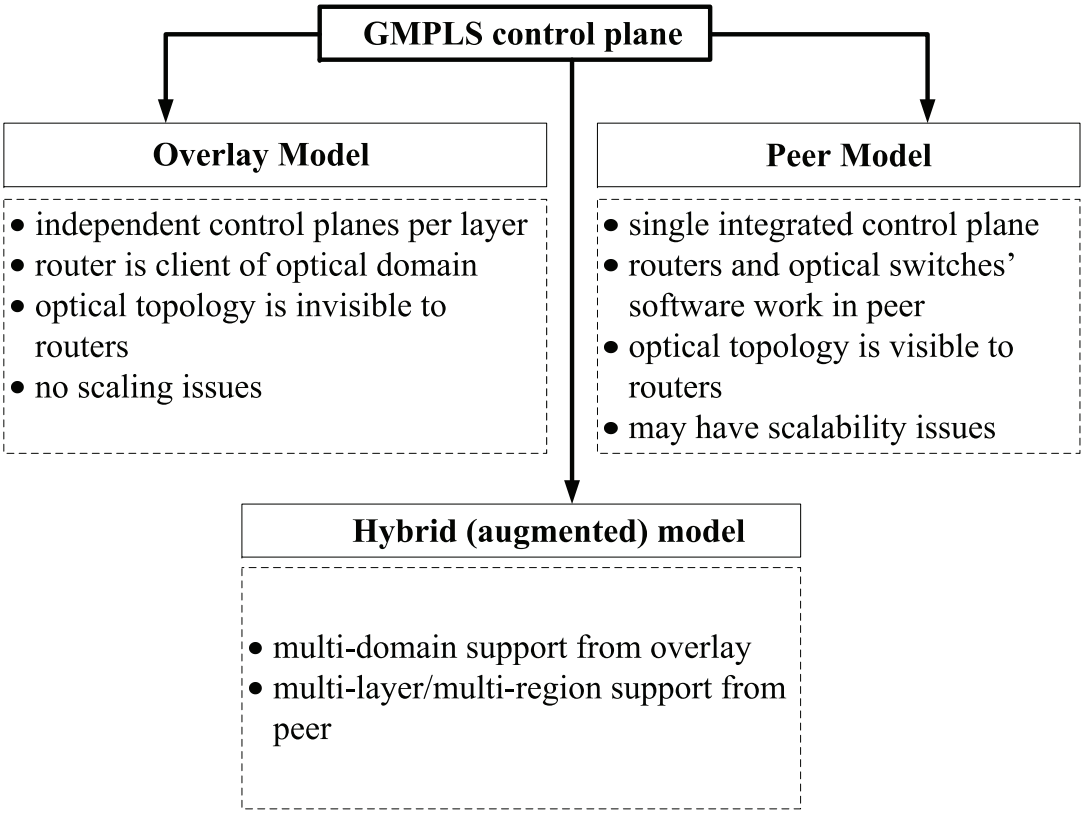


Figure 2.4 GMPLS control plane options

2.2.3 Existing inter-layer path computation and traffic engineering schemes

Most of the existing traffic engineering and path computation solutions address QoS problems in a targeted manner, often applied to a specific technology (layer). This does not capture the entire end-to-end traffic engineering (path computation) problem. Moreover, all works trigger lower layers only when resources become unavailable at the higher layers.

The work of Wang *et al.* (2008) presents a survey of traffic engineering practices. Most of the presented solutions refer to the intra-domain, single-layer case. After that, the work of Fu (2009) also considers end-to-end traffic engineering without considering the multi-layer aspects. This is often due to the complexities involved with the complete consideration of the inter-layer problem.

Interestingly, the work of Szigeti *et al.* (2004) argues a solution for inter-layer/inter-domain routing, but it reduces the problem to the single layer one each time the inter-domain part is addressed. Then, the work of Tomic (2007), under the topic of virtualized optical networks, touches the inter-domain possibility of multi-layer GMPLS networks. Again, this

work only considers the optical layer. Moreover, the proposed solution introduces new node functionalities or network elements not based on existing standards like the PCE architecture.

Subsequently, the work of Harhira et Pierre (2008) presents a novel traffic engineering admission control procedure in GMPLS networks. However, it considers lower layers only when the higher layer does not have sufficient resources.

Finally, the work of King *et al.* (2008) presents the PCE architecture as an enabler for multi-domain consideration when computing GMPLS paths. However, their definition of domain does not apply to separate administrative domain and they do not treat the joint inter-layer/inter-domain problem.

A word on traffic prediction

Additionally, a popular traffic engineering trend is to use traffic predictions for various purposes. These could be from CAC algorithms to CSPF routing algorithms. However, most works on prediction concentrate on the algorithm for the prediction itself as opposed to its real benefits when used within a traffic engineering scheme.

To cite a few works Papagiannaki *et al.* (2005) as well as Cortez *et al.* (2006) present neural network based prediction for traffic but do not concentrate on its real applicability in a traffic engineering scenario. Many other works using Kalman filters have also been proposed for traffic prediction (Anjali *et al.*, 2003).

Concluding remarks

In conclusion, the existing traffic engineering and inter-layer works are often too simple and they do not consider the joint inter-layer/inter-domain problem. Moreover, they only consider lower layers when resources become unavailable at higher layers. Also, traffic prediction works address the accuracy of the predictions as opposed to their usefulness.

Finally, in terms of validation of inter-layer proposals, the significant work of Tsirakakis et Clarkson (2009) recommends to develop a simulator to evaluate the performance of proposed schemes as opposed to use existing and well known simulators like OPNET or NS2, which have many multi-layer modeling shortcomings.

2.3 Related Work for the Inter-Layer CSPF Algorithm

GMPLS technology is the basis for the setting for the proposed solution in the third part of this thesis. The sections below present missing important concepts about GMPLS, as well

as existing works related to GMPLS CSPF algorithms.

2.3.1 GMPLS regions and layers

Generalized MPLS or GMPLS as defined by Mannie (2004) allows for the label switching of not only data packets, but also other switching technologies. The interfaces on a GMPLS router or node can have, as defined by the standard, one or many of the five switching capability interfaces. The interfaces can be 1-Packet switch capable (PSC), 2- Layer 2 switch capable (L2SC), 3-Time division multiplex capable (TDM), 4-Lambda switch capable (LSC) and 5-Fiber switch capable (FSC). These five switch types are defined as regions in GMPLS nomenclature. When considering the optical regions, bandwidth is represented by optical carrier units as presented in Table 2.1. The same optical levels are often used to designate bandwidth in other layers. Moreover, the T1 (1.5 Mbps) and DS3 (50 Mbps) TDM bandwidth rates are commonly used.

Table 2.1 Optical data rates (SONET/SDH)

SONET	SDH	Approximate Bandwidth Used in this Thesis
OC3	STM1	150 Mbps
OC12	STM4	600 Mbps
OC48	STM16	2.5 Gbps
OC192	STM64	10 Gbps

GMPLS aims mainly at the nesting of higher layer/regions LSPs into lower ones. Figure 2.5 shows a conceptual picture of GMPLS nested technologies. As seen, even without GMPLS, these switching layers existed and were carrying the traffic of higher layers into lower layers in a nested fashion. With GMPLS, the signalling means are provided to automatically perform the switching and nesting by representing the connections in each layer by LSPs.

2.3.2 Terminology used and existing works

In both Farrel et Bryskin (2006) and Bryskin et Farrel (2006), the concept of Hierarchical LSPs (H-LSP) is introduced. A H-LSP is described as a LSP created in a lower layer to provide data links to higher layers. Subsequently, LSPs created to provide data links to the same layer are named *stitching* LSPs. In Farrel et Bryskin (2006), the comparison is made between *contiguous/stitched* versus *nested* LSPs in a multi-layer/multi-region environment. It is clearly stated that due to large bandwidth gaps between LSPs of a higher layer when

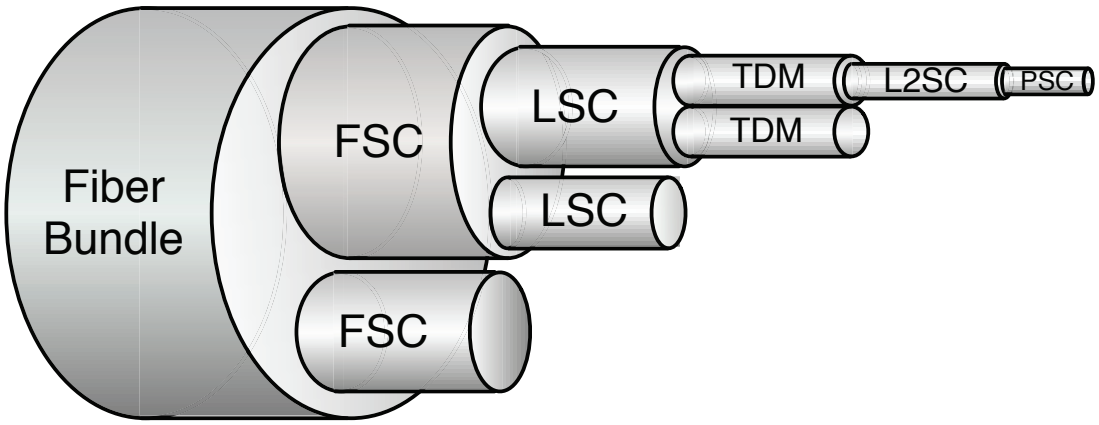


Figure 2.5 Nested LSPs

going down the layers (up in the switching hierarchy), it is intuitively more efficient to opt for nested LSPs.

Moreover, the concept of Virtual Network Topology (VNT) refers to the LSPs at a lower layer that are advertised as links into a higher layer. The work of Shiomoto *et al.* (2008) defines multi-layer/multi-region traffic engineering as the computation of end-to-end paths across layers and the use of mechanisms that control and manage the VNT by deploying and releasing LSPs in the lower layers. The latter concept is called vertical integration between switching regions.

Then, they define the concept of a GMPLS node's Interface Switching Capability (ISC), which is the interface's ability to forward data of a particular data plane technology, uniquely identified by a switching region. A node can have a *single* or *multiple* switching types capabilities. Nodes with multiple switching types capabilities are further categorized as *simplex* or *hybrid*.

A simplex node is capable of terminating a single switch type per interface. A hybrid node is defined as capable of terminating more than one switching technology on the same interface. A hybrid node has thus more than one switching elements (matrices). The term *adjustment* is defined as the property of a hybrid node to interconnect internally the different switching capabilities (matrices) that it provides through its external interfaces. This is explained as the possibility of joining links with different switching capabilities in a node that can adapt (adjust) the signal between the links.

However, in Bryskin et Farrel (2006), authors only mention a node's adaptation capability (i.e., the term adjustment is not mentioned). The adaptation capability of an interface is defined as its capacity to perform a nesting function to use a locally terminated connection

from one layer as a data link of a higher layer. In the same manner, Shiomoto *et al.* (2008) defines the concept of triggered signalling as the nesting of upper layer LSPs into advertised lower layer LSPs.

The actual hybrid node's adjustment functionality consists for example of a node, with PSC and LSC interfaces, that has its electrical IP router ports internally connected to its Optical Cross-Connect (OXC) ports. While such adjustment consisting of extracting the signal from one switching region and putting it into another switching region is technically possible, it is often undesirable due to the loss of bandwidth when going up the switching hierarchy. Another concern with adjustment between switch types is the node's resource consumptions involved with such transformation.

Most of the existing works that consider the multi-layer/multi-region aspect of path computation with adaptation capability constraints use an ambiguous description of the adaptation practice. For example in Shiomoto *et al.* (2003); Jabbari *et al.* (2007); Gong et Jabbari (2008); X.Yang *et al.* (2009), the definition of the term hybrid is ambiguous. It is mostly mentioned and described as a node with multiple switch types capabilities, which is the definition for both simplex and hybrid nodes. Then the hybrid node is assumed capable of adjusting (i.e., completely transforming) from one switch type to another, and this capability is always used, even if a hybrid node also has nesting functionalities, like simplex nodes. Likewise, the work in Mouftah et Naas (2008) considers conversion and regeneration capabilities but only at the optical layer, thus not considering the higher region nesting possibilities and constraints.

To resolve the incomplete consideration of adaptation functions in prior arts, chapter 5 of this thesis proposes a novel algorithm for the routing of end-to-end multi-layer/multi-region LSP requests, based on the formulation of the complete set of constraints involved in GMPLS multi-layer/multi-region environments. One of the reasons for which prior works have simplified the problem is its complexity. The work of Huang *et al.* (2006) has shown that even for simple network topologies where the routing is trivial, just the grooming problem is in reality NP-Hard.

Switching capability and adaptation capability needs a clear definition to understand the material in chapter 5. In Bryskin et Farrel (2006), these important GMPLS terms are described. Network elements may be single switch type capable or multiple switch types capable nodes. Single switch type capable nodes advertise the same Interface Switching Capability (ISC) value as described by Kompella et Rekhter (2005a).

Multiple switch type capable nodes are classified as either *simplex* or *hybrid*. According to Shiomoto *et al.* (2008); Bryskin et Farrel (2006); Farrel et Bryskin (2006), a simplex node

has more than one switching capability, but internally the different switching matrices for each type are not connected. This implies that it cannot adjust or transform the signal or the traffic from one switch type to another. Then for this case, the adaptation capability of the node is restricted to the nesting and un-nesting of multi-layer LSPs when crossing region boundaries.

For the case of hybrid nodes, it is considered that the switch types supported can be transformed or adjusted to the other types supported by the node, given that it has the required internal resources and capacity. Then in this case, the adaptation capabilities of the node are nesting and un-nesting, as well as adjusting (converting) multi-layer/multi-region LSPs.

As for the applicability of hybrid nodes using adjustments of multi-layer/multi-region LSPs, the recommendation of Farrel et Bryskin (2006) is followed, which is to opt as much as possible for nesting multi-layer LSPs in order to avoid node resource consumption and the loss of bandwidth when going to lower layers. In all cases, the option of contiguous (and stitched) LSPs remains the first adaptation choice if available.

A graph transformation method

The method presented in Jabbari *et al.* (2007) and in Gong et Jabbari (2008) will be compared to that of chapter 5. The proposed graph transformation technique consists of transforming the initial network graph G to a graph H. Graph G has a set of network nodes v connected by links that have one or many switching types (e.g. one or many of the five GMPLS switching capability types). For each node v_k that can transport or adapt a switch type s_x in the incoming link $\langle v_j, v_k, s_x \rangle$ to switch type s_y in outgoing link $\langle v_k, v_l, s_y \rangle$, an arc $\langle N_{jkx}, N_{kly} \rangle$ is created in the transformed graph H.

Figure 2.6 shows an example network for which a transformed graph is shown in Figure 2.7. Node 6 is the only non-hybrid node. A source node S and a destination node D are added to the transformed graph before running a shortest path algorithm (e.g. Dijkstra or K-shortest path) on it. The cost of each link in the transformed graph is obtained directly from the original graph's costs of the corresponding links that are to be used.

Clearly, the graph transformation method of Jabbari *et al.* (2007) fails to find a path by giving priority to nesting and un-nesting multi-layer/multi-region LSPs. In fact, it completely neglects the nesting and un-nesting capability of nodes.

In chapter 5, the mathematical programming techniques outlined in Pióro et Medhi (2004) shall be used to define a path computation algorithm that respects multi-layer/multi-region

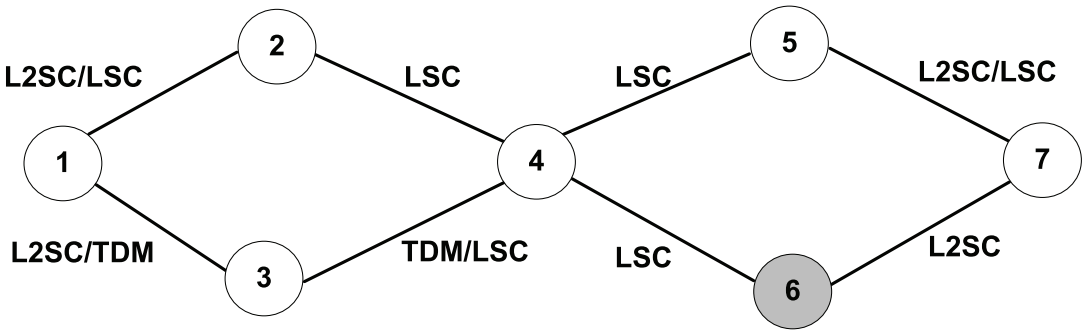


Figure 2.6 Example of multi-region network topology

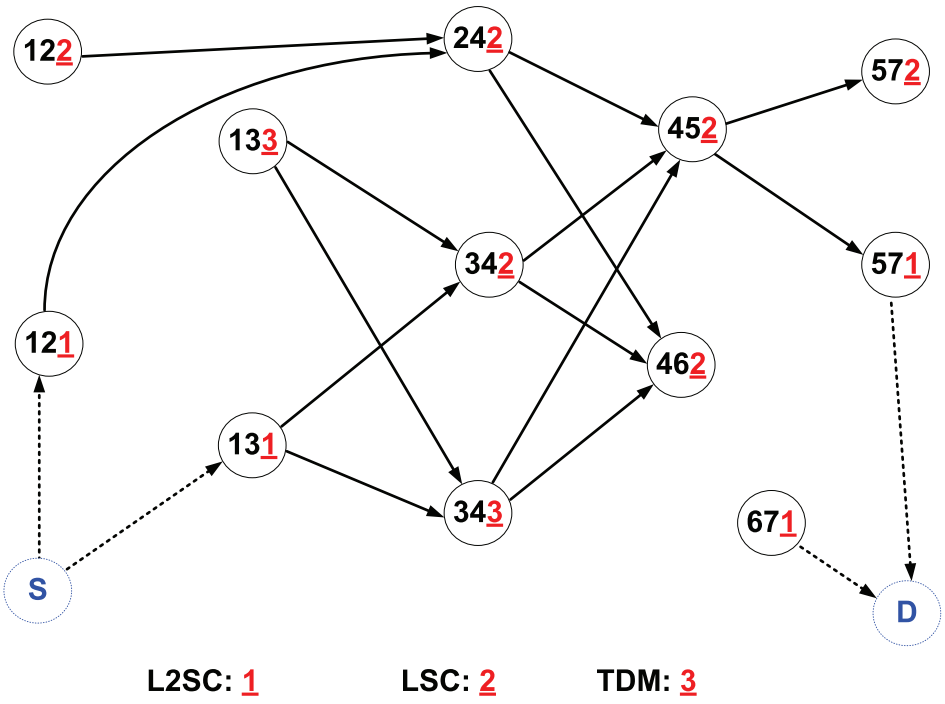


Figure 2.7 Example of multi-region network topology's transformed graph

traffic engineering best practice and adaptation constraints.

Before ending this discussion, it is worthwhile mentioning other CSPF works in GMPLS networks, even if they do not consider any of the above mentioned constraints. For example in the work of Elwalid *et al.* (2003), an optimal design framework is proposed by using GMPLS CSPF. However, the CSPF algorithm used is extremely simple and does not consider any adaptation constraints.

Then, in the work of Martinez *et al.* (2005), a novel CSPF algorithm is proposed, but, it

only considers the optical layer and its wavelength continuity constraint. Finally, in X.Zhang *et al.* (2007), a CSPF algorithm is proposed that maximizes the residual capacity, but ignores all the GMPLS constraints mentioned above.

2.3.3 K-shortest path algorithm

The K-shortest path algorithm has been proposed by Yen (1971). This algorithm is used in chapters 4 and 5 of this thesis. The proposal of chapter 5 relies in particular on this algorithm.

Figure 2.8 presents the K-shortest path algorithm. This procedure operates on a network graph G with N nodes and M links. A non-negative weight/cost is associated with each link. It returns the K shortest paths (if they exist) from a source node s to a destination node d.

```
BEGIN PROCEDURE
k := 1
P := Dijkstra(G,s,d)
S := {(P,s)}
X := {P}
K := {P}
   WHILE k < K and X ≠ ∅ DO
      X:= X\{P}
      w:= DeviationVertex(S,P)
      FOR v ∈ ⟨subPath(w,d) ⊙ d ⟩ DO
         G':= RemoveVerticesEdges(G,s,v,K,P)
         Q := subPath(s,v) ⊕ Dijkstra(G',v,d)
         X := X ∪ {Q}
         S := S ∪ {(Q,v)}
      END
      P := shortest(X)
      K := K ∪ {P}
      k := k+1
   END
TERMINATE
```

Figure 2.8 K-shortest path procedure

This procedure calls another shortest path algorithm to find the shortest path after each iteration, in this case Dijkstra's algorithm (Dijkstra, 1959), which also requires positive link weights. Thus, for $K = 1$ it returns the same result as Dijkstra's algorithm.

The principle is to first compute *the* shortest path P_1. A diversion vertex (or diversion node) is associated with each computed path P_n. The diversion vertex of *the* shortest path is the source node s.

This (P_1, s) information is added to the set S and P_1 is added to set X. Sets S and X are later used by the algorithm. P_1 is also added to set K which holds the shortest paths found so far.

The actual procedure starts with these initial sets X and S. It removes from X the last path P added to K. Then from the set S, it gets the deviation vertex associated with path P, and assigns it to w.

Then, for all nodes v from this deviation node w up to the node before the destination node d in path P, the procedure calls Dijkstra's algorithm on graph G' which is obtained by disabling all vertices and the corresponding links of the nodes in P before the deviation vertex v.

Moreover, the outgoing link from v incident in P has to be removed. This prevents the algorithm from finding paths already found or paths with loops. Then, the sub-path found from v to destination node d is concatenated to the sub-path from source node s to v. This newly found path P_n and the duo (P_n, v) are added to sets X and S respectively.

Then, the set X is examined and the shortest path in X is selected to be added in set K. The current path P becomes this newly added shortest path and the algorithm repeats while $k < K$ and $X \neq \emptyset$. This algorithm has a complexity of $O(KN(M + NlogN))$.

CHAPTER 3

INTER-DOMAIN PATH COMPUTATION SCHEME

As presented during the literature review in section 2.1, compared to the intra-domain case, there are little inter-domain path negotiation schemes thoroughly proven to be effective and optimal. This chapter proposes a straightforward inter-domain path negotiation scheme that could guarantee optimal inter-domain path computation. Moreover, all path negotiation schemes are prone to blocking at deployment/reservation time. This becomes more challenging in the inter-domain case due to longer path computation times that could increase the risk of resource fluctuation hence the potential risk of blocking.

Inter-domain path negotiation schemes suffer from an overall long response time to a path request because the path negotiation scheme has to be performed across multiple ASes. There is a substantial lapse of time between the path computation request and the reception of a reply to this request. By the time the signalling for the deployment of the actual reservations along the optimal path is propagated across the domains, resources once available during the path computation process may not be available anymore. Therefore, blocking errors could occur at this time, requiring more signalling, e.g. crankback defined by Farrel *et al.* (2007), thus resulting in an increased setup time and the possibility of a sub-optimal inter-domain path.

Moreover, in an inter-domain environment, blockage in one domain could require reservation tear downs and redeployments in previous domains, which could complicate the situation even more. It should also be taken into account that in these cases, all the PCEs' resources used to compute the path are now wasted. This makes the need for non blocking inter-domain path computation procedures a real priority. The solution presented in this chapter addresses this issue by using pre-reservations during the path computation process. That is resources are pre-reserved at path computation time, before the actual reservations are made at deployment time. The solution is defined and applied to the proposed scheme using G/MPLS technology, but the pre-reservation idea is valid for other path computation schemes and reservation technologies.

Before presenting the inter-domain path negotiation scheme, this chapter starts by studying the inter-domain blocking risks in section 3.1. Then the proposed inter-domain path negotiation scheme is presented in section 3.2. The scheme makes use of pre-reservations to avoid blockage at deployment time. It also addresses the issue of looping in path computation,

specifically in inter-domain path computation schemes like the one proposed in this chapter. Section 3.3 evaluates the proposed solution with the use of simulation results. Section 3.4 summarizes this chapter by highlighting its main contributions. Because the PCE architecture is defined more precisely around G/MPLS technology, the analysis and descriptions in this chapter refer to the G/MPLS technology as well. However the main idea applies to any other connection based routing.

3.1 Blocking Probability

G/MPLS path deployment, or LSP deployment, is defined along the path taken by the RSVP-TE *PATH* message. Two methods exist to find or to define this path. Normal IP routing can be used to find this path as the *PATH* message is being forwarded. If the LSP path is previously known, it can be defined in the *ERO* object of the RSVP-TE *PATH* message. This second method is clearly the right choice for LSPs with optimal pre-computed paths. Then, as the LSP is deployed, it reserves the required resources along the path (in general bandwidth resources are reserved following a given reservation style). When the *PATH* message is considered in a network, the bandwidth requested is compared with the bandwidth available. If the available bandwidth is not sufficient, a *PathErr* message is returned[1]. In the case of inter-domain PCE path computation, a certain time is necessary to compute the optimal path. Then, the inter-domain LSP is deployed along this path which is usually specified in the *ERO* object. Intuitively, it can be stated that as the path computation time increases, so does the chance of resource fluctuation and the risk of bandwidth unavailability at deployment time. This is because each PCE will determine a route given its current view of resource availability. A long period of time can elapsed between the PCE's local computation time and the time the complete path is returned to the PCC. Thus, once the PCC signals actual deployment of the LSP, the resources under each involved PCE are more prone to have fluctuated, possibly causing the unavailability of the needed resources. Section 3.1.1 below discusses the factors that could influence the path computation time, then section 3.1.2 proceeds by studying the cumulative nature of blocking probability in the inter-domain case.

1. *PathErr* message with Error Code of 01 for Admission Control Failure, and an Error Value of 0x0002, indicating "requested bandwidth unavailable" (Awduche *et al.*, 2001).

3.1.1 Factors affecting path computation response time

Due to longer end-to-end paths and its multiple AS nature, in an inter-domain path computation scenario, network resource availability is more likely to fluctuate between the time an LSP path is requested and the time the LSP is to be deployed. These fluctuations are emphasized as the PCE path computation response time increases. The overall PCE response time could vary as a function of factors described in Table 3.1, such as the number of domains/PCEs to cross, the PCE-PCE communication time, the CPU limitations of the PCEs, the workload of the PCEs, and the complexity of the objective function requested. These factors can be categorized as network related or node related. In an inter-domain

Table 3.1 Factors affecting PCE response time

Factors	Description
Number of domains	As the number of administrative domains (AS) increases, longer inter-domain PCE communication times are to be expected, specially if the inter-domain peering of PCEs is done on demand.
Number of PCEs	The number of PCEs that intervene in a path request could increase depending on the end-to-end network(s)'s size. This will have a cumulative effect on the overall response time. The number of PCEs could also depend on the path negotiation scheme used.
TCP delay	As per standard, PCE communication relies on TCP connections, and will therefore experience delay in case of network congestion.
CPU limitation	PCE response time could be affected by the hardware limitations of the PCE machine, CPU power and the implementation of the algorithms (software vs hardware).
PCE workload	Naturally, as the number of demands to be treated by the PCE increases, the response time increases. This could lead to extremely long delays when request priorities are used and a demand has lower priority while higher priority demands keep coming in.
Objective function	Depending on the objective function used and the level of difficulty involved with the constraints of the requested path, the PCE response time could increase. Thus, before implementing any new objective function, its worse case response time should be analyzed.

environment, both categories of factors are unpredictable by the PCC requesting the path. Thus, the worse case situation should always be accounted for. By worse case it is meant

that some resources once available at path computation time are no longer available at LSP deployment time. This is is a serious issue when low setup times are a requirement. However it should be re-emphasized that PCE path computation is a resource hungry process in terms of PCE node processing and PCE communication message exchanges in certain schemes. Thus, if path resources become unavailable, it causes a waste of all the PCE processing resources. Therefore, it is recommended to minimize LSP blockage in all cases: intra-domain or inter-domain, with or without a low setup time requirement, etc.

3.1.2 Response time and blocking probability analysis

Assuming a first PCE, PCE_1, receives a path computation request message (*[Path/QoS request]*) from a PCC. If the destination of the requested path is out of PCE_1's scope, then it will have to to forward the request to a subsequent PCE. Assuming M PCEs are consulted in order to compute the given path, PCE_M takes T_M^{comp} time to compute its part of the end-to-end path. T_M^{comp} depends on the workload of PCE_M, its CPU power, its memory, its implementation efficiency, its compiler efficiency, the objective function and algorithms used, as well as the network complexity. PCE_M then makes pre-reservations on the local resources of the computed path and sends a *[Path/QoS reply]* message to PCE_{M-1}. PCE_{M-1} takes T_{M-1}^{comp} time to compute its local part of the path. This procedure goes all the way back to PCE_1. Each PCE_{m-1} also adjusts its QoS capability values based on the response from PCE_m and informs PCE_{m-2} in a similar manner (for example when the end-to-end delay must be returned along with path). This answer is propagated all the way back to PCE_1 which gets a complete end-to-end response. PCE_1 or the requester PCC can then choose to use a particular path for the LSP. In the event of inter-domain path computation, in addition to the above factors the PCE to PCE communication time becomes a significant issue as the number of PCEs interrogated increases. Therefore, if PCE_m takes T_m^{comp} time to compute a path, and the PCEP communication from PCE_{m-1} to PCE_m takes $T_{m-1,m}^{TCP}$ time, and the processing time of the PCEP message by PCE^m is t_m^{proc}, then equation 3.1 captures the total time T elapsed from when PCE_1 makes a request and when a reply is received by it.

$$T = 2\sum_{m=1}^{N} t_m^{proc} - t_M^{proc} + \sum_{m=2}^{M} T_{m-1,m}^{TCP} + \sum_{m=2}^{M} T_{m,m-1}^{TCP} + \sum_{m=1}^{M} T_m^{comp} \tag{3.1}$$

The probability of LSP deployment failure can be obtained by equation 3.2 where tc_m is the computation time at which PCE_m is computing its local portion of the end-to-end path. td_m is the time at which the LSP is being deployed for the portion of the end-to-end path belonging to PCE_m's coverage. $Path\_Resources\_STATE_m(t)$ is the network's state at time

t for only the resources required by the optimally computed path in question.

$$1 - \prod_{M} Pr[Path\_Resources\_STATE_n(tc_m) = Path\_Resources\_STATE_n(td_m)] \quad (3.2)$$

Equation 3.2 assumes that the $Path\_Resources\_STATE_m(t)$ reflects the exact state of the network at time t. But, only fluctuations causing blockage are considered. This mean that $Path\_Resources\_STATE_m(tc_m) = Path\_Resources\_STATE_m(td_m)$ holds in cases where there are fluctuations in the network resources that concerns the LSP , but these fluctuations do not threaten its deployment. This equality also captures any insignificant increase of the used bandwidth or even the release of some resources.This is an important point to consider, for example if a connection admission control algorithm is proposed for inter-domain connections [2].

As the path computation times T_m^{comp}, the PCEP processing times t_m^{proc} and the PCEP communication times $T_{m-1,m}^{TCP}$ increase, the difference between times tc_m and td_m in equation 3.2 is emphasized, causing a greater uncertainty about the state of the critical resources. Due to networks dynamic nature, it is safe to assume that as this time difference increases so does the probability of resource fluctuations that could lead to blockage.

In practice, T_m^{comp} could vary in the orders of few tens to hundreds of milliseconds. The t_m^{proc} time depends on the load of the PCE and the pending requests (assuming a First in First out service). The $T_{m-1,m}^{TCP}$ time is dependent on network conditions and could vary in worse case scenarios from one second to a few seconds. As considered by equation 3.1, when the number of PCEs/domains increases, these times are added up and could become significant.

3.2 Proposed Inter-Domain Path Negotiation Scheme

The proposed PCE-based inter-domain path negotiation scheme prevents LSP deployment blockage by pre-reserving the resources in each domain as the inter-domain path is computed. This way when the computed path is used to signal the explicit LSP, resources along that path will be available because they have been pre-reserved at least for certain duration. This technique assures resource availability at deployment time and therefore reduces the probability of blockage. The subsequent sections present the proposed scheme in more detail and analyzes the response time.

2. The CAC algorithm shall not consider all resource fluctuation feedbacks as a possible blocking treats, otherwise hysteresis is not obtained. Thus, the use of threshold-high and threshold-low levels is recommended to estimate, with a certain confidence, the range of fluctuations that could be considered as acceptable and non-blocking.

3.2.1 Detailed description of the path negotiation procedure

The proposed procedure introduces two basic messages: *[Path/QoS request]* and *[Path/QoS reply]*. The functionalities of these basic messages can easily be incorporated in the PCEP protocol, more specifically into *PCReq* and *PCRep* messages. The *[Path/QoS request-confirm]* and *[Path/QoS reply-confirm]* correspond respectively to RSVP-TE *Path* and *Resv* messages exchanged during the deployment of an LSP. They could refer to other deployment signalling messages if the reservation/routing technology used is not G/MPLS. The *[Path/QoS reply-confirm]* message could be omitted if the reservation is initiated by the head node [3].

Each <path, QoS> request is composed of a source-destination path and the required QoS. The required QoS can be represented by the Request Parameters (RP) and/or Objective Function (OF) objects defined by Vasseur et LeRoux (2009) and by LeRoux *et al.* (2009) respectively. After pruning non-feasible paths, the PCE returns one or a set of feasible paths to the PCC. The PCC may have interrogated more than one PCE. In any case, it can select the best available path and signal the LSP.

This procedure requires that the domains exchange information about the computed paths, which might expose confidential details about the traversed domains. However as mentioned previously, the problem is solved by the use of path keys,defined by Bradford *et al.* (2009), which enable the domains to hide from other domains the sensitive information about internal path segments. This way only path entry nodes and path performance information is shared among domains. This selection can be based on local policies, cost, and/or QoS capabilities of the returned paths. The detailed algorithm is described in Figure 3.1. Figures 3.2 and 3.3 give a flowchart view of the procedures in each of the PCC and PCE nodes.

The key point here is that each interrogated PCE along the end-to-end path computes an optimal path within its domain of responsibility, and unlike other procedures, each PCE also pre-reserves the resources in its domain. Any RSVP like pre-reservation mechanism can be used to pre-reserve for a specific time the resources required by the optimal path. The patent of Verchere *et al.* (2006) defines the technicalities for RSVP node based pre-reservation at the data plane (physical or hardware pre-reservation). Control plane (software) pre-reservation is also a possibility. That second alternative could be implemented trough the TED. This works if the TED contains a complete up to date representation of resources including: the actual resource usage (real reservations), the pre-reserved resources, as well as the available resources. It is interesting to see that using the TED to keep record of the pre-reservations

3. The term *head* node is used to avoid using the terms *source* or *ingress* nodes which could infer about the direction of the traffic.

Origin PCC/PCCRouter:

START:
SEND a [Path/QoS Request] to selected PCEs THEN GOTO WAIT_FOR [Path/QoS Reply]

WAIT_FOR [Path/QoS Reply]:
WAIT for [Path/QoS Reply] from of or all PCEs THEN GOTO CHOOSE_OPTIMAL_PATH

CHOOSE_OPTIMAL_PATH:
CHOOSE optimal path among the ones available AND SEND
[Path/QoS Request Confirm] only to the PCE responsible for the chosen Path

ALL PCEs/PCERouters:

START: WAIT_FOR [Path/QoS Request]:
WAIT for [Path/QoS Request] or [Path/QoS Request Confirm] from PCC
THEN GOTO MANAGE [Path/QoS Request] or MANAGE [Path/QoS Request Confirm]

MANAGE [Path/QoS Request]:
IF the destination is not in the PCE's scope,
THEN send [Path/QoS Request] to neighboring PCEs AND GOTO WAIT_FOR
[Path/QoS Reply]
ELSE
Compute PATH and SET RSV_TIMER on its resources and RETURN the
[Path/QoS Reply] to requesting PCC

MANAGE [Path/QoS Request Confirm]:
IF the destination is not in the PCE's scope,
THEN RESET RSV_TIMER on resources for that Path ID THEN forward
[Path/QoS Request Confirm] to neighboring PCE AND GOTO WAIT_FOR
[Path/QoS Reply Confirm]
ELSE
RESET RSV_TIMER on resources and RETURN the [Path/QoS Reply Confirm] to
requesting PCC

WAIT_FOR [Path/QoS Reply]:
WAIT FOR [Path/QoS Reply] from one or all PCEs
THEN GOTO MANAGE [Path/QoS Reply]

WAIT_FOR [Path/QoS Reply Confirm]:
WAIT FOR [Path/QoS Reply Confirm] from PCE
THEN GOTO MANAGE [Path/QoS Reply Confirm]

MANAGE [Path/QoS Reply]:
CHOOSE THE OPTIMAL REPLY if more than one THEN Compute local PATH and SET
RSV_TIMER on resources and RETURN the overall [Path/QoS Reply] to requesting
PCC
THEN GOTO WAIT_FOR [Path/QoS Reply Confirm]

MANAGE [Path/QoS Reply Confirm]:
RETURN the [Path/QoS Reply Confirm] to requesting PCE(PCC)

Figure 3.1 Functional operation of the PCE and PCC in the proposed scheme

allows for additional information to be appended. One such information could be the requests' priorities. For example, this allows a request with higher priority to undo pre-reservations of a lower priority in cases where the available resources are not sufficient.

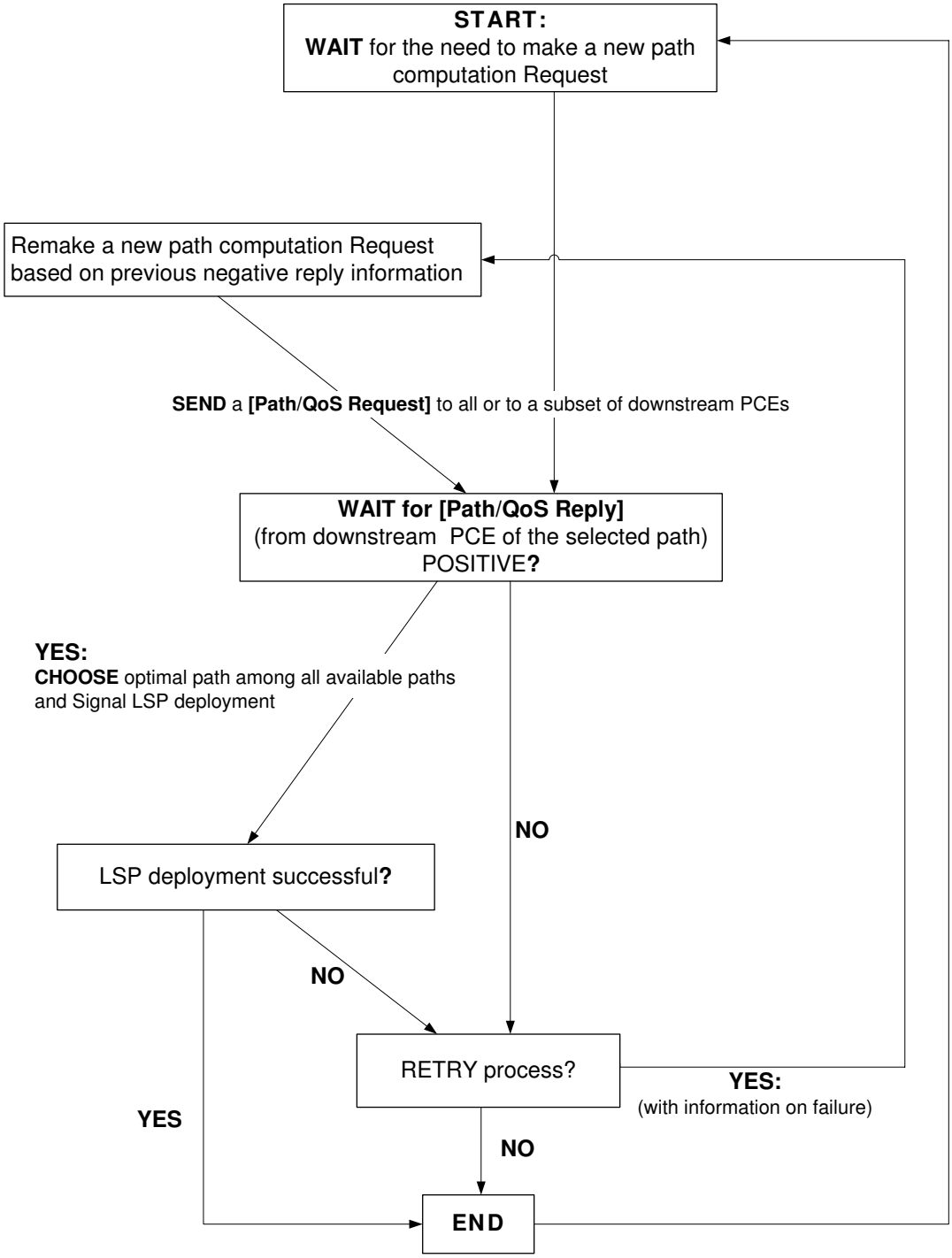


Figure 3.2 Operational flowchart of PCC in the proposed scheme

Irrespective of hardware or software based pre-reservations, their duration should be long enough and not expire until the resources are reserved at the data plane level by the deployment of the LSP. At the same time, they should not hold for too long, in order to avoid

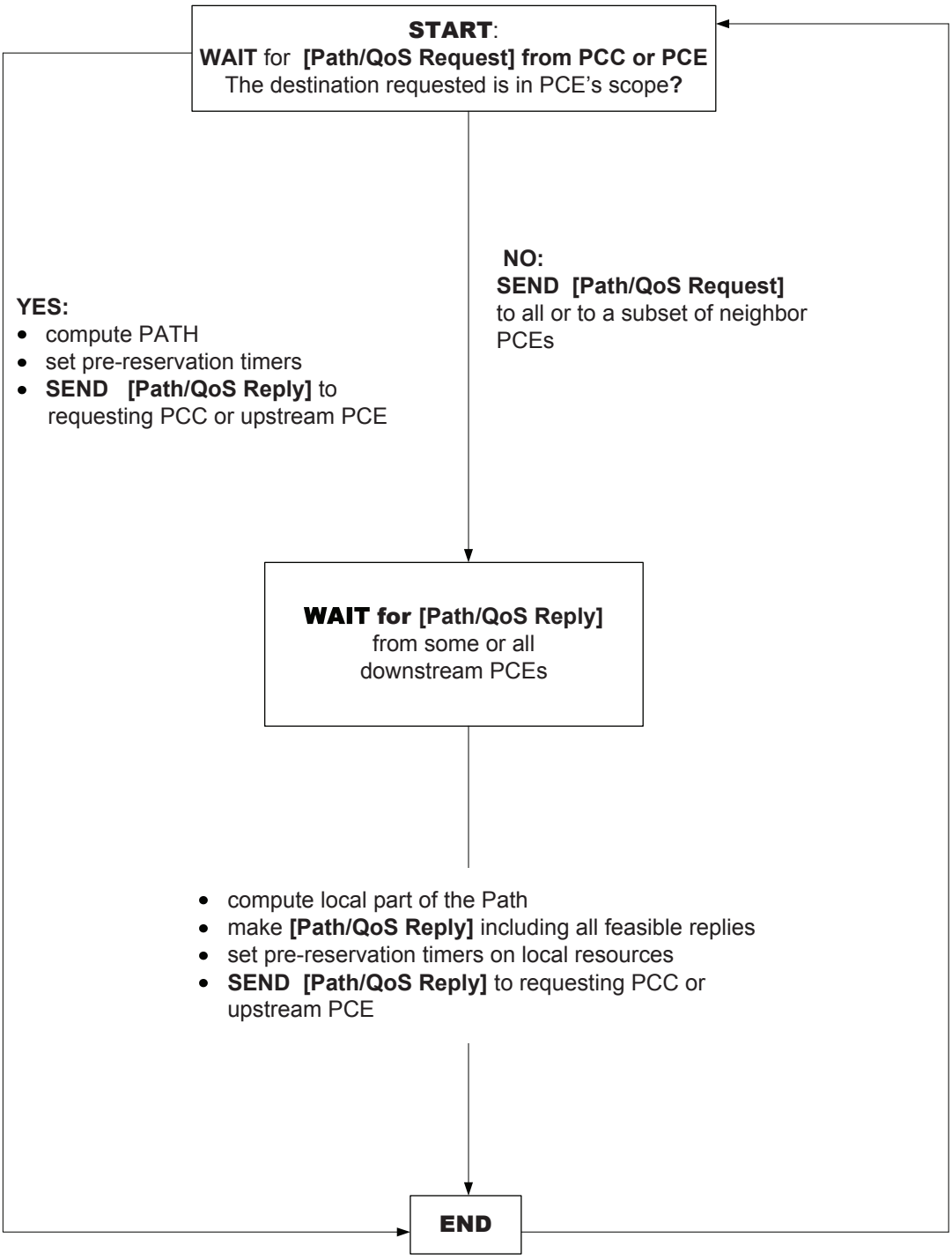


Figure 3.3 Operational flowchart of PCE in the proposed scheme

blocking other requests. This could become a real problem if more than one path computation request is sent out for the same path and resources are pre-reserved along each computed path. This is often the case for inter-domain scenarios like the one proposed here. There-

fore, considering that the proposed solution makes use of pre-reservations on all the possible optimal paths during the computation process, it can be argued that with pre-reservations, unused pre-reserved resources are wasted which could cause the failure of other path requests requiring those same resources at that time. However, it can also be argued that PCE resources are wasted if, at the time of the LSP deployment, resources are no longer available. The simulation results presented in section 3.3 will bring light to this concern.

Clearly, it is very important to make the pre-reservations for the right durations. The solution proposed in this work uses a timer when making the pre-reservations. This timer can expire by its own or upon reception of a tear down or cancellation message [4]. A timer expiring on its own refers to, what is called a soft pre-reservation. Timers awaiting a tear down message to reset the resources refer to hard pre-reservations. The use of soft pre-reservations avoids the propagation of too many tear down messages. However this leads to the problem of correctly setting the expiring timers to avoid the aforementioned problem of under or over pre-reservation (timers expiring too soon or too late). To counter under pre-reservation (timers expiring too soon), refresh messages are required. At the inter-domain level this might not be a suitable solution due to the numerous messages that will need to be exchanged. Thus, a hybrid timer resetting scheme is recommended to allow hard pre-reservations releasing resources upon the reception of a message or after the expiration of a timer. This technique can also account for the loss of a tear down message. The idea of further investigating the use of soft pre-reservations is left for future work as detailed in section 6.3. This said, it is important to note that the proposed scheme is equally valid for any type of pre-reservation being hardware or software, hard or soft.

Another interesting concept about this inter-domain solution is that even though it seems like a flat PCE topology, there are actually minimum two levels of PCEs: the intra-domain PCEs and the inter-domain or domain boundary PCEs. Therefore the solution is hierarchical in nature and allows for eventually more than two levels . For example a consortium of domains (ASes) may opt for a central PCE connected to their domain boundary PCEs, in turn connected to their area boundary PCEs and so forth. One possibility is that different consortia could form among network providers allowing different levels of PCE connection within this hierarchy.

4. The term *tear down* is used if the pre-reservation is hardware based at the data plane level, the term cancellation is used for software based pre-reservations at the control plane. Both terms are used interchangeably in this thesis.

3.2.2 Example of the path negotiation procedure

The description can be better completed with the help of an example. Figure 3.4 shows a PCE network where the PCEs and ASBRs are in the same node for sake of clarity. Internal routers inside each AS are not shown for the same reason.

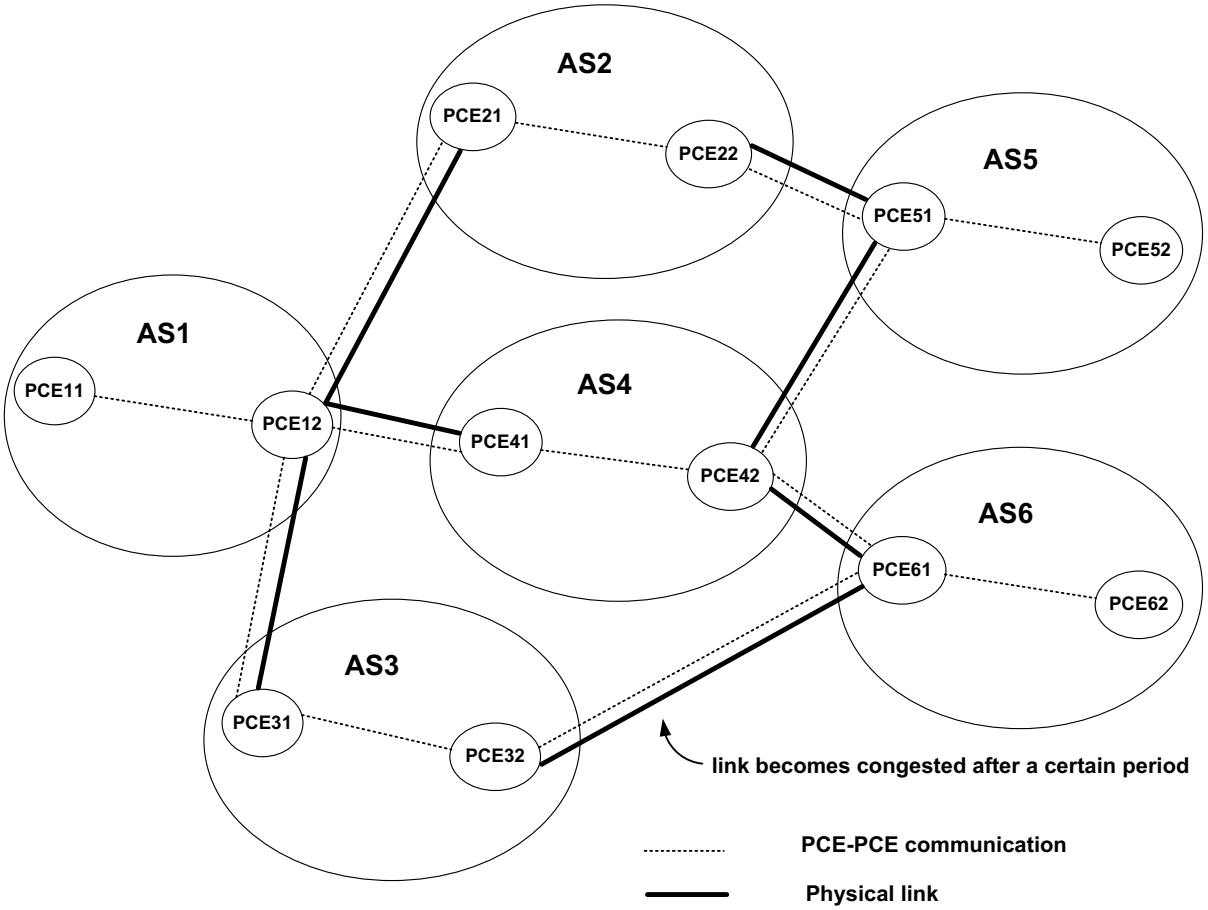


Figure 3.4 Network used as an example for the description of the proposed path negotiation procedure.

In this example, PCE_{11} is initially triggered by a PCC (not shown). It forwards to its neighbours the <path, QoS> request towards the destination router/PCE_{62} . In this example, PCE_{11} receives three replies, has to select the best inter-domain path between the positive replies and signal the LSP deployment.

Figure 3.5 shows the sequence of messages exchanged between PCEs until the end-to-end path is computed. For clarity reasons, the *[Path/QoS request]* messages are only shown for the first PCEs in each of AS2,AS3, and AS4. However these message are forwards

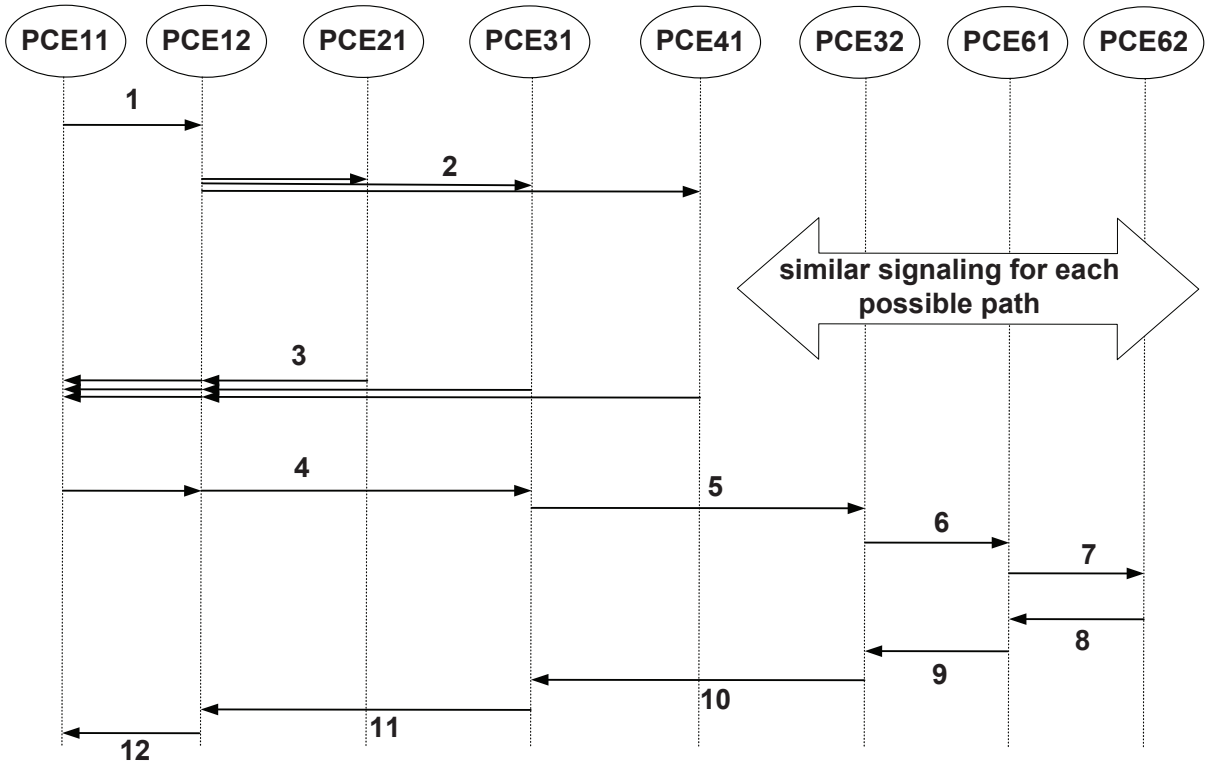
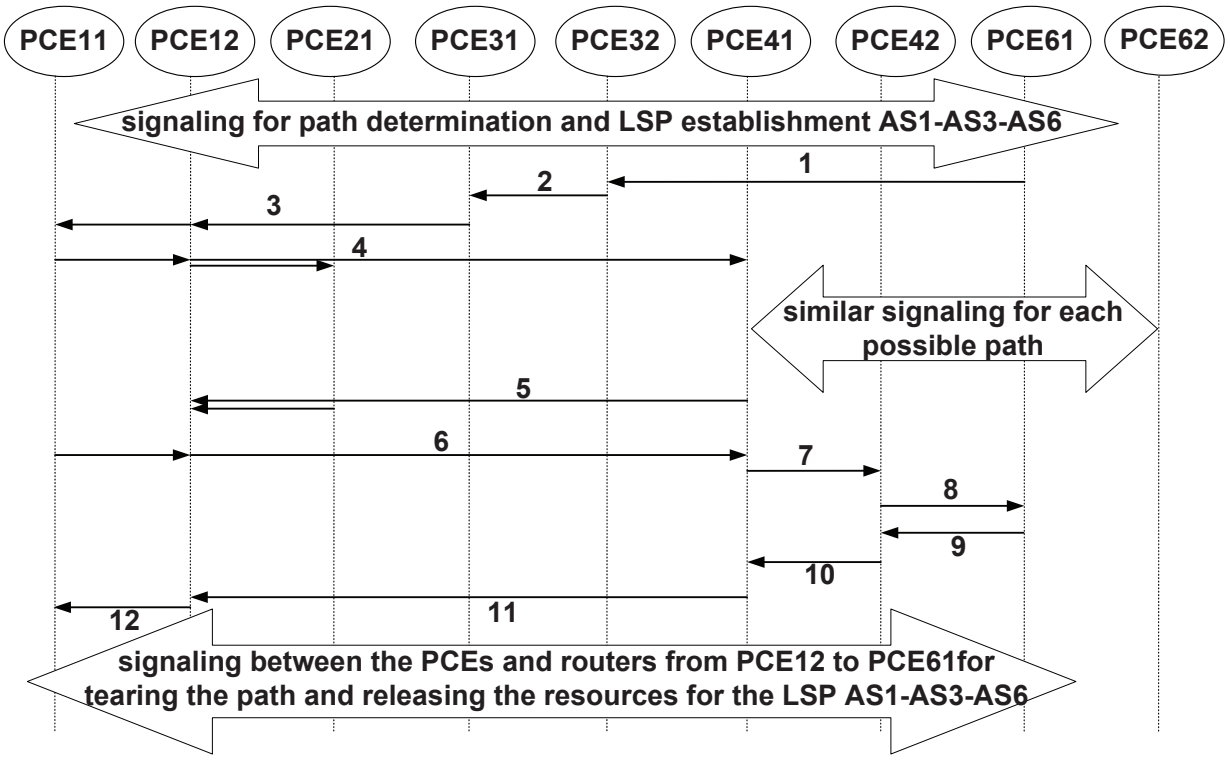


Figure 3.5 Signalling messages for the computation and establishment of an inter-domain path

up to PCE_{62} for the three end-to-end paths of AS1→AS3→AS6, AS1→AS2→AS5→AS6 and AS1→AS4→AS6. The same simplification is done for the *[Path/QoS reply]* messages. Figure 3.5 only shown these messages between PCE_{21}, PCE_{31} and PCE_{41}. These messages are in fact returned from PCE_{62} through the three end-to-end paths mentioned above. In this example it is assumed that PCE_{11}, upon reception of the *[Path/QoS reply]* messages, decides that the best path is the one returned from PCE_{31} which goes through the AS1→AS3→AS6 path. Then as shown in Figure 3.5, messages 5 to 12 are only exchanged between the PCEs involved in this end-to-end path.

Figure 3.6 shows the case where a path re-computation is triggered by a node, PCE_{61} in this example, upon detecting performance degradation. Here a *[Notify State_Change]* message notifies PCE_{11} that a new path is required, due for example to QoS deterioration. PCE_{11} requests a new path towards PCE_{62} but this time, given the information carried in



1-3 :[Notify State_Change], 4 :[Path/QoS request]
5 :[Path/QoS reply]
6-8 :[Path/QoS request-confirm] (Assuming the path found by PCE41 is optimal)
9-12:[Path/QoS reply-confirm]

Figure 3.6 Signalling messages for the re-computation and re-establishment of an inter-domain path

the *[Notify State_Change]* message, it makes the request from PCE_{21} and PCE_{41} for the end-to-end paths AS1→AS2→AS5→AS6 and AS1→AS4→AS6. This is just a choice for this particular example. Similarly to the previous example, upon reception of the *[Path/QoS reply]* messages, PCE_{11} could decide that the best path is the one returned by PCE_{41} and request the deployment by the *[Path/QoS request-confirm]* message.

3.2.3 Loop prevention mechanism

Looping is a well addressed subject in networking and often refers to a routing loop which occurs when a packet is forwarded endlessly without reaching its destination router. Looping can also refer to control messages, like a Label Request message in G/MPLS, which loops across the network due to routing protocol misconfiguration or erroneous explicit route. Within the PCE environment, the same control message routing risk appears. Indeed the

PCReq message has to be routed across PCEs in a way that prevents loops. By prevent it is meant that loops are not allowed, and not left to be detected afterwards. *PCReq* message loops should be prevented to avoid wasting PCE resources. The PCE working group at the IETF has not yet tackled loop avoidance issues as they currently assume that the PCE sequence is determined in advance (Farell *et al.*, 2006) and loops are avoided by policy. However, LeRoux (2007) mentions the risk of *PCReq* loops and the need for a solution.

The proposed scheme is implemented with a simple yet effective loop detection mechanism which consists in carrying a PCE Path Sequence Object (*PCPSO*) in the request messages (*PCReq*). The *PCPSO* is simply a list of all PCE nodes that have already received and processed this *PCReq* message. Each PCE node receiving a request, first verifies that its node ID is not already present in the *PCPSO*. If present, a loop is detected and a reply message (*PCRep*) with loop error is sent back to the requester. If the node ID is not present in the *PCPSO*, the PCE will add its own ID to the list. This mechanism can easily be added to the PCEP protocol. It is important to mention that this loop prevention mechanism only works for inter-domain path computation schemes that consider all domain entry and all exit points for each request. If that is not the case, the information in *PCPSO* object is not sufficient. In fact, the solution to a general loop prevention mechanism is very complicated. This is due to the nature of the problem that PCEs are not necessary routers nor are part of the final traffic route, which needs to be loop free as well. A PCE can in fact be solicited many times for the same request (this is not recommended but is possible depending on the scheme used). In the general case, this situation should be allowed and not detected as a *PCReq* loop. Also for the same request, different *previous* PCEs, can solicit the same PCE where the request message could ask for a different path segment (source-destination pair). Again it is interesting to note that such a case should not be considered as a *PCReq* loop. Thus, the loop prevention method proposed in this chapter is only valid for cases where the scheme considers all domain entry and exit points for each request.

3.3 Performance Evaluation of the Proposed Inter-Domain Path Negotiation Scheme

A simulator is developed to evaluate the performance of the proposed scheme. The simulator is written in JAVA using the *java development kit* (jdk) version 1.6.0_07. The choice of the language is in part due to the Remote Method Invocation library of the language that allows TCP communication between hosts. This allows the transformation of the simulator into a test bed with less effort. Another reason for the choice of the language is its Thread library that gives predictive thread behavior. The simulations are run on a AMD Opteron(tm)

Processor 150 machine with a CPU of 2393.220 MHz, 2GB of RAM, and running Fedora Core release 4 with LSB VERSION 1.3 (2.6.13-1.1526_FC4).

3.3.1 Simulation settings

The performance of the proposed scheme is evaluated by comparing it to the non pre-reservation version. Three performance parameters are measured. The *PCReq success* parameter designates the ratio of successful replies (*PCRep*) to the maximum number of deployable LSPs. The *LSP deployment success* parameter designates the ratio of successful deployments to the total number of LSP deployments initiated. This parameter is the most important one when evaluating the benefits of the pre-reservation solution regarding blockage. The *Overall success* parameter gives insight about the network utilization. It is simply the ratio of successfully deployed LSPs to the maximum number of deployable LSPs. Here the maximum number of deployable LSPs is pre-determined for each scenario before running the simulations. The *LSP deployment success* parameter is the ratio that the proposed scheme is intended to improve. As it approaches 100%, it can be concluded that the resources taken by the PCEs to compute the end-to-end path have not been wasted by resulting in a blocked LSP deployment. It is important to point out that in a real life scenario, a successful reply followed by an unsuccessful deployment will usually result in a subsequent request being made, and thus replicating the amount of work performed by PCEs for the same LSP.

The simulator does not implement the handling of blocked LSPs, i.e., no re-computation of the path or re-deployment of it is initiated. Representative average results, obtained from a minimum of ten runs, are discussed below, along with some important outcomes that can direct future works. For the obtained results, the QoS criteria considered is the number of hops crossed by the path. The number of hops is often a routing criterion in *all* optical networks where a light path can only take a maximum number of hops to avoid power loss and signal degradation (Leblanc *et al.*, 1999). The number of hop criteria is additive. Therefore, as the paths are computed in the simulations, any path that has exceeded the maximum number of hops criteria is discarded. If a PCE does not find any path that does not violate this criteria, it simply returns a NO-PATH error to the requester with specific information about the failure, as described in Vasseur et LeRoux (2009). Moreover, since no pre-defined sequence of PCEs is determined, each PCE which has no visibility on the destination forwards the request to all neighbour PCEs when the request is for a node in another AS. If the request is for a node inside the same AS, it will be forwarded only to intra-AS neighbours. Consequently the loop prevention mechanism described in section 3.2.3 is implemented to prevent the occurrence of request message loops.

Topology

The simulations are performed on the COST266 topology (Hancock, 2006), a real world network, with 28 domains and 57 bi-directional inter-domain OC48 links (Figure 3.7). Intra-domain behavior is simulated uniformly for all domains, to abstract away unnecessary details and focus on the inter-domain PCE procedures.

Demand matrices

Table 3.2 describes the simulated scenarios. It specifies the total number of requests, the number of PCCs and the bandwidth per request for each scenario. Both normal (10 requests per second) and heavy (100 requests per second for a 10 second interval) rates of incoming requests per PCE are simulated. The demand matrices are of 2500 requests for scenario I; 3000 requests for scenarios II, III and V; and 600 requests for scenario IV.

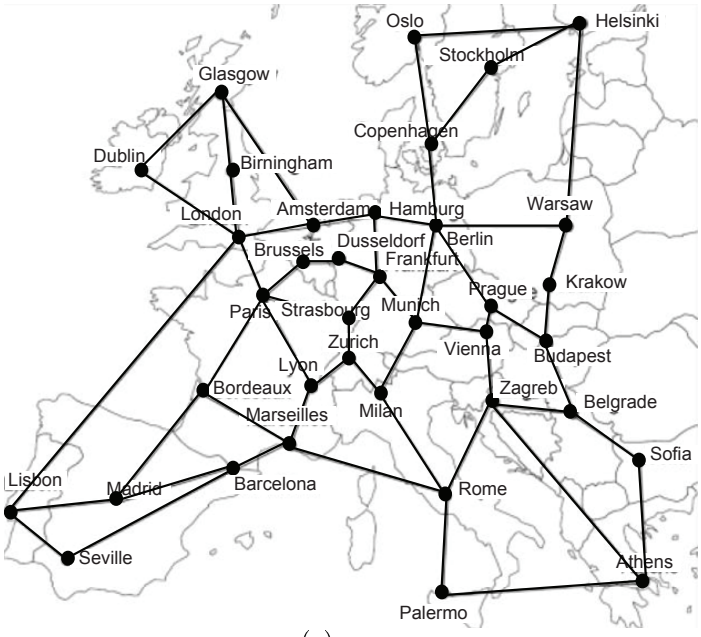
Table 3.2 Simulation scenarios

	Scenario				
	I	II	III	IV	V
Num of Req	2500	3000	3000	600	3000
Num of PCCs	4	15	15	15	15
BW(Mbps)	20	20	50	50	250

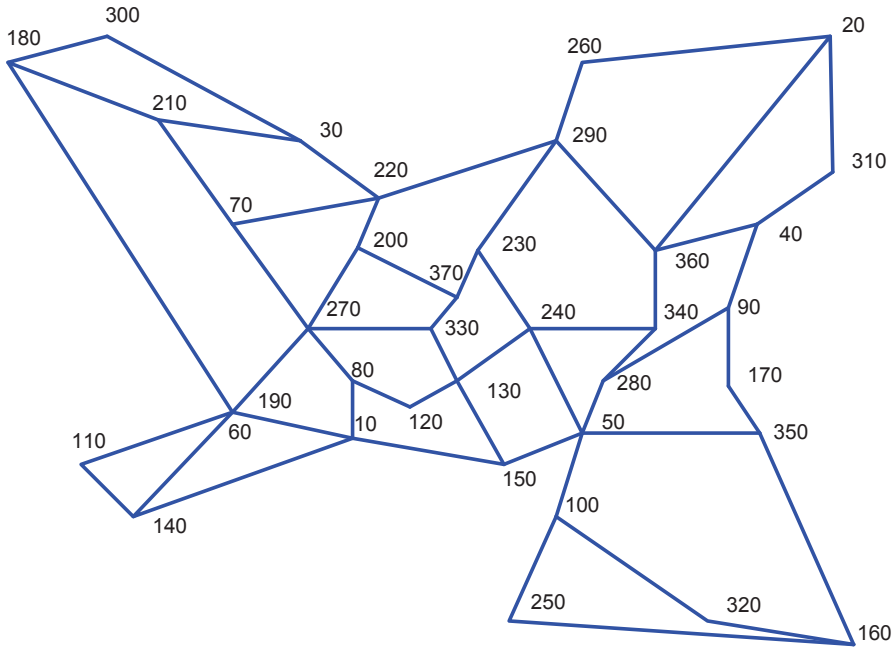
3.3.2 Simulation results

Figures 3.8 to 3.12 show average performance scenarios of the pre-reservation based scheme in comparison to the non pre-reservation method. The demand set used in scenario I (Figure 3.8) has all 2500 requests concentrated between fewer source-destination pairs. The demand set of scenarios II to V contain requests between more diversified source-destination pairs. The demand sets of scenario I and II have requests of 20Mb each; the demand sets of the scenario III and IV have 50Mb requests, and that of scenario V has 250MB requests. The pre-reservation timers are optimally chosen, based on experiments. Optimal pre-reservations last 11 seconds, to 14 seconds. In all scenarios, the maximum reply time is of 10 seconds.

The results suggest that the pre-reservation scheme performs better in all scenarios. By comparing scenario III and IV it is interesting to note that the pre-reservation method is even more beneficial when resources are scarcer. Figures 3.11 and 3.12 are the under-utilized and over-utilized scenarios respectively. It is seen in overall that the pre-reservation method is more beneficial when PCE request messages ($PCReq$) arrive at higher rates. A $PCReq$



(a)



(b)

Figure 3.7 The COST266 network: geographical and graph views

success ratio above 100% shows the need of pre-reservations to avoid over-estimating resource availability when computing paths.

The *overall success* parameter shows slight improvement when using the pre-reservation

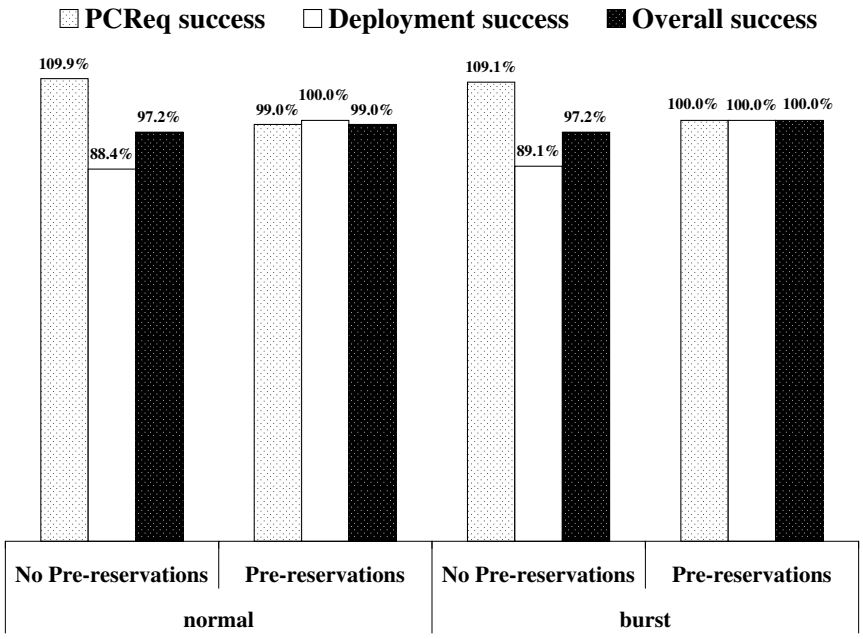


Figure 3.8 Comparison of the pre-reservation versus non pre-reservation version of the proposed scheme, Scenario I

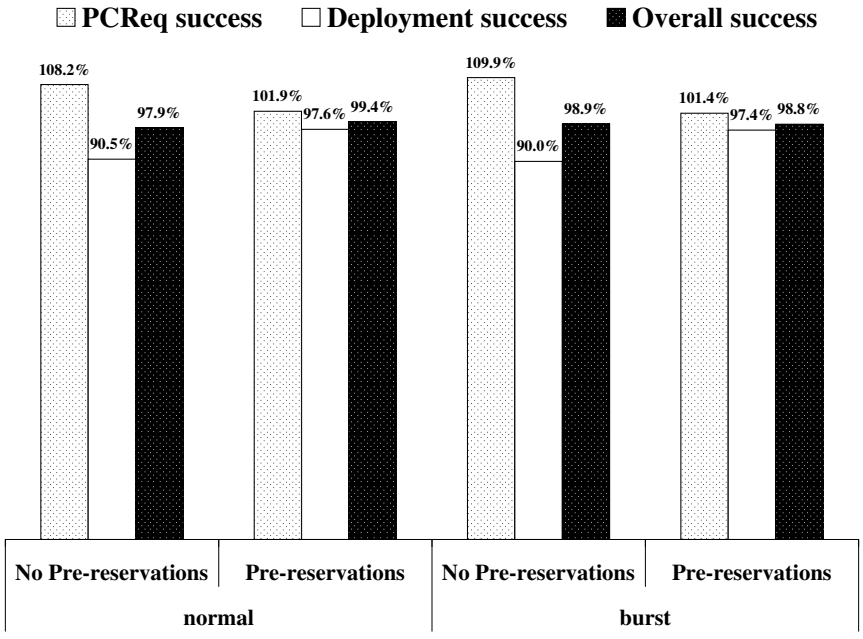


Figure 3.9 Comparison of the pre-reservation versus non pre-reservation version of the proposed scheme , Scenario II

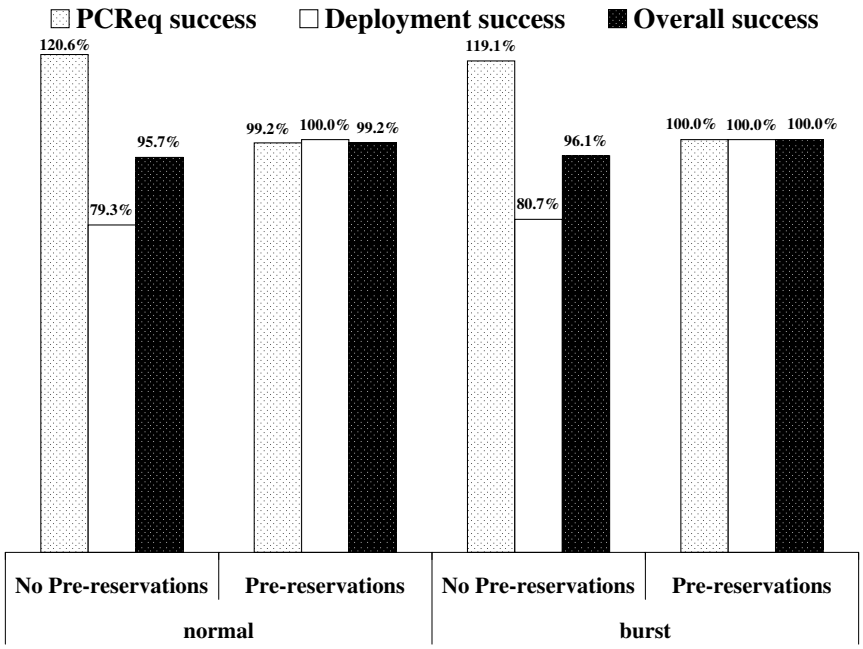


Figure 3.10 Comparison of the pre-reservation versus non pre-reservation version of the proposed scheme, Scenario III

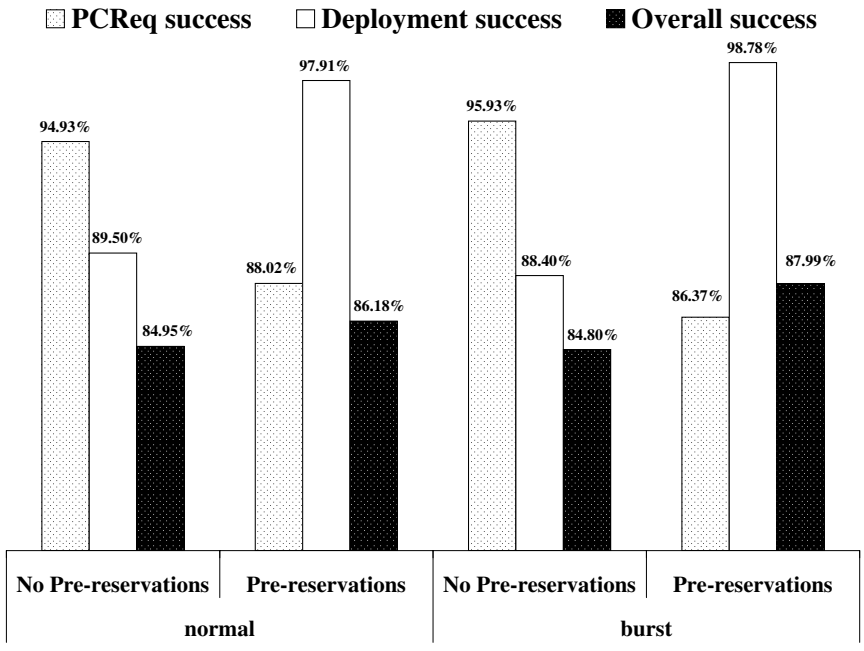


Figure 3.11 Comparison of the pre-reservation versus non pre-reservation version of the proposed scheme, Scenario IV

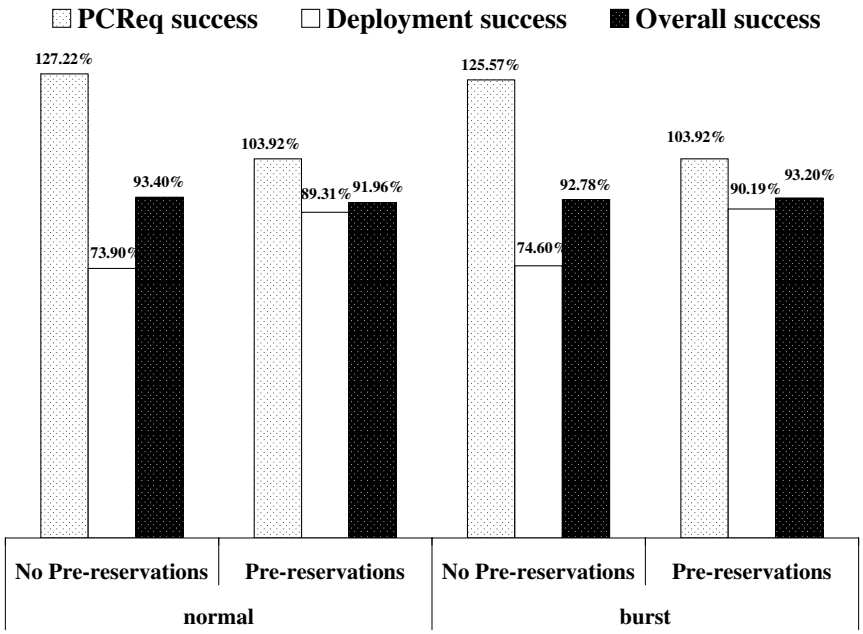


Figure 3.12 Comparison of the pre-reservation versus non pre-reservation version of the proposed scheme, Scenario V

method. This is due to the significantly large number of requests being made in the simulations, compared to the maximum number of LSPs that can be deployed, i.e., the available bandwidth. When a smaller set of requests is considered (i.e., a number close to the maximum number of deployable LSPs), the non pre-reservation procedure results in a lower *overall success* ratio.

Figure 3.13 shows the effects of pre-reservation time on the performance parameters for scenario III described previously. The overall response time for a request is fixed to maximum 10 seconds. The pre-reservation timer varies from 0 to 20 seconds in Figure 3.13 and from 0 to 190 seconds in Figure 3.14. A pre-reservation time of 0 seconds is equivalent to the non pre-reservation counterpart. It is seen that in cases where the pre-reservation time is lower than a certain value, the scheme performs slightly worse than the non pre-reservation counterpart because the resources are not only unused by the actual LSP request for which they are held, but also kept unavailable for any other LSP path computation attempt.

However, as the optimal pre-reservation time is reached, it is seen that the proposed scheme outperforms the non pre-reservation counterpart and behaves perfectly with almost 100% ratios for all three performance parameters. It is important to note the negative effects of longer pre-reservation times on the *PCReq success* ratio and the *Overall success* ratio. Determining the minimum optimal pre-reservation time is subject of ongoing work.

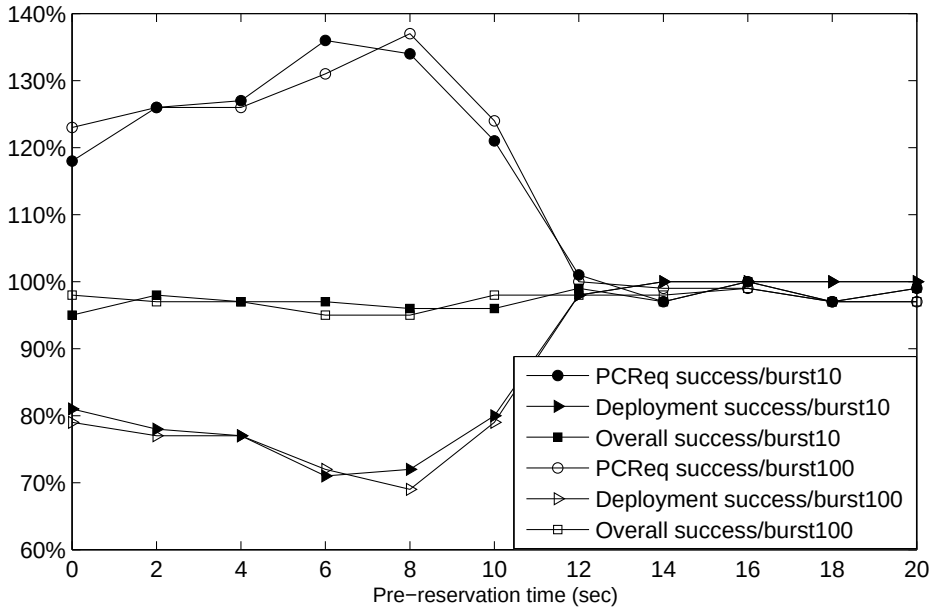


Figure 3.13 Effects of pre-reservation time on the performance parameters in and convergence towards the optimal pre-reservation time (Scenario III)

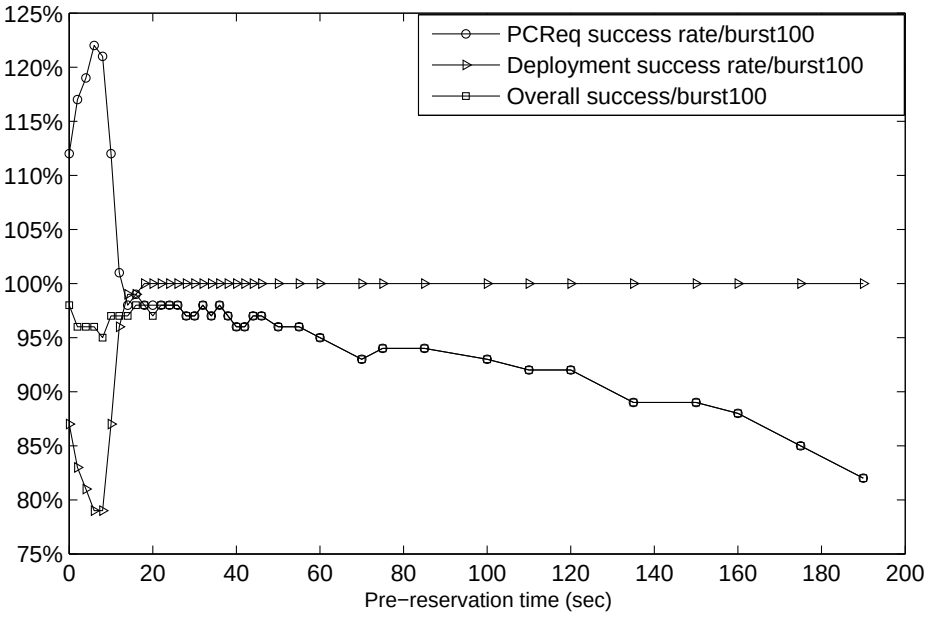


Figure 3.14 Effects of pre-reservation time on the performance parameters in and consequences of longer pre-reservation times (Scenario III)

These results also bring light to the dilemma between deployment blockages due to resource

fluctuations and PCE path computation failures due to numerous pending pre-reservations.

3.4 Summary

This chapter first presents the factors that could affect the response time of PCEs to an optimal path computation request. It also derives the cumulative effect of response times in an inter-domain environment. This derivation leads to the intuitive conclusion that as the path computation response time increases, so does the risks of resource fluctuations and thus LSP deployment blockage.

Then, an inter-domain path negotiation scheme is proposed to allow for optimal path computation in a multi-AS environment. To solve the problem of high blockage risks in such scenarios, the solution includes a pre-reservation of resources at computation time. The solution of pre-reservation at computation time is also valid to any other path computation scheme. Moreover, a loop prevention mechanism is designed to avoid the looping of Path Request messages among PCEs.

Simulation results support the argument that blockage could become a serious problem in an inter-domain PCE environment. The results also give conclusive insights in the benefits of the proposed scheme and show that using pre-reservations is a good solution. According to the simulation results, the solution achieves lower blocking probability at LSP deployment time.

CHAPTER 4

JOINT INTER-LAYER/INTER-DOMAIN TRAFFIC ENGINEERING

As mentioned in section 2.2, networks are inherently multi-layer, a reality that is often overlooked in traffic engineering due to the complexities it brings to the problem. It is also established that, to effectively compute end-to-end optimal paths, the multi-layer nature of telecommunication networks has to be considered. Most importantly, it was mentioned that in reality, a mixture of inter-domain and inter-layer scenarios may occur when considering the end-to-end connectivity of a path. In fact, some network operators may own resources only at one layer, while others may own resources at two adjacent layers, and other major network operators may own resources at all layers.

This chapter focuses on the inter-layer aspect of traffic engineering, in particular path computation, while considering that the actual real world problem is usually a mixture of an inter-layer and inter-domain scenario. This is addressed by proposing a PCE based solution, where the inter-domain part of the problem is mainly solved by the use of the PCE architecture itself and methods similar to the solution of chapter 3.

As presented in section 2.2, most of the work in the inter-layer area focuses on a centralized optimization of inter-layer resources. In reality, this is not always possible, because of scalability issues, of confidentiality among domains (when a layer belongs to a different provider), and restrictions due to internal administrative policies. The latter is very similar to the inter-domain confidentiality issue but applies to a given administrative domain. Here, the problem is not the actual disclosure of resource information, but rather the management rights attributed to different groups. Usually within an administrative domain, different layers are managed by different internal groups. For example, the transport network management group may not allow the IP network management team to intervene in the management of resources at their layer. Even though to reduce the OPEX costs, the tendency is to gradually shift towards a unique management plane, each layer may still have different performance objectives and policies. Thus, opting for a centralized inter-layer solution with the hope of a global view on all layers leads to unrealistic solutions for current OAM practices.

Notwithstanding, the question remains to find a way to consider all layers when computing the end-to-end traffic engineered paths. One natural solution is to extend the proposed method of the previous section, where each layer can represent a domain. Since the proposed

scheme is defined within the PCE architecture, GMPLS is the technology employed to implement this solution. Another reason is that GMPLS is specifically defined for the purpose of inter-layer traffic control.

As a comparison to chapter 3, the inter-domain solution in a G/MPLS environment resulted in a number of per domain LSPs stitched or nested together to form the end-to-end path. In the inter-layer solution, lower layer LSPs will contain, in a nested manner, the LSPs of the higher layers. This is similar to the physical resources being nested into each other (refer to Figure 2.5). One fundamental difference between the inter-domain and the inter-layer path requests is that the former traverses an undetermined number of domains while the latter has a definite number of layers. This will be discussed further along with other major differences.

This chapter treats the multi-layer/multi-domain facet of the problem. First, section 4.1 starts by showing the relevance of multi-layer/multi-domain traffic engineering and path computation. Then, section 4.2 describes the proposed multi-layer/multi-domain and path computation scheme and relates it to the inter-domain procedure of chapter 3. Subsequently, section 4.3 presents the proposed traffic engineering scheme that implements the multi-layer/multi-domain path computation mechanism and uses traffic predictions. Then, section 4.4 presents analytical and simulation results that evaluate the proposed scheme. Finally, section 4.5 summarizes this chapter by highlighting its main contributions.

4.1 Relevance of Joint Multi-Layer/Multi-Domain Path Computation

Chapter 1 stresses on the importance of considering real world scenarios where a single transport network can serve different higher layer networks from various organizations. Furthermore, higher layer networks can be served by more than one transport networks. For the end-to-end criterion in traffic engineered path computation, this implies that inter-layer and inter-domain problems are not completely separate. This section gives a more in depth explanation of this problem and establishes the need for their joint consideration.

In the example depicted in Figure 4.1, it is assumed that a new path request is made between the source Node_S and the destination Node_D. Two routes are possible, one from Node_S to Node_B to Node_D, the other from Node_S to Node_A to Node_D. Interestingly, these layer_{n+1} routes are carried on different transport networks offering the layer_n connectivity service. Moreover, in this example, the two transport networks do not belong to the same administration. Now, a globally optimal path may be found by computing the optimal layer_{n+1} path while considering all layers. This can be done by interrogating the layer_{n+1}

and layer$_n$ resource controllers (e.g. PCEs).

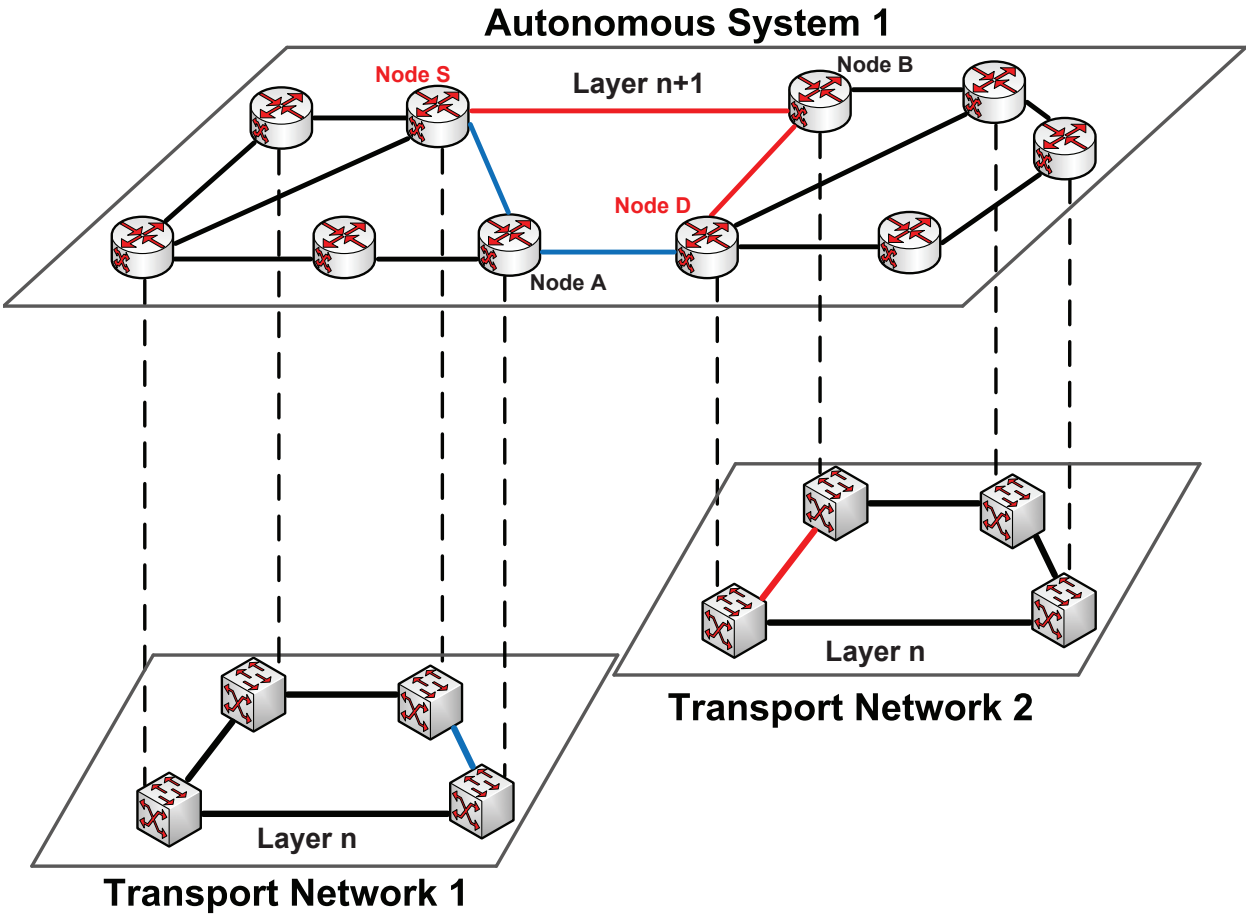


Figure 4.1 Example of upper layer using resources of two different lower layer providers

In this example, layer$_{n+1}$ link weights (e.g. OSPF like link weights equal to the inverse of the available capacity) are used to compute a shortest path from Node$_S$ to Node$_B$ to Node$_D$. Then, it is assumed that the current cost of this shortest path is equal to cost$^1_{n+1}$, if layer$_{n+1}$ alone is considered. By investigating further, that is considering layer$_n$ resources as well, it is found that if new layer$_n$ connections in transport networks 1 and 2 are provisioned, the shortest paths total costs at layer$_{n+1}$ would be cost$^2_{n+1}$ and cost$^3_{n+1}$ respectively. Here, the provisioning cost of a new layer$_n$ path is included in cost$^2_{n+1}$ and cost$^3_{n+1}$. In this example, it is assumed that cost$^3_{n+1}$ < cost$^2_{n+1}$ < cost$^1_{n+1}$. Therefore, it makes more sense to signal (provision) a new layer$_n$ connection (inter-layer and inter-domain) in transport network 1, then signal a new layer$_{n+1}$ connection on this new capacity for a total cost of cost$^2_{n+1}$. This new layer$_{n+1}$ connection can now be used for the deployment of the path from Node$_S$ to Node$_D$. Another variation of this example could be similar to the one presented above,

where Figure 4.1's AS would also own transport network 1. The same explanations and assumptions as above will still hold.

It is important to notice that the inter-domain aspect mentioned here is considerably different as the neighbour domain does not offer part of the end-to-end connectivity, but rather offers part of the lower layer connectivity required by the upper client layer.

4.2 Proposed Inter-Layer/Inter-Domain Path Negotiation Scheme

The proposed PCE-based inter-layer path negotiation scheme consists of a *distributed* solution for finding traffic engineered paths while considering multiple layers. Again, multi-layer path computation could also be performed if a *central* node had a complete view of all resources at all layers, but as mentioned above, this assumption is often not always realistic and an overlay model is often more suitable. As previously introduced in section 2.2, the GMPLS control plane supports an overlay model, an augmented model, and a peer model. Currently, GMPLS is more suitable for controlling each layer independently (overlay model). In the future, with the GMPLS single control plane paradigm, this management gap may be removed to form a single management plane with a complete view and control over all layers (peer model). Even so, it is needless to re-mention that in some cases the inter-layer and inter-domain problems are to be dealt with simultaneously, because not all ISPs own their transport networks. Therefore the overlay model will always need to be supported. These matters are discussed further in section 4.2.1 below. Section 4.2.2 describes the details of the proposed inter-layer/inter-domain procedure and the signalling involved. Section 4.2.3 discusses the instability risks associated with multi-layer traffic engineering. Section 4.2.4 compares the inter-layer scheme to its inter-domain counterpart. And finally section 4.2.5 discusses the risks of PCEP message loops in a multi-layer setting.

4.2.1 Reason behind a distributed solution

The reasoning as to why a distributed procedure for the path computation may be beneficial is somewhat similar for both the inter-domain and the inter-layer cases. However, slight differences exist in the interpretation of the arguments. In the inter-domain case, the main reason behind a distributed solution is the visibility/confidentiality issue which, along with the scalability requirement, does not allow a central entity to have a global view of all domains internal resources. This reasoning also applies to the mixture case of inter-layer and inter-domain scenarios where one layer may use at a lower layer the services of one or more different transport network operators. This situation was depicted in Figure 4.1 above.

Even if the GMPLS framework ultimately aims at a peer model where global visibility across technology layers is needed, the idea is not always applicable if the higher layer's network is not an established carrier with its own transport network. In these cases, it is inconceivable that operators would agree to have an outside entity (e.g. a PCE from another layer of a client network) view and participate in the control of their internal resources. Thus, the GMPLS peer model is only applicable for large providers who own the resources at every layer and agree to merge different capacity management groups into a single group. For others, a GMPLS overlay model might be the only possibility. Thus, particularly for these cases, the distributed solution is the only option if inter-layer path computation is to be considered in a realistic manner.

Currently, the mixed inter-layer/inter-domain problem is not an unusual situation because a very big percentage of today's ISPs (more than 90 percent) use more than two different transport network providers. Thus, the inter-layer path computation mechanism may trigger inter-domain path computation procedures. Today, this provisioning is done statically with human intervention. The procedure takes usually from a few days to a few months. The proposed solution of this chapter automates the procedure and provides dynamic multi-layer provisioning means within, and between domains.

The other argument is scalability. Usually multi-layer traffic engineered path computation problems are extremely complicated even for small non-realistic networks (Huang *et al.*, 2006). Network scalability problems are usually solved by *partitioning* or by *layering*. These two techniques are orthogonal in the sense that partitioning is a horizontal process while layering is a vertical process. Therefore, they can often coexist. The choice as to when to opt for one or another depends on the context. If scalability can be achieved by dividing the problem into smaller *similar* sub-problems, then partitioning is the right choice. This is the case of the inter-domain path computation problem where each domain is responsible for a similar sub-problem, from a technological point of view. If the problem can be divided into smaller but *different* sub-problems, then the best solution is layering. This is the case of inter-layer problems where the nature of the complexity in each layer is slightly different due to various technologies, granularities and optimization/management goals.

4.2.2 Detailed description of the proposed inter-layer/inter-domain path negotiation procedure

The main idea with the proposed inter-layer/inter-domain path negotiation procedure is to consider and evaluate all or a subset of possibilities in order to allow the selection of the optimal inter-layer path. This is achieved by a distributed process where PCEs will trigger

each other for path requests in a recursive manner. The PCEs considered and discussed here are mono-layer, i.e., they only have visibility into their respective layer. The solution is however not limited to this and can be applied to cases where some PCEs have full visibility into more than one layer.

Compared to inter-domain path computation, the response time to an inter-layer path request is usually smaller because the number of all possible PCEs and layers is usually limited. Therefore, there is no intrinsic need for the pre-reservation of resources at path computation time. Like the inter-domain case, there is no risk of a chain reaction of tear down messages due to the blockage at a given domain in the inter-domain path. However, it is still possible to apply the pre-reservation mechanism to the inter-layer solution, if an operator decides it is necessary.

Another important point, as mentioned in section 2.2, in that the existing PCE base inter-layer standard considers lower layers only when bandwidth becomes unavailable at the current layer, i.e., the current layer PCC receives a *PCRep* message carrying the *NOPATH* Object. The solution here allows to consider all layers simultaneously for each request requiring a globally optimal path. That is a layer can invoke lower layers even if it has enough resources to accommodate the request. This is why the path requests at different layers are said to be simultaneous. The OAM team can apply policies to, for example, have any request with a required bandwidth above one OC3 to trigger lower layers even if the bandwidth is available at the current layer. Another example would be to trigger lower layers when the path request demands the exclusion of certain routes, at the same layer or at lower layers. Moreover, if traffic forecasts are available, they could be used along with pertaining path requests to emphasize the gain of triggering lower layers for new capacity when answering a path request.

Besides, since an *optimal* end-to-end path is being computed, the notion of loose hops is not considered as it leads to uncertainty in the characteristics of the path. This is due to the fact that the section of the path referred to by the loose hop is calculated on the fly at path setup time.

Again, since the PCE architecture is considered, a connection refers to a deployed LSP. Figures 4.2 and 4.3 are used as example to show the possible inter-layer signalling implied with the proposed scheme. The proposed inter-layer scheme uses the same two basic messages: *[Path/QoS request]* and *[Path/QoS reply]*. The functionalities of these basic messages can easily be incorporated into the *PCReq* and *PCRep* messages of the PCEP protocol. The *[Path/QoS request-confirm]* and *[Path/QoS reply-confirm]* correspond respectively to RSVP-TE *Path* and *Resv* messages exchanged during the deployment of an LSP, or any other

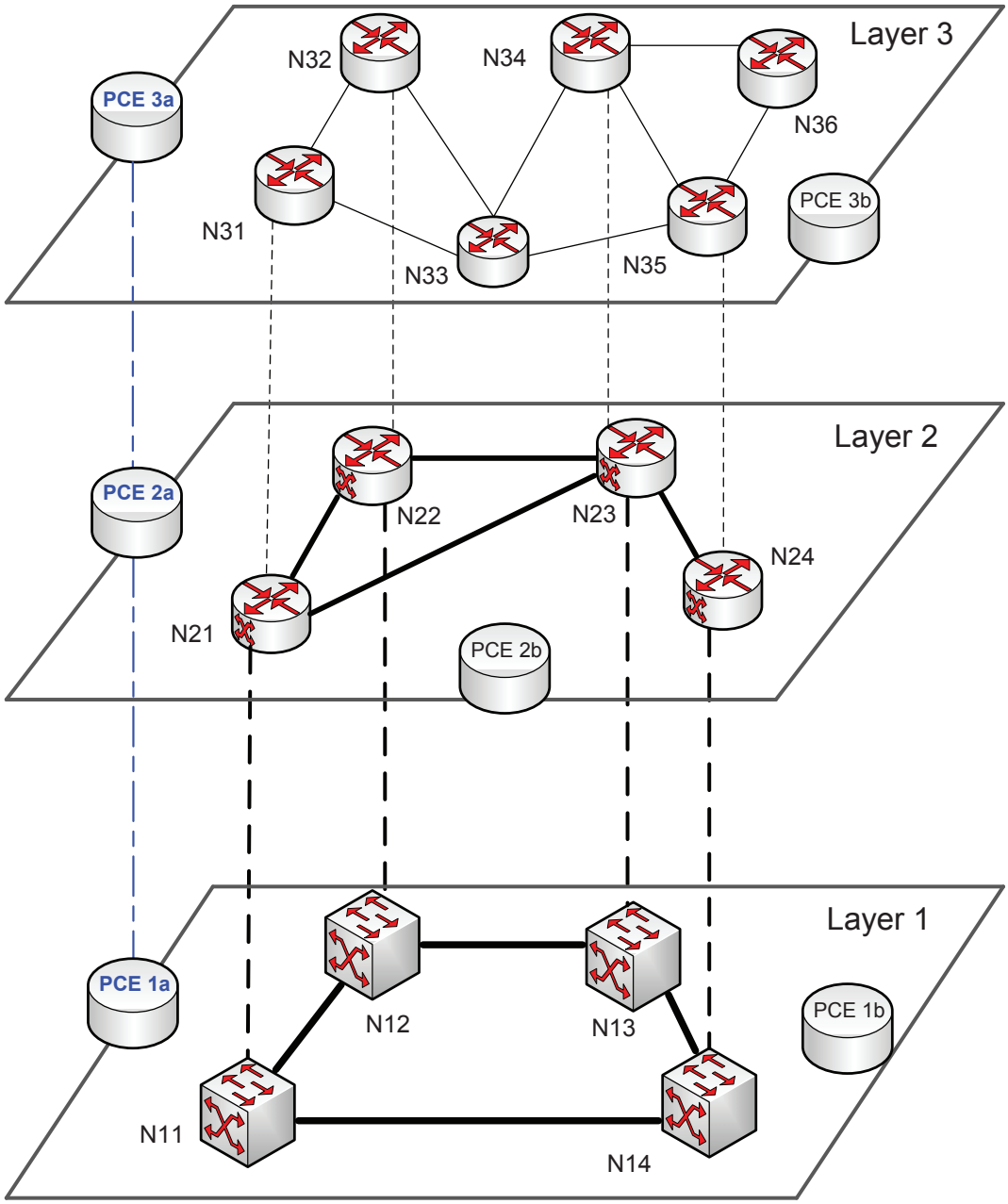


Figure 4.2 The network used as an example for the description of the proposed multi-layer path negotiation procedure.

deployment signalling if a technology other than GMPLS is used. The process to signal a higher-layer LSP that has an explicit route and includes hops traversed by LSPs in lower

layers is defined by Kompella et Rekhter (2005b) by the use of an interface identifier with the $IF\_ID\ RSVP\_HOP$ object that replaces the common $RSVP\_HOP$ object carried in the *Path* message. In the example of Figure 4.2, each layer has one inter-layer PCE (PCE_a)

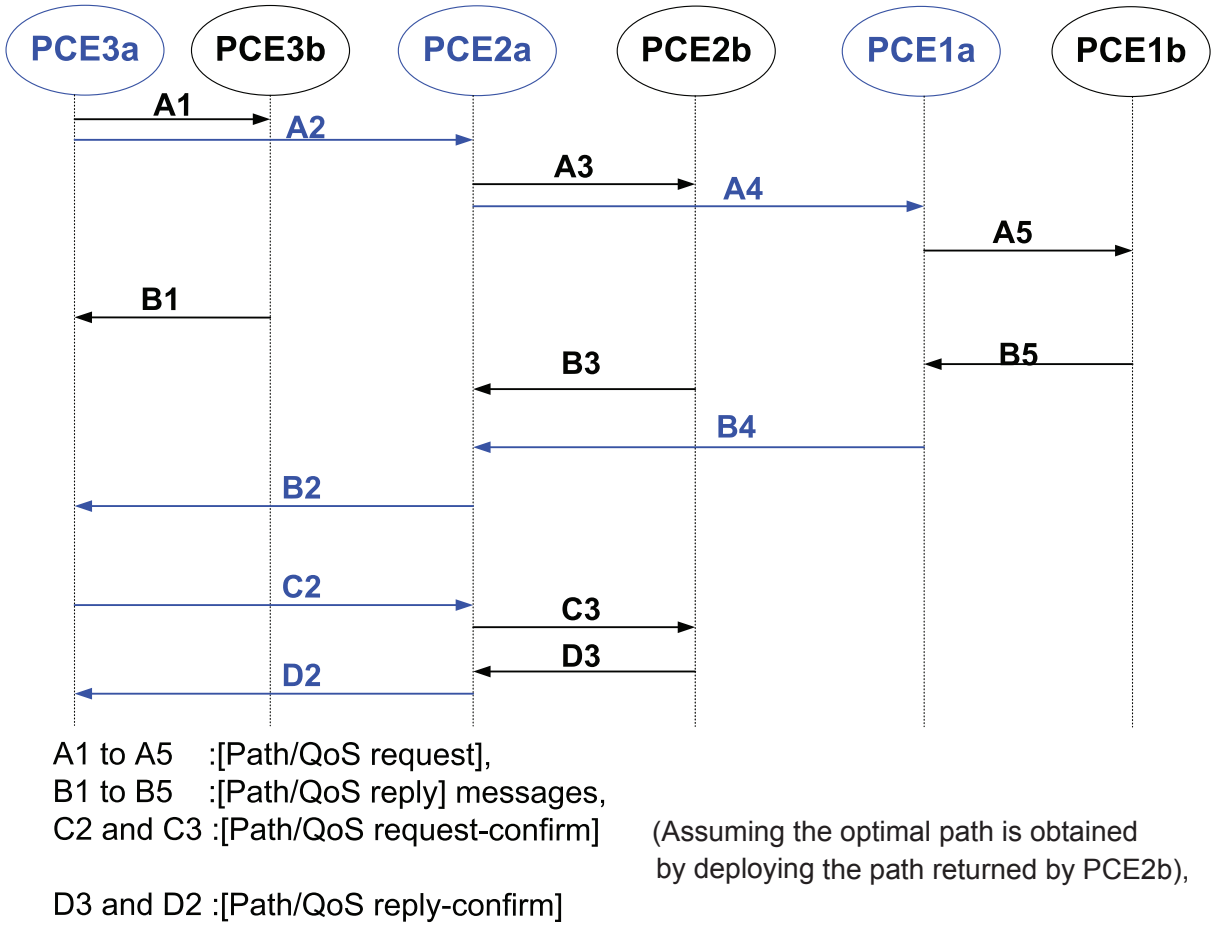


Figure 4.3 Signalling messages for the computation and establishment of an inter-layer path

and one intra-domain PCE (PCE_b). The intra-layer PCE connections between the PCEs or to other nodes are not shown for sake of clarity; only inter-layer PCE connections are shown. To simplify and clearly show the process, it is assumed that in each layer, the inter-layer PCE (PCE_a) receives the initial path request, then it forwards it to the intra-domain PCE (PCE_b). This is just a particularity of this example, otherwise PCE_a could have performed the functionalities of PCE_b.

Starting from layer 3, PCE_{3a} receives a path request (from a PCC that is not shown). The path request is, for example, for the least loaded path with the minimum end-to-end delay between the $Node_{N31}$ and $Node_{N35}$. PCE_{3a} first relays the request to PCE_{3b} and to PCE_{2a}. PCE_{3b} will use the information from the layer 3 TED and the objective function

carried in the request message to compute the best path and return it to PCE_{3a}. This is shown by messages $A1$ and $B1$ in Figure 4.3.

As mentioned, PCE_{3a}, forwarded the request to PCE_{2a} for a layer 2 connectivity between its $Node_{N31}$ and $Node_{N35}$. PCE_{3a} will use the reply from PCE_{2a} to compare the cost of the layer 3 path found by PCE_{3b} to the cost of the layer 3 path routed on a new connection as returned by PCE_{2a}. If the latter is the best choice, then the layer 2 path returned by PCE_{2a} will be provisioned (signalled) and then the layer 3 path will be routed on this new capacity [1].

PCE_{2a} has performed similar actions as PCE_{3a} and computed the possible paths before replying to PCE_{3a}. This is shown by request and reply messages $A2$ and $B2$. In the same manner, PCE_{1a} received a request from PCE_{2a} to find a path from layer 2 $Node_{N21}$ to $Node_{N24}$. The request and reply messages here are labeled $A4$ and $B4$. Through this recursive process, PCE_{3a} receives all possibilities with their associated costs. It can then chose among the available path possibilities, not only by considering the cost of the paths, but also Operation and Management (OAM) policies, traffic forecasts, upcoming downtimes, national security directives etc.

Finally, in this example, the best path as selected by PCE_{3a} is the one received from PCE_{2a} coming from PCE_{2b} (as opposed to the one coming from PCE_{1a}). PCE_{3a} could trigger the signalling for the provisioning of the path. Here the deployment signalling is shown between PCEs because it is assumed that the PCE nodes have path signalling capabilities, i.e., are G/MPLS capable. Therefore $C2$ and $C3$ represent RSVP-TE *Path* messages and $D3$ and $D2$ represent the *Resv* messages.

4.2.3 Instability risk of the inter-layer/inter-domain path negotiation procedure

Due to the nature of the technologies, lower layer connections are likely to have considerably larger capacities than the higher layer connections. This means that when a new $layer_n$ connection is setup to accommodate a $layer_{n+1}$ connection (demand), usually the bandwidth provisioned at $layer_n$ is a lot more than the require bandwidth of the demand at $layer_{n+1}$. The following is an example scenario that shows how this fact can cause instability in the network. It is assumed that a new $layer_{n+1}$ connection triggered the establishment of a new $layer_n$ connection. Depending on the traffic engineering practices in place, this new $layer_n$ connection could trigger a re-routing of existing connections at $layer_{n+1}$. Then the previous used $layer_n$ connection can become unused and torn down. Now if a new $layer_{n+1}$ comes

1. Today, this is often performed by human intervention to estimate the cost of using the available capacity versus requesting new lower layer connections. The optimality of such process depends on human factor such as experience. Needless to mention it is more time consuming, static, and error prone.

along and triggers the re-establishment of the layer$_n$ connection that was torn down, instability has occurred. Other scenarios could cause the tear down/re-establishment of lower layer connections in a similar cyclic manner. Stability is therefore an important issue for any inter-layer traffic engineering scheme. Thus, in practice the network operator should make use of thresholds in terms of deciding when to tear down a lower layer connection. In the same way, the right cost should be associated for the provisioning of lower layer connections, to discourage PCE path computation schemes to opt for lower layer connections for any small higher layer request. The proposed inter-layer path negotiation procedure faces the same instability risks if careful traffic engineering policies are not applied. However, by applying an extra cost or policies in the establishment of lower layer connections, the instability problem can be limited and avoid the cyclic tear down/re-establishment example described above.

4.2.4 Comparison between the inter-domain and the inter-layer path computation schemes

The recursive path computation scheme of chapter 3 has been adapted in order to provide a distributed inter-layer path computation scheme. As already mentioned, the technology layers replace the domains in the inter-domain scheme. One inherent difference is the content of the reply message to a path request.

Figure 4.4 simplifies the example of Figures 4.2 and 4.3 and shows the reply messages' path tree. The request is for a path from PCE$_{3a}$ to PCE$_{3b}$, assuming they are routers with PCE capabilities, and abstracting away intermediate nodes. Here it is clear that three different paths are possible. One path is obtained from the reply of PCE$_{3b}$. Another one is obtained from the reply of PCE$_{2a}$ coming from PCE$_{2b}$. And the last one is obtained from the reply of PCE$_{2a}$ coming from PCE$_{1a}$.

Figure 4.5 converts the same example to its inter-domain counterpart by replacing each layer by a domain and keeping the same PCEs' connectivity. Obviously, the path request will be translated into in inter-domain path computation from PCE$_{3a}$ to PCE$_{1b}$. It is interesting to note that the reply messages' path tree here has only one PCE/tail, whereas the inter-layer case has three different PCE/tails.

It is important to mention that the inter-domain scenario and its inter-layer counterpart are not physically comparable. The comparison here shows that the same inter-domain process applied by replacing domain boundary definitions with layer boundary definitions, yields different results in terms of Virtual Shortest Path Tree (VSPT).

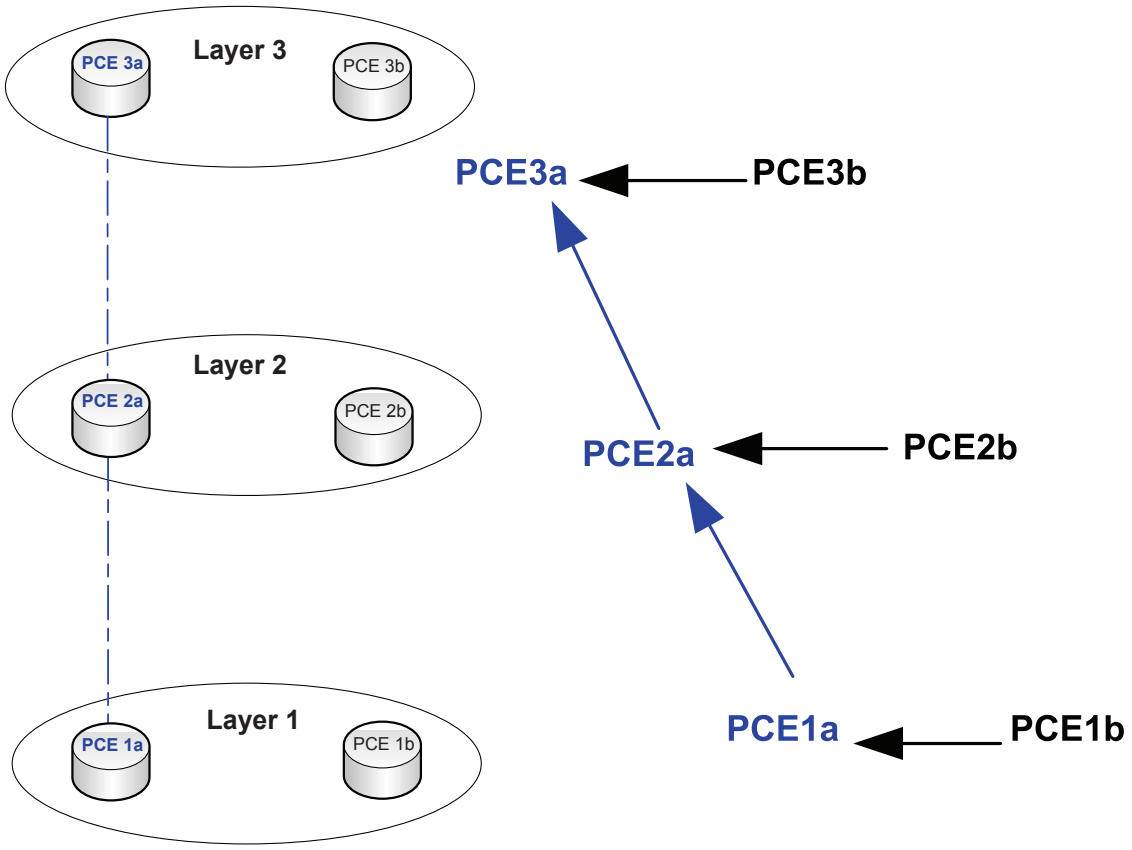


Figure 4.4 Return message tree for inter-layer path computation

4.2.5 Risk of PCEP message loops

Section 3.2.3 in the previous chapter proposed a simple loop prevention mechanism applicable to the inter-domain scheme. Even though the same mechanism is applicable here, there is a less apparent risk of PCEP message looping here because the layer border PCEs are well identified and should only trigger layer border PCEs of a lower layer. Therefore the risk of a *PCReq* message being sent back from a lower layer to the layer above is negligible.

4.3 End-to-End Multi-Layer/Multi-Domain Traffic Engineering Scheme

The end-to-end traffic engineering scheme consist in applying the multi-layer/multi-domain path computation scheme to each incoming request. Moreover, lower layer PCEs are triggered for each request, using a bundled demand composed of the current request and predicted ones. The prediction algorithm is assumed to be known. Figure 4.6 gives a flowchart view of the

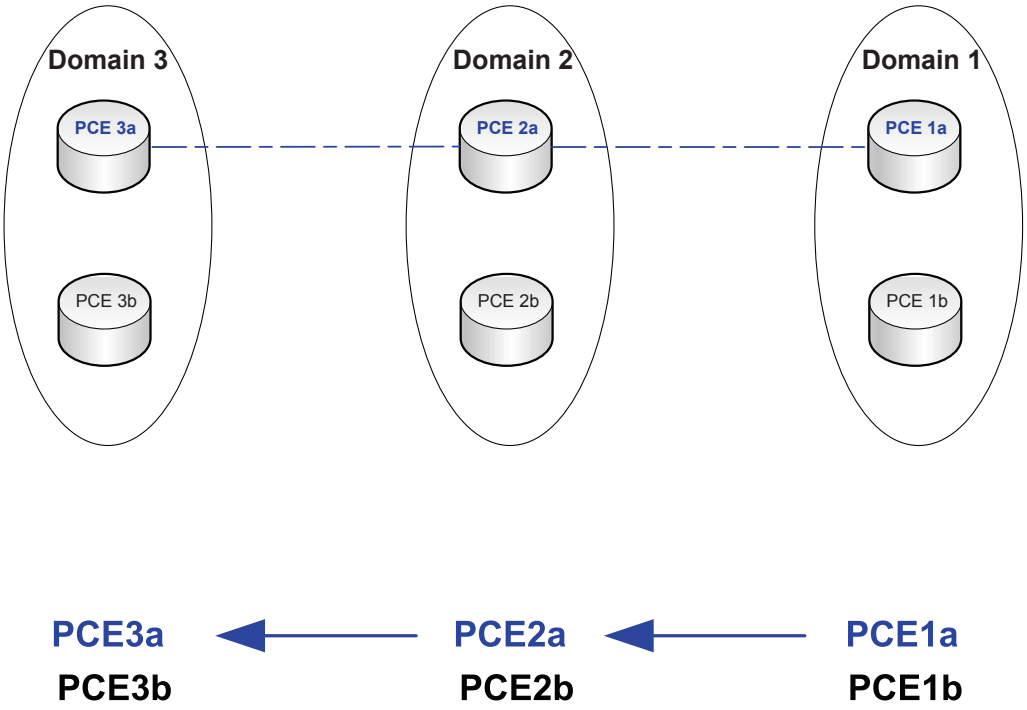


Figure 4.5 Return message tree for inter-domain path computation

traffic engineering procedure as implemented in a PCE node. The client's PCC sends a path request at layer$_N$ to the PCE of this layer. The PCE will more probably cooperate with other PCEs in the same layer to compute the path. However, it will also consider a bundle request formed from prediction results and the current request. The predicted requests that are for the same *source/ destination* or that may share common links with the current request are used to estimate the future required bandwidth. Other criteria can be used to estimate the bandwidth of the bundle (e.g. the desired link utilization). Thus, the PCE at layer$_N$ sends the bundle request to layer$_{N-1}$. Once the PCE at layer$_N$ receives all the replies, it can choose the one that is more suitable, i.e., the one returned by the PCEs at layer$_N$ or the one returned from layer$_{N-1}$. The selection criteria may be influenced by various factors, such as the priority associated with the predicted requests used in the bundle, the setup time requirement of the current request, policy enforcements, etc. Once the selection is made, the requested path can be deployed.

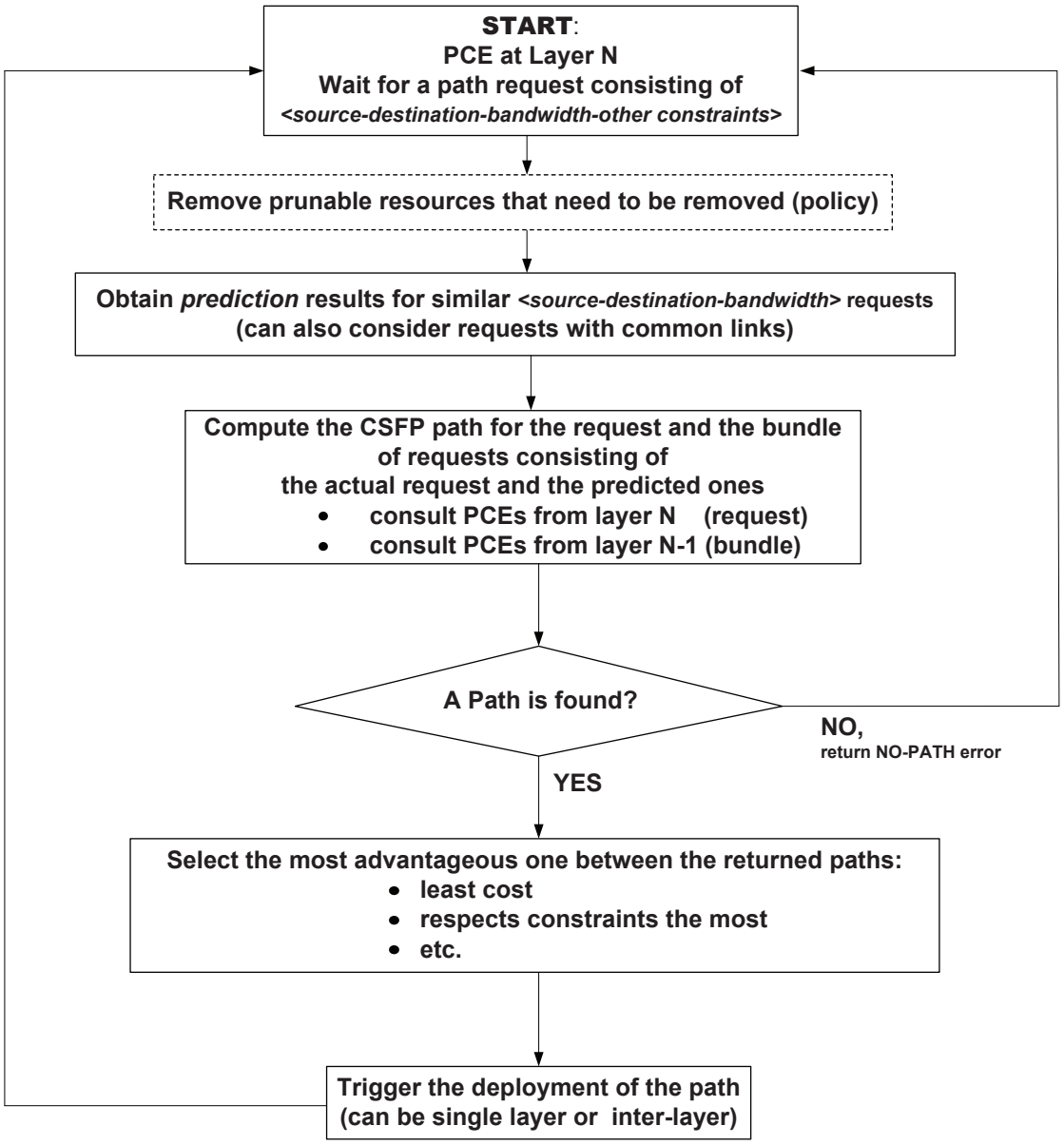


Figure 4.6 Proposed traffic engineering algorithm

4.4 Validation of the End-to-End Traffic Engineering Scheme

First, the proposed inter-layer/inter-domain path computation scheme is analyzed both qualitatively and quantitatively. The qualitative analysis in section 4.4.1 compares the proposed scheme to the centralized approach, and to the current approaches of inter-layer/inter-domain traffic engineering path computation. The comparisons are done for different properties and features of the mentioned schemes. Then, the quantitative analysis of section 4.4.2

presents a mathematical analysis which brings to light the key timing values affecting the setup time of an multi-layer path (LSP deployment time). Then, the simulations are conducted to evaluate the overall traffic engineering scheme. Section 4.4.3 outlines the details and parameters used in the simulations, and section 4.4.4 presents the results.

4.4.1 Qualitative analysis of the proposed scheme

Table 4.1 compares the proposed distributed approach to the well investigated centralized approach. The important aspects that stand out are the scalability and inter-layer/inter-domain compatibility of the distributed approach. The only inconvenient is the inter-PCE communication overhead as well as the optimality of the end-to-end path. To achieve global end-to-end optimality, the inter-layer distributed scheme has to be well designed. The scheme presented in this chapter allows the discovery of the optimal end-to-end path, the same way as the inter-domain scheme of chapter 3.

Table 4.1 Properties of centralized versus distributed multi-layer path computation approaches

Multi-layer PCE approach	Centralized	Distributed
TED	Single	Multiple (usually one per layer)
Scalability	No	Yes
Assured connectivity withing PCE's reach	Yes	Yes
Assured connectivity if destination is under another administration	No	Yes
Inter-layer path computation among different domains	No	Yes
Confidentiality respected	Yes	Yes
Inter-PCE communication	No	Yes
Optimal path	Yes	Yes

The proposed scheme requires the triggering of the lower layers even if resources are available at the higher layer. The proposed approach is to do so whenever the requested bandwidth exceeds a certain limit or has a disjoint path requirement. This is different from current approaches where the lower layers are triggered only when the higher layer cannot accommodate the request (e.g. not enough bandwidth).

Table 4.2 compares the two approaches for different criteria. The important aspect that stands out in this comparison is that the proposed scheme may trigger too much signalling

which could hinder scalability. But this can be overcome by using the distributed concept again and allocating multiple PCEs per layer. Considering all layers is necessary if a globally optimal path is to be computed. But this implies that the proposed scheme has a higher risk of instability because there is a chance of creating new lower layer connections at a higher rate. This may disrupt the higher layer path computation processes, as discussed in section 4.2.3. Moreover, for cases where a lower layer path is required, the proposed scheme has a lower path computation time because each layer performs the path computation in a disjoint and concurrent manner. The usual schemes perform the same computations sequentially, that is when the higher layer fails to find a path, only then they trigger the lower layer. The effect of such practices on the actual path (LSP) setup time is studied in section 4.4.4.

Table 4.2 Features of the proposed distributed multi-layer scheme

Multi-layer distributed approach	Proposed scheme	Usual schemes
TED	Multiple (usually one per layer)	Multiple (usually one per layer)
Inter-layer PCE communication	Yes	Yes
Triggering of lower layers	Always or when BW request exceeds a certain value	Only when BW is not available at current layer
Scalability	More at risk	Less at risk
Concurrent path computation	Yes	No
Path global optimality	Yes	No
Risk of instability	Higher	Lower

4.4.2 Analytical analysis of the proposed scheme

It is considered that a global end-to-end path request scenario consists of N layers, with the demand originating at layer N. It is assumed that the PCEs are mono-layer. In each $layer_n$ there are M^n PCE nodes responsible for covering the whole layer. The processing time of the PCEP message on PCE_j^n is represented by $t_j^{proc,n}$, for $j = 1$ to M^n and $n = 1$ to N. Then it is assumed PCE_j^n takes $T_j^{comp,n}$ time to compute its part of the end-to-end path. $T_j^{comp,n}$ depends on the workload of PCE_j^n, its CPU power, its memory, its implementation efficiency, the compiler's efficiency, the objective function and algorithms chosen, as well as the complexity of the network. Moreover, the PCEP communication from PCE_{j-1}^n to PCE_j^n takes $T_{j-1,j}^{TCP,n}$ time. It is also assumed that in each $layer_n$ only PCE_1^n can communicate to the

layer border PCEs of the adjacent layers $n-1$ and $n+1$. The TCP delays between PCE_1s in adjacent layers are denoted as $T^{TCP}_{1^{n-1},1^n}$ and $T^{TCP}_{1^n,1^{n+1}}$. For each layer, the time T^n elapsed from when PCE^n_1 sends a request *[Path/QoS request]* and when it receives a *[Path/QoS reply]* from an adjacent PCE in its own layer ($layer_n$), can be expressed as equation 4.1.

$$T^n = 2\sum_{j=1}^{M^n-1} t^{proc,n}_j + t^{proc,n}_{M^n} + \sum_{j=2}^{M^n} T^{TCP,n}_{j-1,j} + \sum_{j=2}^{M^n} T^{TCP,n}_{j,j-1} + \sum_{j=1}^{M^n} T^{comp,n}_j \tag{4.1}$$

This time represents only the per layer part of the overall path computation delay. For $layer_1$, the path computation response time $T^1_{response}$ is T^1. In a multi-layer case when N layers are interrogated, a certain timing overlap is to be considered between adjacent layers. Since it is assumed that PCE_1 in each layer is the only multi-layer PCE capable of communicating with adjacent layers, then for $layer_2$ the path computation time $T^2_{response}$ is

$$\max(T^1_{response} + T^{TCP}_{1^{layer_2},1^{layer_1}} + T^{TCP}_{1^{layer_1},1^{layer_2}}\ ,\ T^2)\ ;$$

for $layer_3$ the path computation time $T^3_{response}$ is

$$\max(T^2_{response} + T^{TCP}_{1^{layer_3},1^{layer_2}} + T^{TCP}_{1^{layer_2},1^{layer_3}}\ ,\ T^3)\ ;$$

for $layer_n$ the path computation time $T^n_{response}$ is

$$\max(T^{n-1}_{response} + T^{TCP}_{1^{layer_n},1^{layer_{n-1}}} + T^{TCP}_{1^{layer_{n-1}},1^{layer_n}}\ ,\ T^n)\ ;$$

and finally for the highest $layer_N$ the path computation time $T^N_{response}$ is

$$\max(T^{N-1}_{response} + T^{TCP}_{1^{layer_N},1^{layer_{N-1}}} + T^{TCP}_{1^{layer_{N-1}},1^{layer_N}}\ ,\ T^N)\ .$$

The response time $T^N_{response}$ at $layer_N$ is the actual response time of the complete inter-layer path computation procedure. Upon reception of the response from PCE^{N-1}_1, PCE^N_1 decides by comparing the different path costs if it is better to chose the path found in its own layer or if it is more advantageous to trigger the path returned by PCE^{N-1}_1. Although it is possible, at this point there is not a need to recompute a path at $layer_N$ by considering the path returned from $layer_{N-1}$ because the information of the path is already in the reply from PCE^{N-1}_1. The same reasoning applies to others layers and their lower layer neighbours.

From when a PCC sends a request and when a path (LSP) is established for the traffic, $T^N_{response}$ time plus a certain setup delay T_{setup} has elapsed. At each $layer_n$ the setup delay

T^n_{setup} depends on the transmission delay of the link and the length of the setup message, the propagation delay (between a few to a few tens of ms), the queuing delay and the processing delay of the signalling message at each node (RSVP-TE processing delay is between 1 ms to 10 ms per node). The setup delay also includes the configuration of the LFIB tables and switching matrices in each node (this can take from half to one and a half second for optical cross connects) . The overall T_{setup} can be obtained by

$$T_{setup} = \sum_{n=l}^{N} T^n_{setup} \ ,$$

given that the optimal path is obtained by requesting new connections at $layer_l$ up to $layer_N$. It is clear that the setup delay may be longer if new capacity has to be deployed at lower layers, with the worse case of $l = 1$. Therefore, if the *[Path/QoS request]* has a short response time requirement (example on demand recovery path), then it is better to consider lower layers only if the upper layer does not have enough bandwidth. Depending on the path computation algorithm used, this can be incorporated as a constraint.

4.4.3 Simulation settings

A simulator is developed for the performance evaluation of the proposed multi-layer/multi-domain traffic engineering and path computation scheme. The performance of the proposed scheme is evaluated by first comparing the benefits of dynamic multi-layer path computation compared to current practices. Then, the simulation results compare the proposed scheme to existing ones that trigger lower layers only when the upper layer does not have resources. It also evaluates the benefits of using demand predictions when performing traffic engineering. The simulator is written in MATLAB using the 64 bit version 7.8.0.347 (R2009a). The choice of the language and simulation environment is in part due to the ability to easily implement and debug graph algorithms and matrix manipulations. The simulations are run on an Intel Core2 Duo CPU P8400 (at 2.26GHz) machine with 4GB of RAM, running the 64 bit Windows 7 Professional operating system.

Four performance parameters are used for the evaluations. First, the percentage of blocked path requests is measured. This is the ratio of successfully routed requests on the number of requests made during the simulation time. Second, the average and mean path length in number of hops is measured to compare the quality of the returned paths. It should be recalled that the number of hops is often a routing criterion in *all* optical networks where a light path can only take a maximum number of hops to avoid power loss and signal degradation (Leblanc *et al.*, 1999). Third, the average path setup time value is measured to

compare the setup delay of the requests (i.e., LSP deployment time). Section 4.4.2 presented the factors affecting the path setup delay. The fourth performance parameter is the average link utilization throughout the simulations. This gives an insight on the overall quality of the resource management and traffic engineering scheme.

The results are obtained for various test scenarios, topologies and demand matrices, as described below. The five test plan scenarios for the evaluation of the proposed end-to-end traffic engineering scheme are subdivided under three main class of inter-layer traffic engineering approaches. Scenario I is the way existing solutions perform inter-layer traffic engineering. Scenarios II-A and II-B are close to the proposed scheme and make lower layer path requests in bundles consisting of a pre-determined minimum bandwidth. Scenarios III-A and III-B use traffic predictions to determine the bandwidth of these bundles. The five scenarios along with the test networks and demand traffics are further described below. Table 4.3 summarizes the characteristics of the simulated scenarios. Moreover, each established path (LSP) has an infinite life, that is not torn down throughout the simulation.

Scenario I

Scenario I considers only the higher layer. When bandwidth is not available for a given path request, it is considered as blocked. This is different from the other scenarios in which the lower layer network can be triggered. The reasoning is that when dynamic lower layer provisioning is not available, as it is the case of most carriers today, the management team from a higher layer has to consult the management team of the lower layer to order the missing capacity. This usually takes a few days to months, thus the path request can be considered as blocked. Dijkstra's algorithm (Dijkstra, 1959) is used to compute the shortest path for each path request. Because the QoS criterion is the demand's bandwidth, the bandwidth matrix is pruned before calling Dijkstra, i.e., all links with an available bandwidth less than the demand's bandwidth are removed from the network graph. The path setup delay (LSP deployment) is affected by the RSVP-TE processing delay which varies from 1 ms to 10 ms, the propagation delay on the link which varies from 4 ms to 57 ms with an average of 26 ms. Obviously, the length of the shortest path has an effect on the total setup delay.

Scenario II

Scenario II represents a transition between existing proposals and the way of doing presented in this chapter on multi-layer traffic engineering. Here, two layers are considered. The lower layer is triggered only when the higher layer has no more bandwidth available on one

or more links of the shortest path. The lower layer network can be triggered to provision for up to once the original bandwidth of each link in the higher layer. The path request is considered first by running Dijkstra on the pruned graph to satisfy the bandwidth requirement of the request. If no path could be found, then the lower layer is triggered with K-shortest path algorithm [2]. In the simulations, K is set to 5. Then the first shortest path among the five is selected based on two criteria: 1-it must have enough bandwidth on all links, or 2- if criteria 1 is not met, the links that do not have enough bandwidth should not have reached the maximum number of lower layer upgrades.

Again, the path setup delay (LSP deployment) is affected by the RSVP-TE processing delay which varies from 1 ms to 10 ms, the propagation delay on the link which varies from 4 ms to 57 ms with an average of 26 ms. Here, the length and depth of the shortest path has an effect on the total path setup delay. The configuration of the optical switches at the lower layer vary from 500 ms to 1500 ms [3]. The minimum lower layer bandwidth request is the equivalent of a T1 (1.5 Mbit/s) in scenario II-A, and it is equivalent to a DS3 (50 Mbit/s) in scenario II-B, which brings it closer to the proposed method and scenarios III-A and III-B.

Scenario III

This scenario may trigger the lower layer even when bandwidth is available at the higher layer, for every demand. For the case where bandwidth is not available on the higher layer, this scenario performs exactly like Scenario II. Otherwise, for each demand, the scheme considers a prediction of upcoming path requests. The prediction window is set to 100 time slots and its accuracy is varied in Scenario III-A and Scenario III-B. The prediction of upcoming requests can be a function of marketing and sales forecasts as well as a prediction mechanism using Kalman filters or Neural Networks. The actual path request prediction mechanism is outside the scope of this work.

Topologies

The performance of the proposed scheme is evaluated by comparing five different scenarios drawn from three approaches. Each of these scenarios is tested on two different networks. The network of Figure 4.7 represents the National Science Foundation Network. The network of Figure 4.8 is a fictive carrier backbone. Table 4.4 presents the networks' characteristics.

2. The K-shortest path algorithm as defined by Yen (1971) is presented in more detail in section 2.3.

3. The various timing delay values are measured by Song *et al.* (2005).Their work shows that RSVP-TE signalling transmission delays are negligible; but, its processing delay cannot be ignored. Moreover, OXC cross-connection delays are the most important factors in setup delays and should be minimized.

Table 4.3 Simulation Scenarios

Summary description of each scenario	
Existing solutions:	
Scenario I-A:	Single layer scenario. If bandwidth is not available for a path request, it will be considered as blocked.
Minimum lower layer request:	
Scenario II-A:	The lower layer is dynamically triggered when, for a given request, bandwidth is not available on one or more of the higher layer links on the TE-LSP. Minimum lower layer bandwidth request is the equivalent of a T1.
Scenario II-B:	Same as II-A but minimum lower layer bandwidth request is the equivalent of a DS3.
Prediction based lower layer request:	
Scenario III-A:	For all requests, the lower layer is dynamically triggered when bandwidth is not available on one or more of the higher layer links on the TE-LSP when considering traffic predictions with 100% accuracy.
Scenario III-B:	Same as III-A but considering traffic predictions with 50% accuracy.

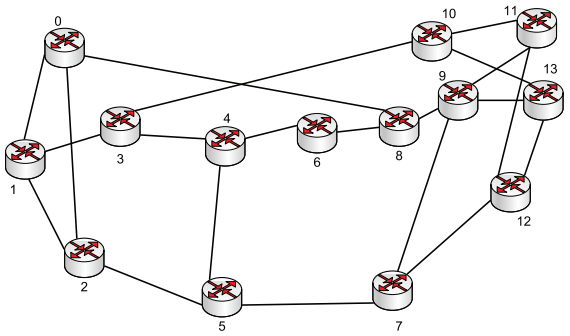
The nodes at the higher layer represent typical GMPLS routers. The nodes at the lower layer represent optical switches controlled by GMPLS. Both networks are dimensioned with OC48/STM16 links (approximately 2400 Mbit/s) at the higher layer. For scenarios II and above, the lower layer network can be triggered to provision for up to once the original bandwidth of each link in the higher layer.

Table 4.4 Characteristics of the simulated networks

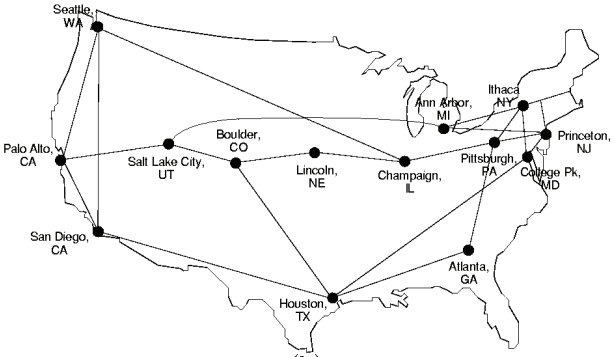
Network	Number of Nodes	Number of bi-directional links	Average nodal degree
NSFNET	14	21	3.0000
Carrier Backbone	24	43	3.5833

Demand matrices

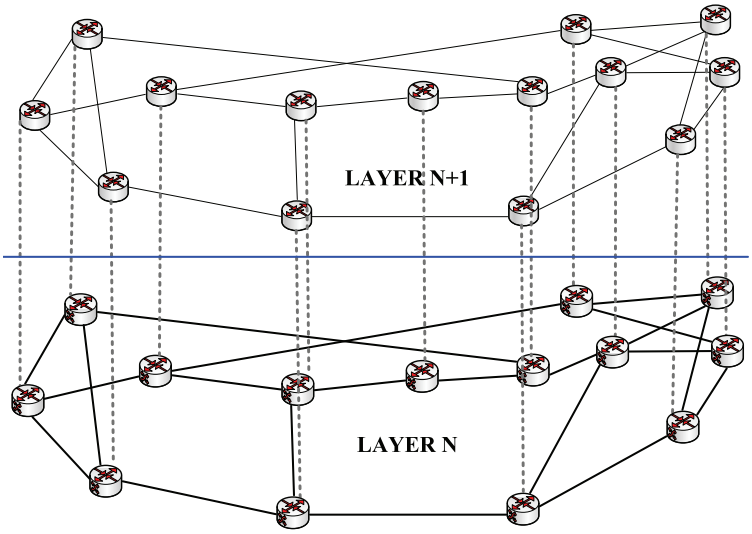
For each of the two network topologies, a demand matrix is randomly created and used for all scenarios. A demand is a *source/destination* in the higher layer, with a bandwidth requirement. The demand matrix contains 1500 time slots. Each demand set can be consid-



(a)



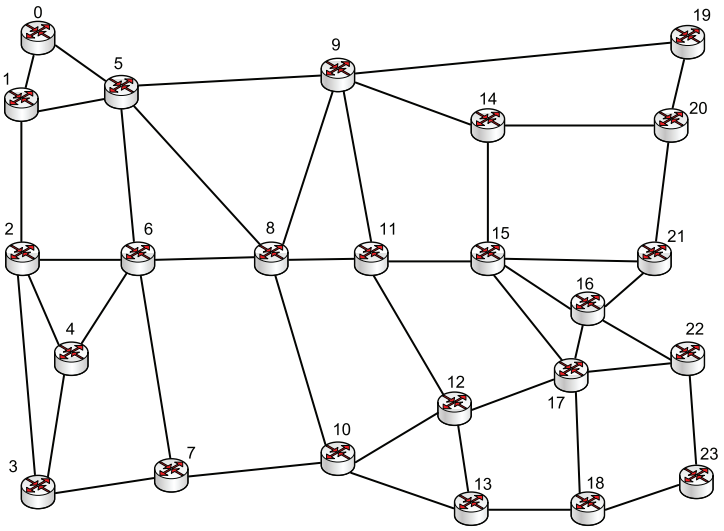
(b)



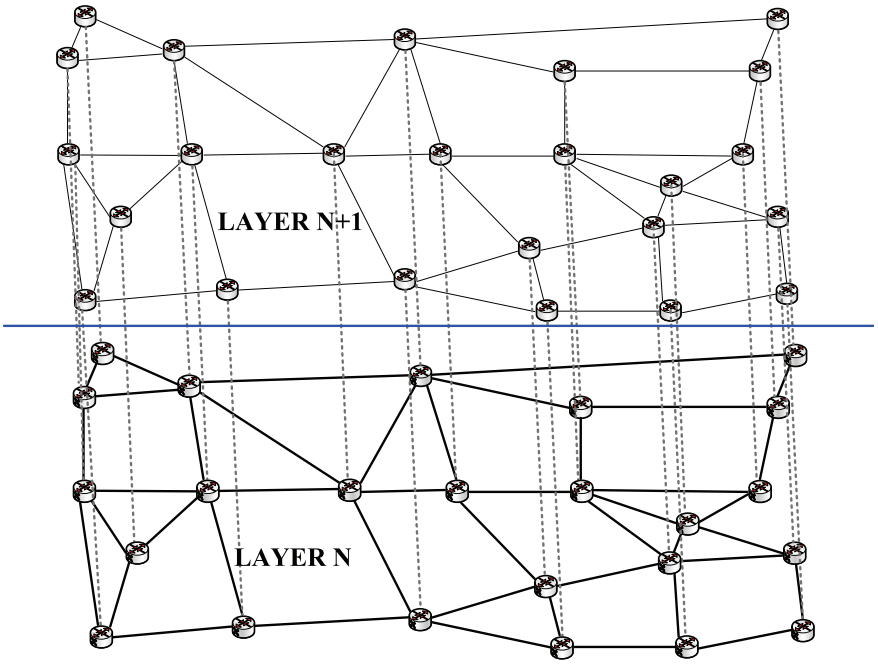
(c)

Figure 4.7 The National Science Foundation Network (NSFNET)

ered as a daily or hourly set of path requests. In each time slot, a complete *Number of Nodes* $\times$ *Number of Nodes* matrix is created containing the path requests from each source (line) to



(a)



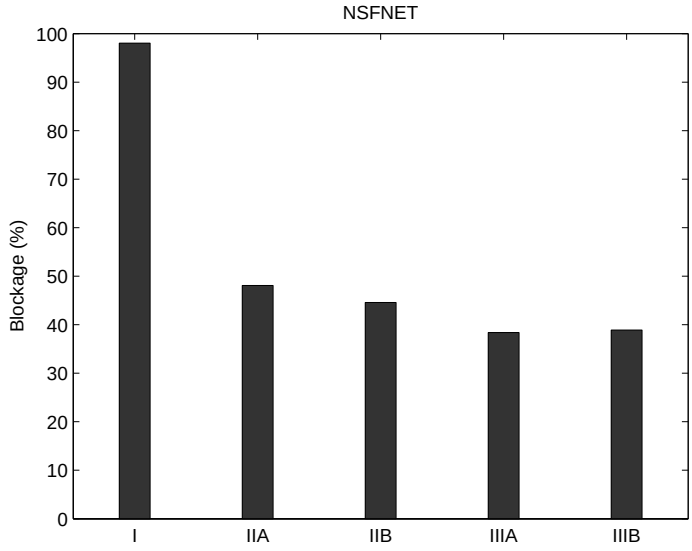
(b)

Figure 4.8 Carrier Backbone network

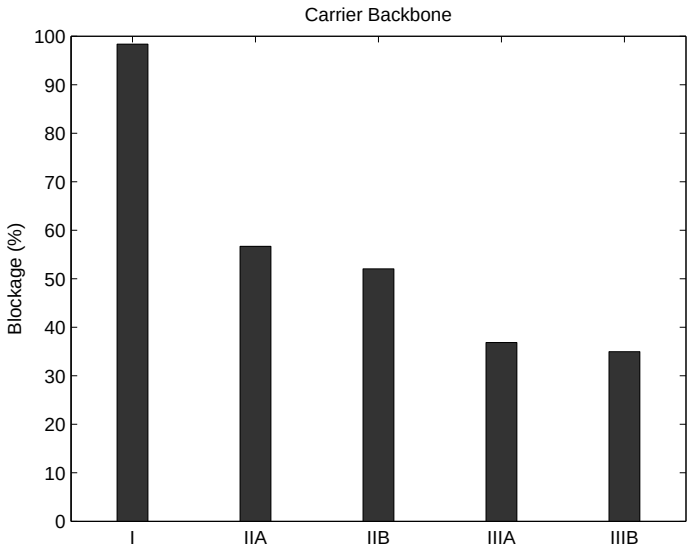
each destination (column) in the matrix. The NSFNET demand matrix contains a total of 75865 path requests. The Carrier Backbone demand matrix contains 229495 path requests. The sum of all demands in Mbit/s is 136761 for the NSFNET demand matrix and 1208348 for the Carrier Backbone demand matrix.

4.4.4 Simulation results

The simulation results, obtained from representative average results of a set of extensive experimentations, are presented in Figures 4.9 to 4.13. The five test cases from three main scenarios are annotated as Scenarios I, II-A, II-B, III-A and III-B, as described in Table 4.3 above.

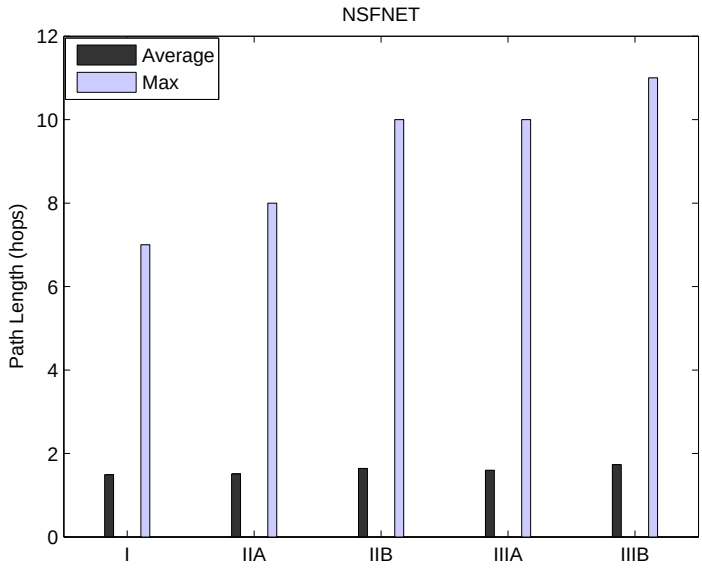


(a)

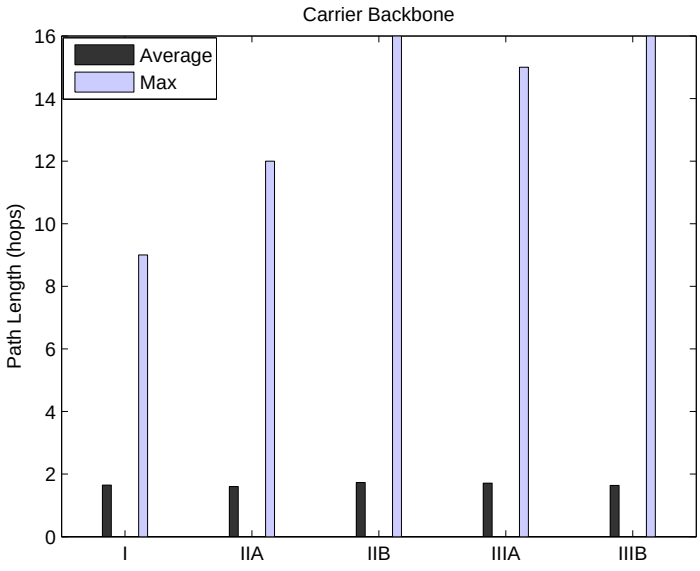


(b)

Figure 4.9 Percentage of blocked path requests



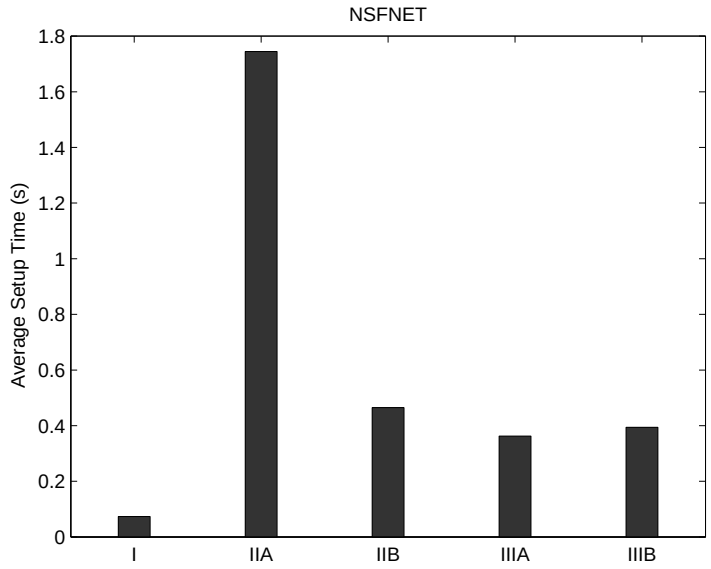
(a)



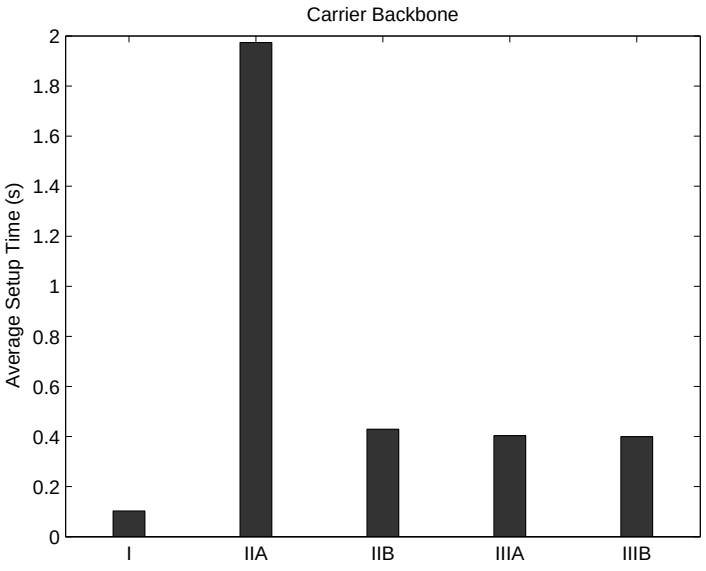
(b)

Figure 4.10 Path length in number of hops

Figure 4.9 presents the percentage of blocked path requests for each network. As expected scenario I has the worse performance because it does not dynamically trigger the lower layer. Scenario III-A where accurate predictions are available, has the best performance, but the difference is small enough to make it comparable to scenarios II-B and III-B. This makes questionable the benefits of using very precise predictions versus less accurate ones or even



(a)

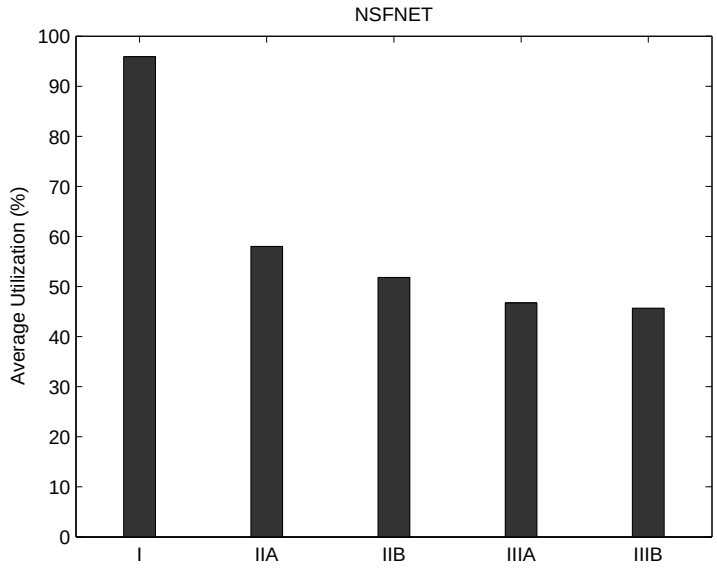


(b)

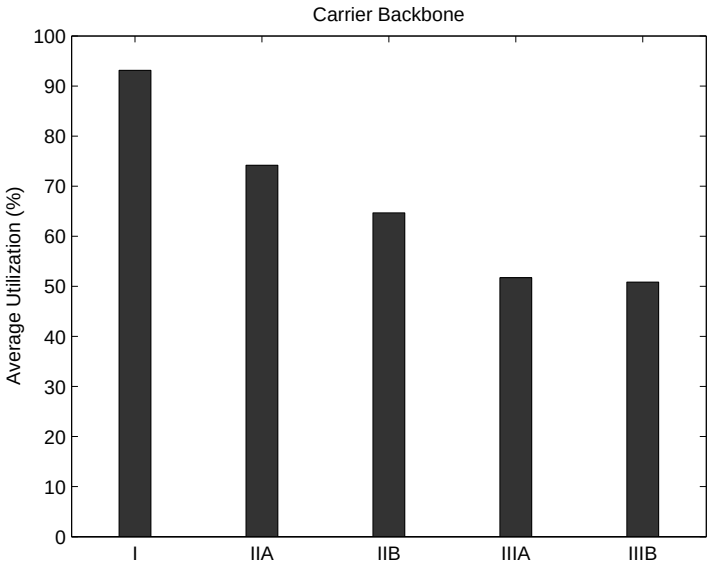
Figure 4.11 Path setup time in seconds

versus using, like in scenario II-B, a fixed value minimum lower layer demand.

Figure 4.10 presents the cost of the path in terms of number of hops or path length for both networks. As expected, scenarios II and III have a slightly higher average path lengths than scenario I because they accommodate more requests by triggering the lower layer in a dynamic manner and using the K-shortest path algorithm. This is why they find longer



(a)



(b)

Figure 4.12 Average link utilization

paths but offer a better throughput (reduced blockage).

Figure 4.11 presents the average path setup delay. The factor influencing the most the path setup delay is when the lower layer has to be triggered. The delay is added by the configuration of the optical switches at the lower layer. Thus, it is better to trigger the lower layer for bundles of upper layer requests, as proposed by the traffic engineering scheme. That

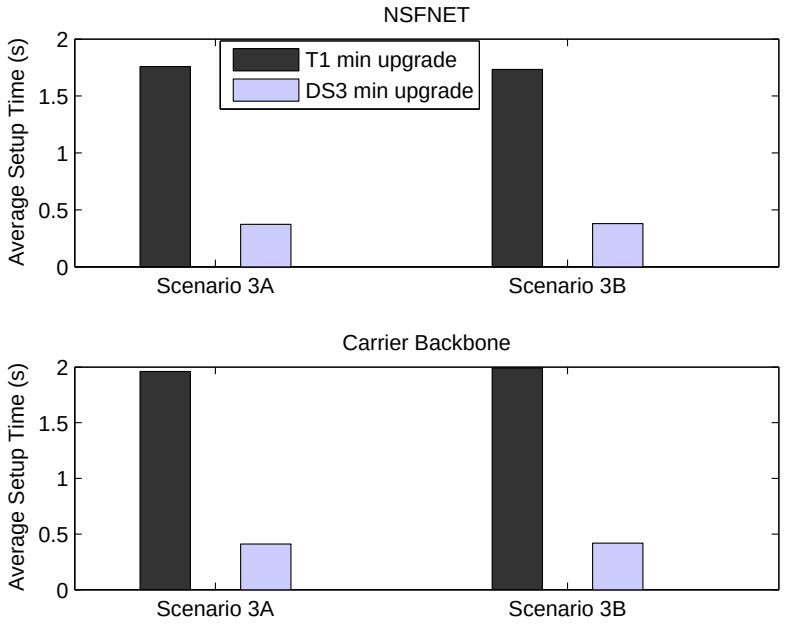


Figure 4.13 Effect of the minimum bandwidth that can be requested from a lower layer on the setup time

is exactly the effect of scenario III which uses predictions. Moreover, the same effect is seen in scenario II-B which is the case where the minimum bandwidth upgrade is large enough and can be considered as some sort of average like prediction. Again, this is an extremely interesting result that will be discussed below when addressing the benefits and guideline for using traffic predictions in traffic engineering.

Figure 4.12 presents the average utilization values for both networks. These results follow the overall trend that scenario III-A has the best performance but which is comparable to scenarios II-A, II-B, and III-B. Scenario I suffers from higher link utilization, due to its inability to trigger the lower layer for an on demand upgrade in bandwidth. Scenarios II-A to III-B have relatively similar overall utilization.

Figure 4.13 compares once again the setup time results of scenarios III-A and III-B, this time with a prediction window of 250. It is seen that even for larger prediction window, the impact of prediction accuracy is minimal on the setup time.

Therefore, the question as to weather invest or not in accurate traffic prediction mechanisms is answered by these results. It is clear that the benefits of considering upcoming demands whenever making a demand to a lower layer, that perhaps belongs to a different administration, is crucial. However the exactitude of demand predictions does not play a role

in the end results of blockage/throughput, setup delay, average path length and utilization.

4.5 Summary

This chapter presents an overall process of end-to-end traffic engineering where multi-domain and multi-layer path computation could occur concurrently. A PCE based multi-layer path negotiation scheme is proposed, which, given its distributed nature, can be applied to cases where the lower layer belongs to a different management group or organization. The proposed ideas are discussed and analyzed both qualitatively and quantitatively. The path setup time is mathematically formulated considering various factors that affect its value.

Relying on the proposed scheme, a simulator is developed and used to analyze, first the effect of performing inter-layer traffic engineering, and then the effect of using traffic demand predictions for more efficient path computation. Analysis and simulation results support the argument that constant multi-layer traffic engineering is essential for better resource allocation and for a faster path setup time. Then, the use of accurate predictions versus less accurate ones is studied. The results prove that in fact the same benefits are obtained with 100% accurate predictions, with 50% accurate predictions or just with a larger average like value.

CHAPTER 5

MULTI-LAYER PATH COMPUTATION ALGORITHM WITH ADAPTATION CONSTRAINTS

The two previous chapters tackled the general inter-domain and the collaborative multi-domain/multi-layer path computation schemes. The proposed solutions in these chapters consist in distributed PCE based schemes with traffic engineering guidelines in multi-domain/multi-layer scenarios. Each PCE is responsible for a complete or part of a network. The work in these chapters assume that the PCEs have already implemented the right set of algorithms for constraint shortest path firth (CSPF) path computation. As presented in section 2.3, CSPF routing has been subject of great interest in the scientific community. However, when is comes to the multi-layer/multi-region GMPLS CSPF problem, the existing works tends to over simplify the constraints or to completely neglect certain technological aspects.

To this end, this chapter tackles the problem of defining one such CSPF path computation algorithm for multi-layer/multi-region GMPLS networks. The proposed algorithm has its novelty in the fact that it addresses the problem in its entire form considering technological and traffic engineering constraints. The specific problem consists of finding a path that respects the switching capability constraints of GMPLS nodes, as described in section 5.1.2 below. The literature review of section 2.3 presented related works and discussed how they fail to address the actual problem. Thus, the proposed solution addresses those shortcomings. The algorithm presented in this chapter can be implemented in any PCE that may be solicited to compute end-to-end multi-layer/multi-region paths that respect switching capability constraints and traffic engineering rules.

Section 2.3 discussed previous works that have proved that in a multi-layer network, the process of finding a minimum cost path that crosses different layers is NP-Hard. This implies that the optimization of inter-layer routing can be solved in an exact way by mathematical programming for static traffic data in small networks. However, for on demand dynamic routing of multi-layer LSPs in real world larger networks, a heuristic method is more suitable. To this end, this chapter proposes a novel algorithm for the general multi-layer/multi-region CSPF LSP routing problem. It involves in one part the computation of the shortest paths, in another part the solving of a binary integer program (BIP) which integrates the multi-layer/multi-region constraints of GMPLS networks. The solution is compared to the work presented in Jabbari *et al.* (2007) as well as Gong et Jabbari (2008) , already introduced in

section 2.3. This comparison is done even though the proposed algorithm already outperforms these works just by the fact that it considers crucial constraints neglected by the authors.

The rest of this chapter is organized as follows. Section 5.1 classifies different path computation constraints before categorizing the ones considered in this chapter. Section 5.2 presents the algorithm, which is based on the K-shortest path algorithm as well as the exact solution of a binary integer program. Section 5.3 presents results obtained by simulating the proposed algorithm on real world networks with various traffic demand loads. Section 5.4 summarizes this chapter by highlighting its main contributions.

5.1 Overview of Constrained Shortest Path First problems

Path computation in general can be classified into various categories, which also leads to different solving methods. Section 5.1.1 presents a suggested taxonomy of path computation classes based on the constraints. Then section 5.1.2 defines precisely the problem for which the BIP of this chapter proposes a solution.

5.1.1 Taxonomy of path constraints

Table 5.1 classifies path constraints into five major categories. The prunable constraints are easy to solve; as the name suggests, the resources that do not satisfy the constraint are pruned before finding the shortest path using the remaining resources. Then there are the additive, non-additive, and adaptation constraints. The latter is a sub-class of non-additive constraints. These are harder to solve as they often lead to NP-hard problems. Then the policy constraints are a special class which can be applied on top of any other class. Thus, a policy could lead to a prunable constraint, or to an additive, or non-additive constraint. When discussing path computation schemes, a policy constraint can also describe the application of policies within the PCE architecture itself. Examples are applying policies per service to answer specific service requirements; applying policies during the selection of providers (inter-domain/inter-layer); applying policies to decide which constraints to apply for each type of LSP request depending on the LSP's switch type and SLA. Policies can also be used to impose a certain route for given ingress/egress nodes, and for better load balancing practices.

Table 5.1 Taxonomy of path computation constraints

Constraints' categories and typical examples		
Prunable	Bandwidth	Bandwidth and node capacity constraints require the pruning of the resources that do not satisfy these requirements
	Protection	In order to compute disjoint paths for protection purposes, the primary path's resources can be pruned
	Switching type	Switching type and encoding requirements allow the pruning of links and nodes that do not satisfy the requirements
	Processing time	The packet processing or service time of nodes can be a constraint solved by removing non compliant nodes
	Security	Some network resources may need to be avoided due to security risks, for example when they belong to another operator
Additive	Latency	The overall latency of a path is the sum of latencies induced by each link and node on the path
	Path length	Path length or the number of node hops is an additive constraint
	Optical impairments	Linear optical impairments like attenuation, dispersion, are additive constraints
Non-Additive	Wavelength continuity	Wavelength continuity constraint in an all-optical network
	Label continuity	Ethernet VLAN label continuity constraint
	Optical impairments	Non-linear optical impairments like cross-talk, or lambda availability based on adjacent channel usage are non-additive constraints
Adaptation	Lambda	Constraint imposed by nodes capable or not of converting from one wavelength to another
	Switching type	Constraint imposed by GMPLS nodes capable or not of converting from one switch type to another
Policy*	Applied directly to other constraints	For example some network resources may need to be pruned due to policy reasons; or each LSP could receive a different treatment given its importance, etc.

5.1.2 Switching adaptation capability constraints

The constraint path computation scheme in this chapter tackles the problem of finding the optimal path for a *source-destination-bandwidth-switch type* set that will satisfy GMPLS multi-layer/multi-region technological and traffic engineering constraints.

There are four possible actions or *adaptation actions* that a GMPLS node can perform

when connecting two signals. The most efficient possibility is to forward the traffic/signal using the same switch type, i.e., contiguous/stitched LSP. If this is not an option, the second more efficient choice is the nesting or un-nesting of the LSP to route the signal from one switch type to another. The last possibility is the hybrid node's capability to convert or adjust the LSP's switch type from one to another. Due to bandwidth granularity gaps when going up the switching hierarchy (going down the layers), and due to node resource consumption involved with this process, this should be left to a last recourse, that is be used only if no other path can be found by using the other adaptation actions. This can be considered as a traffic engineering rule. Another issue occurs when a node uses the nesting or

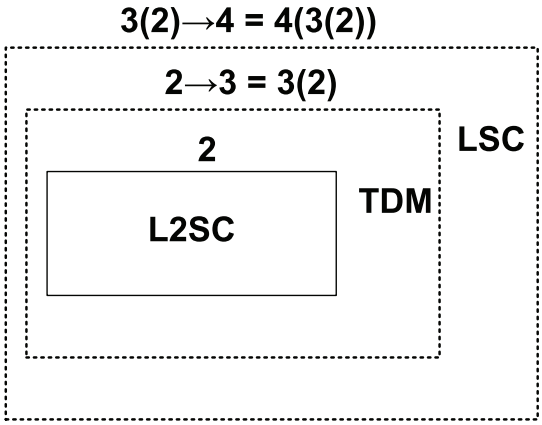


Figure 5.1 Example of multiple nesting of LSPs from different switching regions

un-nesting adaptation actions . When a LSP with switch type A is nested into type B, then somewhere along the path it needs to be un-nested from B back to A. Then, if more than one nesting/un-nesting adaptation actions are performed, the sequence in which these are done becomes critical. For example in Figure 5.1, if switch type 2 (L2SC) is nested into type 3 (TDM) which is in turn nested into type 4 (LSC), then the un-nesting has to be performed in the same order, that is un-nesting from 4 to 3, then from 3 to 2. This can be classified as an adaptation constraint.

Another technological issue occurs when the adaptation action is conversion. When a signal is converted to a lower layer signal, then it will occupy the minimum usable bandwidth of that layer's switch type. For example, a LSC signal is minimum one OC48/STM16. Thus, if an OC3/STM1 TDM signal is to be converted to LSC, it will waste the remaining bandwidth on the lambda used to carry it. This is a technological restriction that can be classified as an additive constraint. Again, this is the reason why nesting/un-nesting should be prioritized over conversion.

Another technological issue is that the end-to-end path must begin and terminate with the same switch type. This can be considered as a technological restriction an can be addressed as a pruning constraint. This requires assuring that the source and destination nodes support the switch type before making the path request.

5.2 Proposed Multi-Layer/Multi-Region Path Computation Algorithm

The problem consists of finding the optimal path for a *source-destination-bandwidth-switch type* set that will satisfy the above mentioned multi-layer/multi-region technological and traffic engineering constraints. Figure 5.2 gives a flowchart view of the procedure to be implemented in a PCE. Thus, the proposed algorithm for the search of the optimal path for a given request consists of three phases:

1. obtain all or a large number of paths by running the non-looping K-shortest path algorithm on the normalized network graph ;
2. for each shortest path found by the K-shortest path algorithm, optimize the cost of assignations of switch types and adaptation actions per node/link respecting technological and capacity constraints, by using the proposed binary integer program (BIP) ;
3. compare the objective function value of the optimal solution of each of the K shortest paths and select the minimum .

In the first step, the normalized graph refers to the network graph composed of all nodes, connected by a link if there is at least one link with a given switch type between them. The cost of the link on the normalized graph is set to 1 and therefore the K shortest paths are found based on the number of hops. Nevertheless, these costs could be set as inversely proportional to the delay, thus favouring paths with smaller delays. In this case, the K-shortest path algorithm needs to be re-run every time the congestion or delay state of a link changes. The second step's goal is mainly to determine the correct assignation of switch type per link and adaptation type(s) per node along each path. The BIP's solution is based on the costs associated with the use of each switch type per given link and each adaptation type per given node. The third step chooses the path with the minimum cost among the K paths. If there is a tie, then the number of hops is used to select the best path.

5.2.1 K-shortest path algorithm

The K-shortest path algorithm presented in section2.3.3 is used by the proposed algorithm. The computation of the K shortest paths allows for the pruning of links, nodes, paths

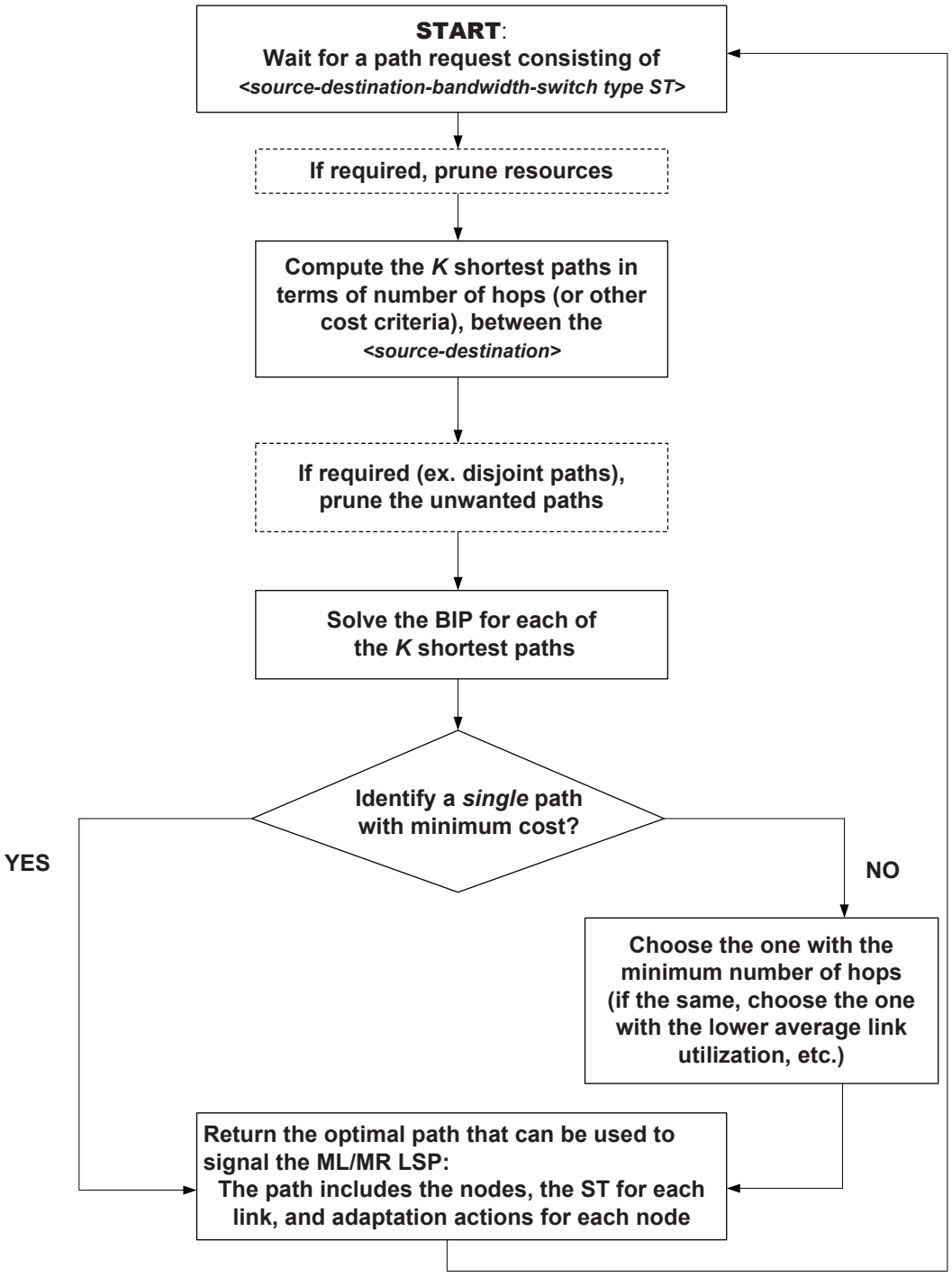


Figure 5.2 Proposed algorithm

and sub-paths. This is extremely useful for the computation of backup paths, which need to be disjoint from the primary path. Such flexibility is also useful for policy based exclusions. Also, since the topology of the network does not change as do the capacity and costs used

by the optimization model, this allows for faster response times since the K shortest paths on the normalized graph are pre-computed at the beginning and need to be recomputed only when a new node or a new physical link is added to the network.

5.2.2 Network model for the binary integer program

The multi-layer/multi-region network is represented by a graph G with V nodes and A links. The cost of each link depending on the switch type used is denoted by ω_{ij}^{st} for $(i,j) \in A$. Each switch type st is represented by a number $st = 1$ to 5 corresponding to the five switching regions defined in GMPLS: 1-PSC, 2-L2SC, 3-TDM, 4-LSC and 5-FSC. These represent the cost of using the given link with the given switch type. In the same manner, the capacity of each link in terms of bandwidth capacity is defined by C_{ij}^{st}. Then the binary variables x_{ij}^{st} are defined and equal 1 if the link (i,j) is used with switch type st and equal 0 otherwise.

For each of the simplex and hybrid GMPLS nodes in V, adaptation actions are defined. The possible adaptation actions are: 1- connect two LSPs in a contiguous manner, 2- perform nesting , 3-undo a nested LSP, or 4- convert from one switch type to another (only hybrid nodes). The cost of each adaptation action is denoted by $\Omega_{n,(st_i,st_j)}^{adapt}$ for $n \in V$ and $adapt \in \{1:4\}$. Here st_i represents the incoming port's switch type, and st_j represents the outgoing port's switch type; $(st_i, st_j) \in \Psi$ where Ψ is a 5×5 matrix of possible relations between the switch types. A simplex node can only perform the contiguous/stitching, nesting and un-nesting functions ($adapt = 1, 2, 3$), for the switch types it supports. A hybrid node can in addition to these, perform conversion of switch types between those it supports ($adapt = 4$). In the same manner, the capacity of each node expressed in bandwidth units for each adaptation action is defined by $\phi_{n,(st_i,st_j)}^{adapt}$. Then, the binary variables $y_{n,(st_i,st_j)}^{adapt}$ are defined and equal 1 if the node n uses adaptation action $adapt$ for incoming and outgoing switch types st_i and st_j respectively. Otherwise this variable equals 0.

5.2.3 Binary integer program model of the multi-layer/ multi-region path constraints problem

The binary integer program formulation of path computation with multi-layer/multi-region adaptation constraints is presented here. The network model presented in section 5.2.2 above is to be considered for each of the K shortest paths. The formulation of the path computation problem from source s to destination d is formulated as described below. The objective function to be minimized is the cost of adopting the set of switch types and adaptation actions represented by the binary variables x_{ij}^{st} and $y_{n,(st_i,st_j)}^{adapt}$.

Indices:

• Links are represented by $(i, j) \in A$;
• Switch type relations in a node are represented by $(st_i, st_j) \in \Psi$;
• Link's selected switch type is represented by $st \in \{1 : 5\}$;
• Node switching adaptation used is represented by $adapt \in \{1 : 4\}$;
• Set $\mathbb{P}$ of pre-computed shortest paths based on the number of hops, each path in $\mathbb{P}$ is represented by P_k for $k \in \{1 : K\}$;
• $\mathcal{P}_k$ is the set of $M - 1$ sub-paths of path P_k from p_k^1 to p_k^{M-1}, where P_k has M nodes. The first sub-path p_k^1 is composed of the first and second nodes in P_k; the last sub-path p_k^{M-1} $= P_k$.

Constants:

• b is the bandwidth requirement of the demand;
• ω_{ij}^{st} is the cost of using switch type st on link (i, j);
• C_{ij}^{st} is the capacity of link (i, j) for switch type st;
• min_{bw}^{ST} is the minimum bandwidth that can be signaled for a given switch type ST;
• $\Omega_{n,(st_i,st_j)}^{adapt}$ is the cost of using adaptation action $adapt$ from incoming switch type st_i to outgoing switch type st_j on node n;
• $\phi_{n,(st_i,st_j)}^{adapt}$ is the capacity on node n for adaptation action $adapt$ from switch type st_i to switch type st_j on node n;
• P_k is one of the possible paths in the set $\mathbb{P}$ of shortest paths, being treated by the BIP;
• $\delta_{i,j,k} = 1$ if link (i, j) belongs to path P_k, and 0 otherwise;
• $\zeta_{n,k} = 1$ if node n belongs to path P_k, and 0 otherwise.

Variables:

The variables that are to be optimally assigned are:
• x_{ij}^{st} binary variables which indicate the switch type used per link on the shortest path P_k;
• $y_{n,(st_i,st_j)}^{adapt}$ binary variables which indicate the adaptation actions(s) used per node on the shortest path P_k .

Objective function:

$$\text{minimize} \quad \sum_{(i,j)\in A} \sum_{st\in\{1:5\}} x_{ij}^{st} \cdot \omega_{ij}^{st} \quad + \sum_{adapt\in\{1:4\}} \sum_{n\in V} \sum_{(st_i,st_j)\in\Psi} y_{n,(st_i,st_j)}^{adapt} \cdot \Omega_{n,(st_i,st_j)}^{adapt} \tag{5.1}$$

Subject to constraints:

$$\sum_{st_i\in\{1:5\}} (\lceil \frac{b}{min_{bw}^{ST}} \rceil \cdot min_{bw}^{ST} - b) \cdot y_{n,(st_i,st)}^{adapt} + b \cdot x_{nj}^{st} \leq C_{nj}^{st} \ , \quad \forall_{\ (n,j)\in A,\ st\in\{1:5\}} \ \big|_{\ adapt=4,\ ST=\text{demand's ST}} \tag{5.2}$$

$$b \cdot y_{n,(st_i,st_j)}^{adapt} \leq \phi_{n,(st_i,st_j)}^{adapt}, \ \forall_{\ adapt\in\{1:4\},\ n\in V,\ (st_i,st_j)\in\Psi} \tag{5.3}$$

$$\sum_{st\in\{1:5\}} x_{ij}^{st} = \delta_{ij,k} \ , \quad \forall_{\ (i,j)\in A} \tag{5.4}$$

$$\sum_{adapt\in\{1:4\}} \sum_{(st_i,st_j)\in\Psi} y_{n,(st_i,st_j)}^{adapt} \cdot \zeta_{n,k} \geq \zeta_{n,k} \ , \quad \forall_{\ n\in V} \tag{5.5}$$

$$\sum_{adapt\in\{1:4\}} \sum_{(st_i,st_j)\in\Psi} y_{n,(st_i,st_j)}^{adapt} = 0 \ , \quad \forall_{\ n\in V} \ \big|_{\ \zeta_{n,k}=0} \tag{5.6}$$

$$\sum_{adapt\in\{1:4\}} \sum_{st_j\in\{1:5\}} y_{n,(st,st_j)}^{adapt} = 1 \quad \big|_{\ n=source,\ st=\text{demand's ST}} \tag{5.7}$$

$$\sum_{adapt\in\{1:4\}} \sum_{st_i\in\{1:5\}} y_{n,(st_i,st)}^{adapt} = 1 \quad \big|_{\ n=destination,\ st=\text{demand's ST}} \tag{5.8}$$

$$\sum_{adapt\in\{1:4\}} \sum_{st_i\in\{1:5\}} y_{n,(st_i,st)}^{adapt} \geq x_{n,j}^{st} \ , \quad \forall_{\ n,j\in V\ \big|\ n\neq destination\ ,\ st\in\{1:5\}} \tag{5.9}$$

$$\sum_{adapt\in\{1:4\}} \sum_{st_j\in\{1:5\}} y^{adapt}_{n,(st,st_j)} \geq x^{st}_{i,n} \ ,$$
$$\forall_{\ n,i\in V \ \mid n\neq source\ ,\ st\in\{1:5\}} \tag{5.10}$$

$$\sum_{adapt\in\{1:4\}} \sum_{st_i\in\{1:5\}} y^{adapt}_{n,(st_i,st)} + x^{st}_{n-1,n} = \sum_{adapt\in\{1:4\}} \sum_{st_j\in\{1:5\}} y^{adapt}_{n,(st,st_j)} + x^{st}_{n,n+1}$$
$$\forall_{\ n\in V,\ st\in\{1:5\}} \ \mid \text{for } st\text{=demand's ST, } x^{st}_{n-1,n}\text{=1 if } n\text{=}source\ ,\ x^{st}_{n,n+1}\text{=1 if } n\text{=}destination$$
$$,\ \text{otherwise}\ \ x^{st}_{n-1,n}\text{=0 if } n\text{=}source\ ,\ x^{st}_{n,n+1}\text{=0 if } n\text{=}destination \tag{5.11}$$

$$\sum_{adapt\in\{1,2,4\}} \sum_{(st_i,st_j)\in\Psi} y^{adapt}_{n,(st_i,st_j)} \leq 1 \ , \quad \forall_{\ n\in V} \tag{5.12}$$

$$\sum_{n\in V} y^{adapt}_{n,(st_i,st_j)} = \sum_{n\in V} y^{adapt^*}_{n,(st_j,st_i)} \ , \quad \forall_{\ (st_i,st_j)\in\Psi \ \mid\ adapt=2,\ adapt^*=3} \tag{5.13}$$

$$\sum_{n\in V^m_{sub}} y^{adapt}_{n,(st_i,st_j)} \geq \sum_{n\in V^{m+1}_{sub}} y^{adabt^*}_{n,(st_j,st_i)} \ ,$$
$$\forall_{V^m_{sub}\in p^m_k,\ p^m_k\in\mathcal{P}_k,\ m\in\{1:M-2\},\ (st_i,st_j)\in\Psi \ \mid\ adapt=2,\ adapt^*=3} \tag{5.14}$$

$$\sum_{(st_i,st_j)\in\Psi} y^{adapt}_{n,(st_i,st_j)} = 0 \quad \mid\ n=source,\ adapt=3 \tag{5.15}$$

$$\sum_{(st_i,st_j)\in\Psi} y^{adapt}_{n,(st_i,st_j)} = 0 \quad \mid\ n=destination,\ adapt=2 \tag{5.16}$$

$$x^{st,adapt}_{ij} \ \in\{0,1\} \ , \quad \forall_{\ (i,j)\in A,\ st\in\{1:5\},\ adapt\in\{1,2\}} \tag{5.17}$$

$$y^{adapt}_{n,(st_i,st_j)} \ \in\{0,1\} \ , \quad \forall_{\ n\in V,\ (st_i,st_j)\in\Psi,\ adapt\in\{1:4\}} \tag{5.18}$$

By transformation of the satisfiability problem to binary integer programming (Cook, 1971), this problem is proven to be NP-hard. However, since the number of binary integer variables is small, the problem can be solved to optimality for real-size instances rapidly.

Equation 5.1 is the objective function to minimize. It is the total cost of the path including all the links' costs given the switch type used and the adaptation costs used per node. By adjusting the costs per adaptation action, it is possible to prioritize one type of adaptation over the other (i.e., prioritize contiguous/stitching over nesting, and nesting over conversion). It is proposed to add a constant value α to the costs of the hybrid node's ($adapt = 4$, converting). Again, this is important since adaptation of a signal from a type to another is

more resource hungry than the signalling involved in nesting LSPs. Therefore it is assured that whenever possible, the simplex mode will be used over the hybrid mode. The α is calculated as per equation 5.19.

$$\alpha = \sum_{adapt \in \{1:3\}} \sum_{n \in V} \sum_{(st_i, st_j) \in \Psi} \Omega^{adapt}_{n,(st_i,st_j)} \tag{5.19}$$

The equations in 5.2 and 5.3 assure that the link $(i,j) \in A$'s capacity per switch type ($st \in \{1:5\}$) is respected as well as the adaptation ($adapt \in \{1:4\}$) capacity per switch type relation $(st_i, st_j) \in \Psi$ per node is respected. Equation 5.2 in particular considers that when adaptation action 4 (conversion) is used, then the demand's bandwidth needs to be adjusted to match the minimum possible bandwidth for the switch type ST. Equation 5.4 assures that only one switch type is selected per x^{st}_{ij} on path P_k. Equation 5.5 assures that minimum one adaptation action $y^{adapt}_{n,(st_i,st_j)}$ is selected per node n on path P_k. Equation 5.6 assures that zero adaptation $y^{adapt}_{n,(st_i,st_j)}$ has been selected for nodes not on path P_k. Equation 5.7 assures that the source node adapts the demand's switch type. Equation 5.8 assures that the destination node adapts back to the demand's switch type. Equation 5.9 assures that if a link x^{st}_{nj} is selected, then node n used an adaptation that converted to the st of x^{st}_{nj}. Equation 5.10 assures that if a link x^{st}_{in} is selected, then node n used an adaptation that converts from the st of x^{st}_{in}. Equation 5.11 assures that inside each node n, the adaptations performed on each switch type match in number. That is whenever an adaptation is performed from st to st_j, including the outgoing link $x^{st}_{n,n+1}$, this equality assures that st was available either from the incoming link ($x^{st}_{n-1,n}$) or from other adaptations ($y^{adapt}_{n,(st_i,st)}$) inside the node. Equation 5.12 restricts inequalities of equations 5.9 and 5.10 for adapt types other than nesting and un-nesting. It allows a node to do more than one adaptation only if it consists un-nesting. Equation 5.13 assures that along the path P_k, the sum of all *nestings* equals the sum of all *un-nestings* for each nesting $(st_i, st_j) \in \Psi$ and un-nesting (st_j, st_i). Equation 5.14 assures that on each sub-path $p^m_k \in \mathcal{P}_k$, the sum of *nestings* is greater than the sum of *un-nestings* for each nesting $(st_i, st_j) \in \Psi$ and un-nesting (st_j, st_i). Equations 5.17 and 5.18 are integrality constraints assuring that the solution variables are either selected (1) or not (0).

5.3 Performance Evaluation of the Proposed Multi-Layer/Multi-Region Path Computation Algorithm

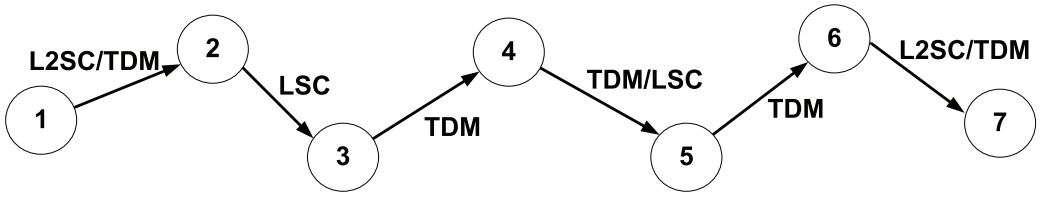
The proposed algorithm is implemented in MATLAB's 64 bit version 7.8.0.347 (R2009a). The choice of the language and simulation environment is in part due to the ability of the language to easily implement and debug routing algorithms and matrix manipulations.

Moreover MATLAB's Optimization Toolbox is used to solve the BIP. The BIP is tested with numerous test cases to validate its correctness. Three of these test cases are presented in section 5.3.1 below. Then, the complete algorithm is evaluated by simulations on real world networks, as presented below in section 5.3.2 which outlines the details and parameters used in the tests, and section 5.3.3 which presents the simulation results. For the simulations, the BIP part of the algorithm is solved using the branch-and-bound algorithm in MATLAB R2009's optimization toolbox, with branch strategy set to *maximum integer in infeasibility* and the node search strategy set to *best node search*. The simulations are mainly run on a computer with Intel Core2 Duo CPU P8400 (at 2.26GHz), with 4GB of RAM, running the 64 bit Windows 7 Professional operating system.

5.3.1 Testing the proposed binary integer program

Before presenting the simulation results, this section presents a few of the test cases used to verify the proposed BIP. The BIP gives the complete set of switch types and adaptation type(s) to use when signalling the LSP. The verifications assure that the given set respects all the technological and traffic engineering constraints, as discussed previously.

As a first example, Figure 5.3 presents a typical solution returned by the proposed BIP. Here a single request from $Node_1$ to $Node_7$ for switch type 2 (L2SC) is treated. All nodes are hybrid but if contiguous adaptation is not a possibility priority is given to nesting/un-nesting, as opposed to costly signal conversion. In this example the capacity is not a real issue, i.e., all links and nodes have available capacity, but in practice the available capacity information is crucial and can be collected by network monitoring systems in real time. The cost of each link, based on its switch type, increases when going up the hierarchy (down the layers). Figure 5.3(b) shows the exact configuration that is returned which respects adaptation constraints, multi-layer traffic engineering constraints (nest before convert), as well as link bandwidth and node adaptation capacity constraints. In this example the end-to-end LSP consists of receiving the demand traffic with switch type 2 (L2SC). Nesting from L2SC to TDM in $Node_1$: <adapt 2, st2->st3> . Then taking the TDM link from $Node_1$ to $Node_2$. Nesting from TDM to LSC in $Node_2$: <adapt 2, st3->st4>. Then taking the LSC link from $Node_2$ to $Node_3$, and un-nesting from LSC to TDM in $Node_3$: <adapt 3, st4->st3>. Then taking the TDM link from $Node_3$ to $Node_4$. Then using a contiguous LSP and taking the TDM links from $Node_4$ to $Node_5$: <adapt 1, st3->st3>, and from $Node_5$ to $Node_6$: <adapt 1, st3->st3>. Then un-nesting from TDM to L2SC in $Node_6$: <adapt 3, st3->st2>. Then taking the L2SC link from $Node_6$ to $Node_7$. $Node_7$ will just deliver the demand <adapt 1, st2->st2> directly, i.e., without further adaptation.



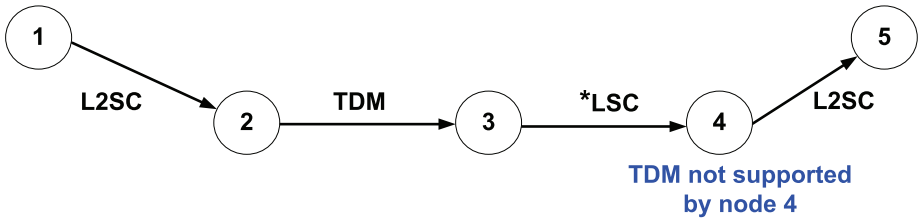
(a)

```
bintprog() called for demand from 1 -> 7 for BW= 1.5 ST = 2-+-+-+->
Optimization terminated.

Shortest path is :
~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.
             L      E       G       E       N       D
~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.
        GMPLS Switching Types:
        PSC: 1 L2SC: 2 TDM: 3 LSC: 4 FSC: 5
        ADAPT types:
        1: contiguous 2:nest 3:un-nest 4:convert(hybrid only)
~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.

[Incoming STYPE: L2SC]--------------------------> [NODE 1]

[NODE 1]<adapt 2, st2->st3> ---------{ST(3)}---------> [NODE 2]

[NODE 2]<adapt 2, st3->st4> ---------{ST(4)}---------> [NODE 3]

[NODE 3]<adapt 3, st4->st3> ---------{ST(3)}---------> [NODE 4]

[NODE 4]<adapt 1, st3->st3> ---------{ST(3)}---------> [NODE 5]

[NODE 5]<adapt 1, st3->st3> ---------{ST(3)}---------> [NODE 6]

[NODE 6]<adapt 3, st3->st2> ---------{ST(2)}---------> [NODE 7]

[NODE 7]<adapt 1, st2->st2>
```

(b)

Figure 5.3 Example of GMPLS path computation with the BIP algorithm

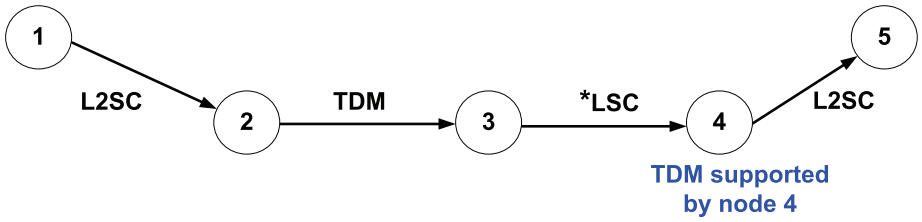
As a second example, Figure 5.4 presents an example where the the nesting and un-nesting constraints are put to test. It is important to note that $Node_3$ is not hybrid and cannot convert to LSC switch type (denoted by *LSC). In this example, the path request is from $Node_1$ to $Node_5$ for switch type 2 (L2SC). In Figure 5.4(a), $Node_4$ does not support TDM switch type whereas Figure 5.4(c) shows the same scenario where $Node_4$ supports TDM. Moreover, since $Node_3$ is not hybrid it can only nest to LSC. The solution of the BIP in Figure 5.4(b) is correct; here $Node_4$ receives a double encapsulation nested LSPs, [LSC[TDM[L2SC]]]. $Node_4$ can only un-nest from or to the switch types it supports. Therefore no solution is possible here because $Node_4$ does not support TDM. Then the BIP is tested with the example of



(a)

```
bintprog() called for demand from 1 -> 5 for BW= 1.5 ST = 2-+-+-+->
The problem is infeasible.
```

(b)



(c)

```
bintprog() called for demand from 1 -> 5 for BW= 1.5 ST = 2-+-+-+->
Optimization terminated.

Shortest path is :
~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.
              L       E       G       E       N       D
~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.
        GMPLS Switching Types:
        PSC: 1 L2SC: 2 TDM: 3 LSC: 4 FSC: 5
        ADAPT types:
        1: contiguous 2:nest 3:un-nest 4:convert(hybrid only)
~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.

[Incoming STYPE: L2SC]---------------------------> [NODE 1]

[NODE 1]<adapt 1, st2->st2> ---------{ST(2)}---------> [NODE 2]

[NODE 2]<adapt 2, st2->st3> ---------{ST(3)}---------> [NODE 3]

[NODE 3]<adapt 2, st3->st4> ---------{ST(4)}---------> [NODE 4]

[NODE 4]<adapt 3, st3->st2 3, st4->st3> ---{ST(2)}---> [NODE 5]

[NODE 5]<adapt 1, st2->st2>
```

(d)

Figure 5.4 A more complex example of GMPLS path computation with the BIP algorithm

Figure 5.4(c), which is the same network as Figure 5.4(a) but with the only difference that $Node_4$ supports TDM. It is seen that the result shown in Figure 5.4(d) is correct: $Node_4$ will un-nest from LSC to TDM, and then from TDM to L2SC before using the L2SC link between $Node_4$ to destination $Node_5$.

5.3.2 Simulation settings

The performance of the proposed algorithm based on the BIP is evaluated by comparing it to the GT method. Then, the proposed algorithm is evaluated for different values of K in the K-shortest path algorithm. Three performance parameters are used for the evaluations. First, the percentage of blocked path requests is measured. This is the ratio of successfully routed requests on the number of requests made during the simulation time. Second, the average and maximum values of the path costs are measured. The path cost depends on the switch type and adaptation type cost matrices. The BIP find a path that minimizes this cost. Third, the average and maximum values of the path length in terms of hops is measured.

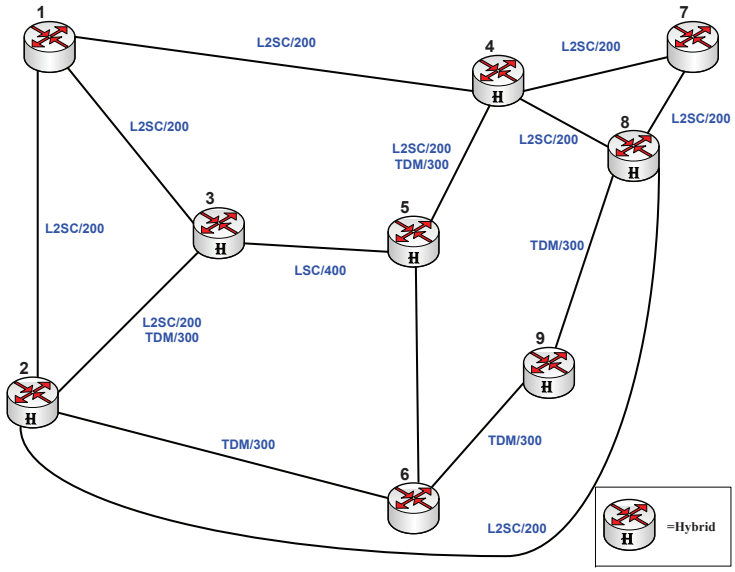
Topologies

The proposed algorithm is tested on two different networks, the Simplified Hybrid Optical and Packet Infrastructure (HOPI) Network as well as the National Science Foundation Network (NSFNET), shown in Figures 5.5(a) and 5.5(b) respectively. The HOPI network is the one used in Jabbari *et al.* (2007) and Gong et Jabbari (2008), who propose a graph transformation (GT) method.

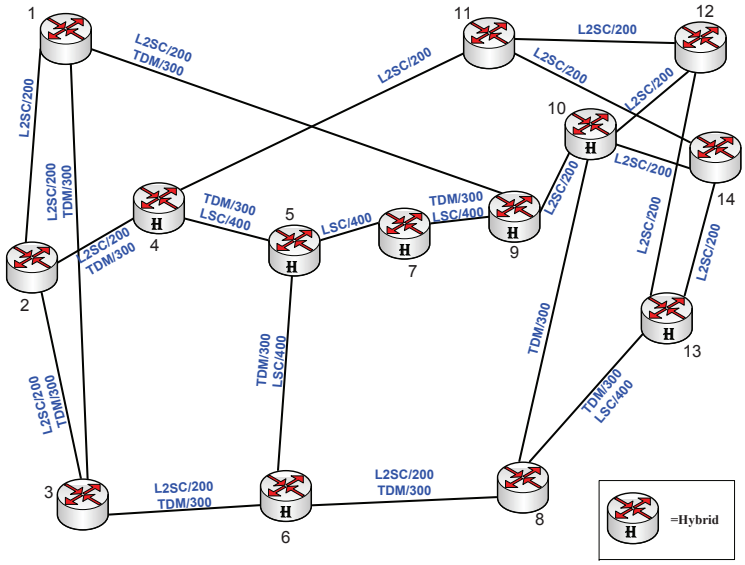
In these Figures, each connection between two nodes is labeled with the switch types it supports. Therefore when more than one switch types are supported, the connection could also be considered as separate links. Each switch type is followed by a cost that will be used by the BIP to minimize the overall cost of the selected switch types along the path. The cost and capacity values are set uniformly per switch type, as shown in Table 5.2. The capacities are multiplied by a factor of 10 to allow for a larger number of permanent LSPs in the demand sets.

Demand matrices

The results are obtained for a set of LSP setup requests randomly generated between different node pairs. To not falsify the results, each node pair is selected only if a physical path exists between them, and if the demand's switch type (randomly selected) is supported



(a) HOPI



(b) NSFNET

Figure 5.5 Networks used for the simulations

Table 5.2 Cost/Capacity

Region in HOPI	Link capacity	Link cost($)	adaptation capacity	adaptation cost($)
L2SC	3×OC3/STM1	200	3×OC3/STM1	200
TDM	OC12/STM4	300	OC12/STM4	300
LSC	OC48/STM16	400	OC48/STM16	400

by both the source and destination nodes. Table 5.3 presents the different test cases, and gives the number of demands, the minimum and maximum values of the demands, and the minimum usable bandwidth per switch type. This last factor is important when the adaptation action is conversion, thus causing certain bandwidth loss if the demand's bandwidth is less than the minimum usable value of the type to which it is converting to.

Table 5.3 Generated demand sets' parameters

Parameters used for generating the demand sets for HOPI and NSFNET:			
set	number	min,max BW per demand	min usable BW per layer [PSC L2SC TDM LSC FSC]
Set I	500	DS3,OC48/STM16	[0 0 OC3/STM1 OC48/STM16 OC192/STM64]
Set II	500	T1,OC48/STM16	[0 0 T1 OC48/STM16 OC192/STM64]
Set III	200	DS3,OC48/STM16	[0 0 DS3 OC48/STM16 OC192/STM64]

Table 5.4 presents the characteristics of the demand sets generated. The values represent average of source-destination pairs, which are all initially considered feasible, that is the source and destination support the switch type. The HOPI network has 9 nodes and the NSFNET has 14 nodes.

Table 5.4 Generated demands' characteristics

Source/Destination demand's BW			
Set	Network	average per demand (Mbit/s)	average total (Mbit/s)
I-	HOPI	113.54	1168.30
	NSFNET	226.40	1158.42
II-	HOPI	38.38	571.65
	NSFNET	193.93	1055.98
III-	HOPI	51.13	326.21
	NSFNET	222.87	554.14

For each demand represented by the *source-destination-bandwidth-switch type* set, the BIP is solved on each of the K shortest paths, the optimal result is the solution with the smallest objective function value. If there are two paths with the same optimal value of the objective function (path cost), the tie is broken by using the path with the minimum number of hops.

5.3.3 Simulation results

The results for the blockage, for the path costs, for the average and number of hops, for each of the three demand sets in both the HOPI and NSFNET networks are presented in

Figures 5.6 to 5.11. As discussed below, the BIP algorithm finds in general paths with lower costs because it explores the nesting/un-nesting possibilities. Moreover, in average BIP finds shortest paths in terms of number of hops. The performance of the BIP is comparable but with the advantage of finding feasible LSP paths considering all adaptation constraints and multi-layer/multi-region traffic engineering guidelines (i.e., nesting/un-nesting).

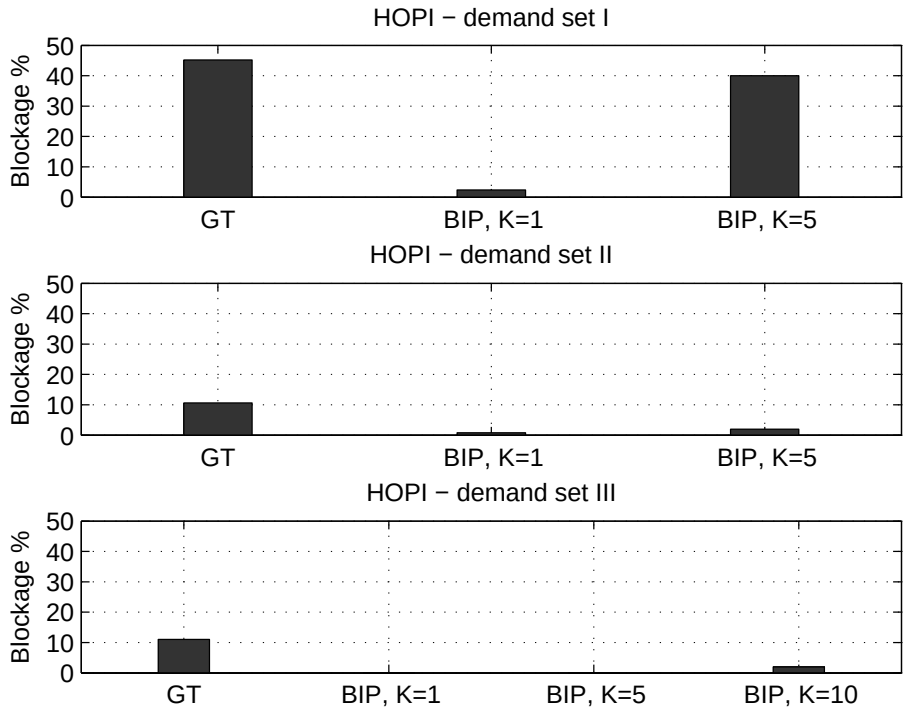


Figure 5.6 Percentage of blocked requests for various demand sets in HOPI

Percentage of blocked path requests

The throughput is considered higher when for the same traffic matrix the blockage rate is smaller. Therefore, in terms of throughput, Figures 5.6 and 5.7 show that in general the BIP method performs better than the GT method.

Referring back to Table 5.4 , in HOPI, demand set I is heavier than demand sets II and III. In NSFNET, demand sets I and II are slightly heavier than demand set III. When the traffic demand is heavier, the difference between the proposed method and the GT method becomes more prominent. This result is intuitive, in the sense that by prioritizing nesting as opposed to conversion, the proposed method saves in the bandwidth loss that occurs when

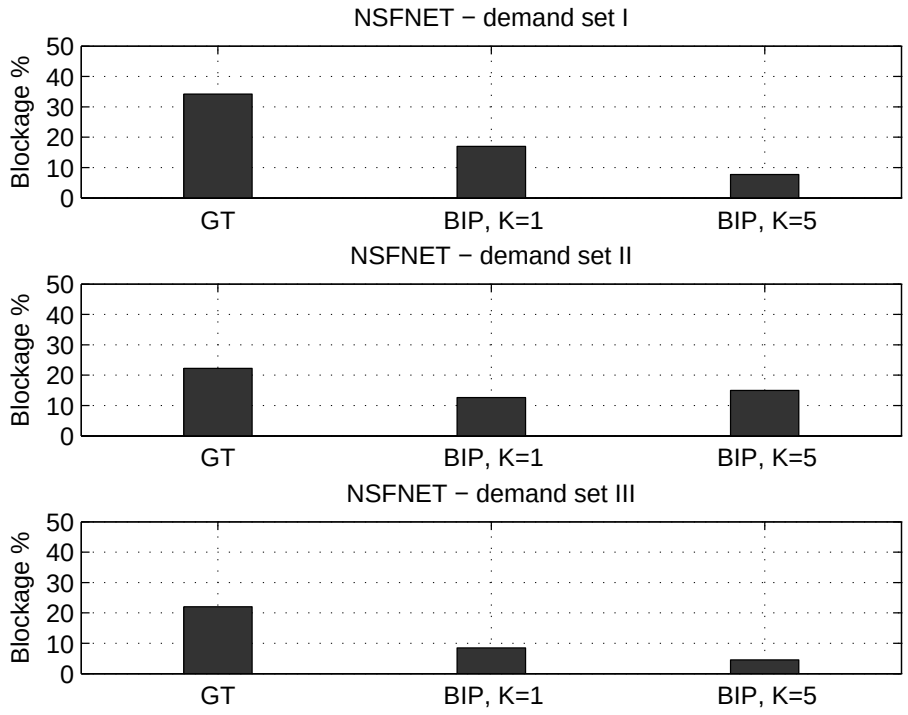


Figure 5.7 Percentage of blocked requests for various demand sets in NSFNET

the signal is converted to a lower layer (higher switch type). It should be recalled in the case of conversion, the minimum usable bandwidth of the lower layer has to be respected, which is often a lot higher than the demand's bandwidth, thus causing loss.

A very interesting result here is that increasing K does not necessarily produce a better end result in terms of blockage/throughput. The average cost, as it will be discussed below, is diminished. However in some cases, depending on the traffic set and its order, using a larger K may reduce the number of requests that can be accommodated. This situation is shown to its extreme in Figure 5.6 for demand set I, with $K = 5$. This is counter-intuitive, in the sense that the best performance was expected with a very large K, preferably large enough to account for all possible paths. However, depending on the cost matrices, this approach may result in average lower costs, but reduce the overall throughput. This means that there is a random relation between the best value for K and the end result in terms of throughput. Therefore when the main traffic engineering goal is to reduce blockage (increase the throughput), the best value for K can only be determined by performing the simulations for each set of demands, perhaps obtained from traffic forecasts.

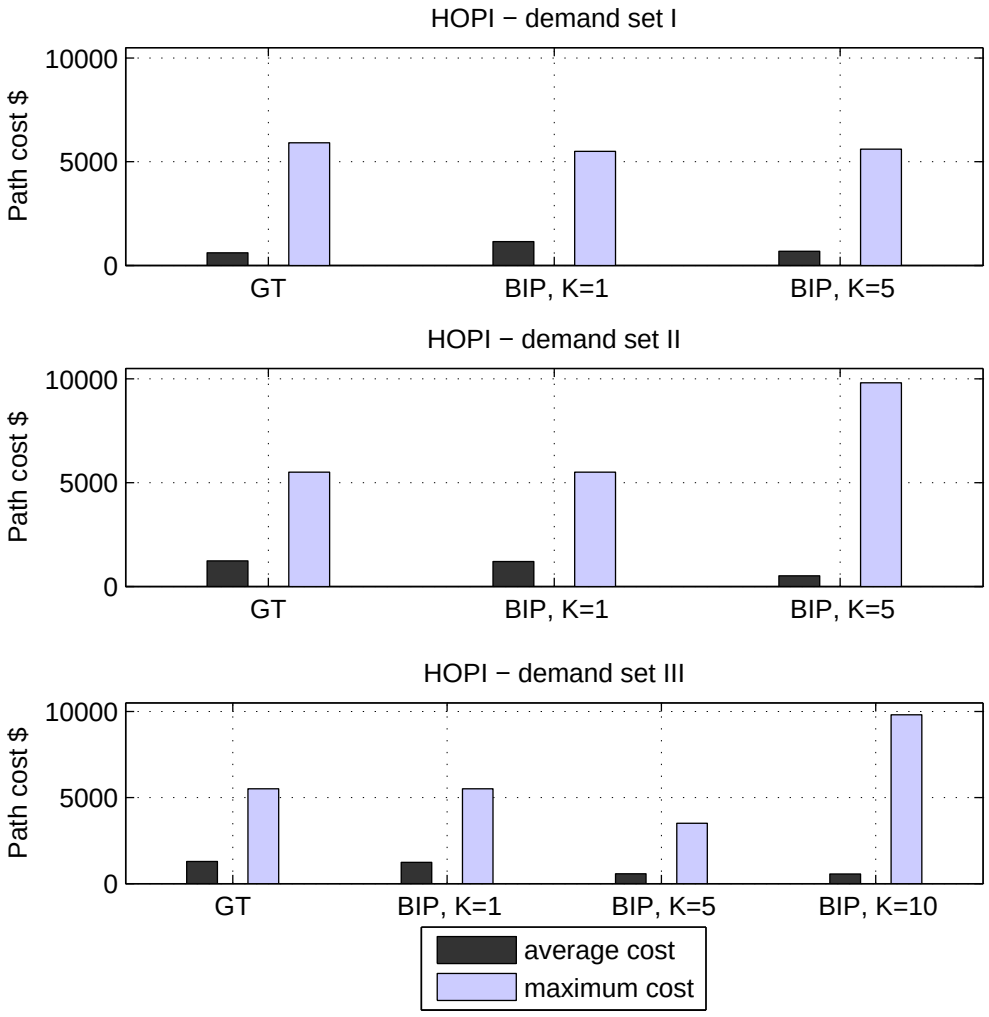


Figure 5.8 Path costs for various demand sets in HOPI

Average and maximum path costs

Figures 5.8 and 5.9 present the average and maximum path costs obtained in the HOPI and NSFNET networks for the same three demand sets. The comparison between the proposed BIP method and GT method has to be done by recalling the previous results on blockage. The fact that in some cases GT has a slightly lower average cost than the BIP method is explained by the lower number of requests that GT was able to accommodate. In fact, the demand set/K values for which the GT method has a smaller path cost correspond to the same demand set/K values for which it had a higher blockage rate.

However, when looking at lighter traffic sets which cause lower blockage rates, as expected the BIP method has lower average costs compared to the GT method. These path cost

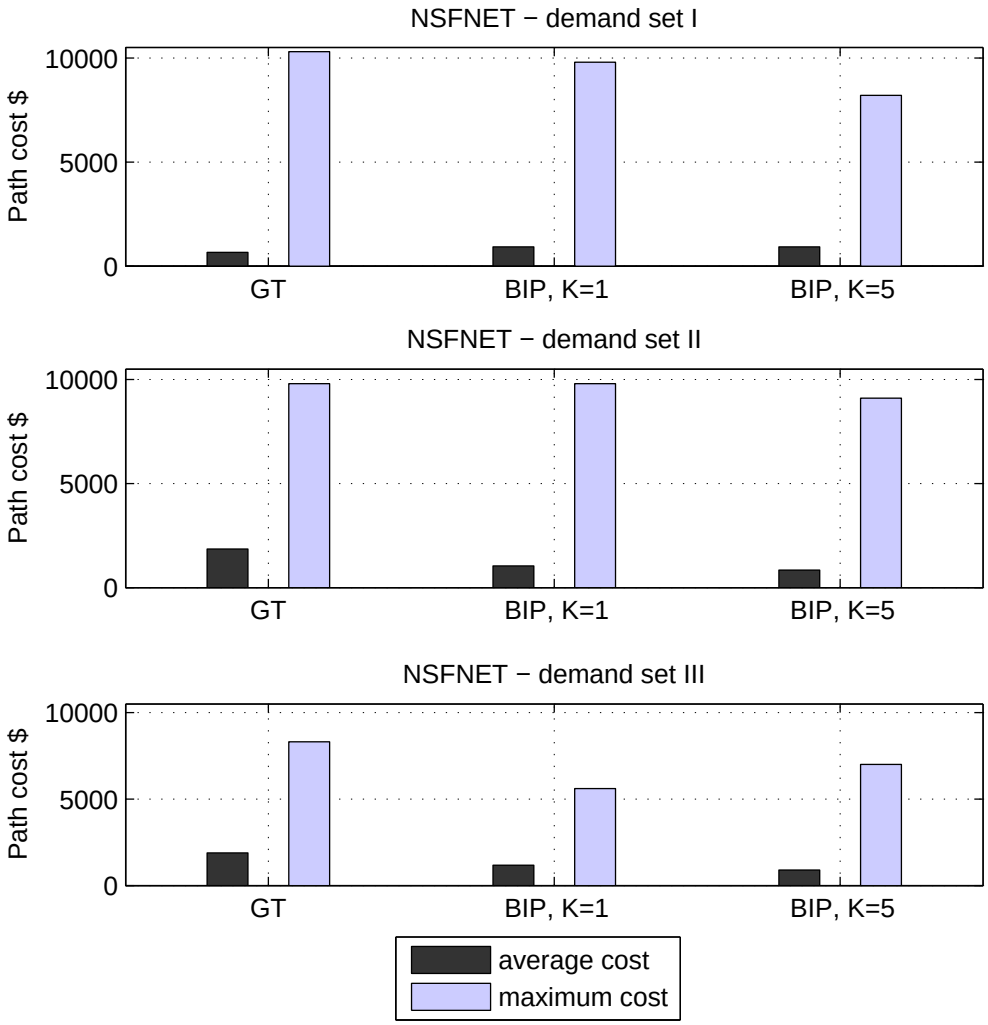


Figure 5.9 Path costs for various demand sets in NSFNET

values reflect the arbitrary link and adaptation costs chosen for ω_{ij}^{st} and $\Omega_{n,(st_i,st_j)}^{adapt}$ matrices. Therefore their actual values are not of great significance and they are presented just for sake of comparison between the different scenarios and methods.

Average and maximum number of hops

Figures 5.10 and 5.11 present the average and maximum number of hops per path. In both HOPI and NSFNET topologies, the overall number of hops is more or less similar when comparing different methods and values of K. Nonetheless, as GT is more restrained by only allowing conversion, it has a slightly higher number of hops. Moreover, by increasing K the number of hops increases slightly. This is expected due to the initial fact that the K shortest

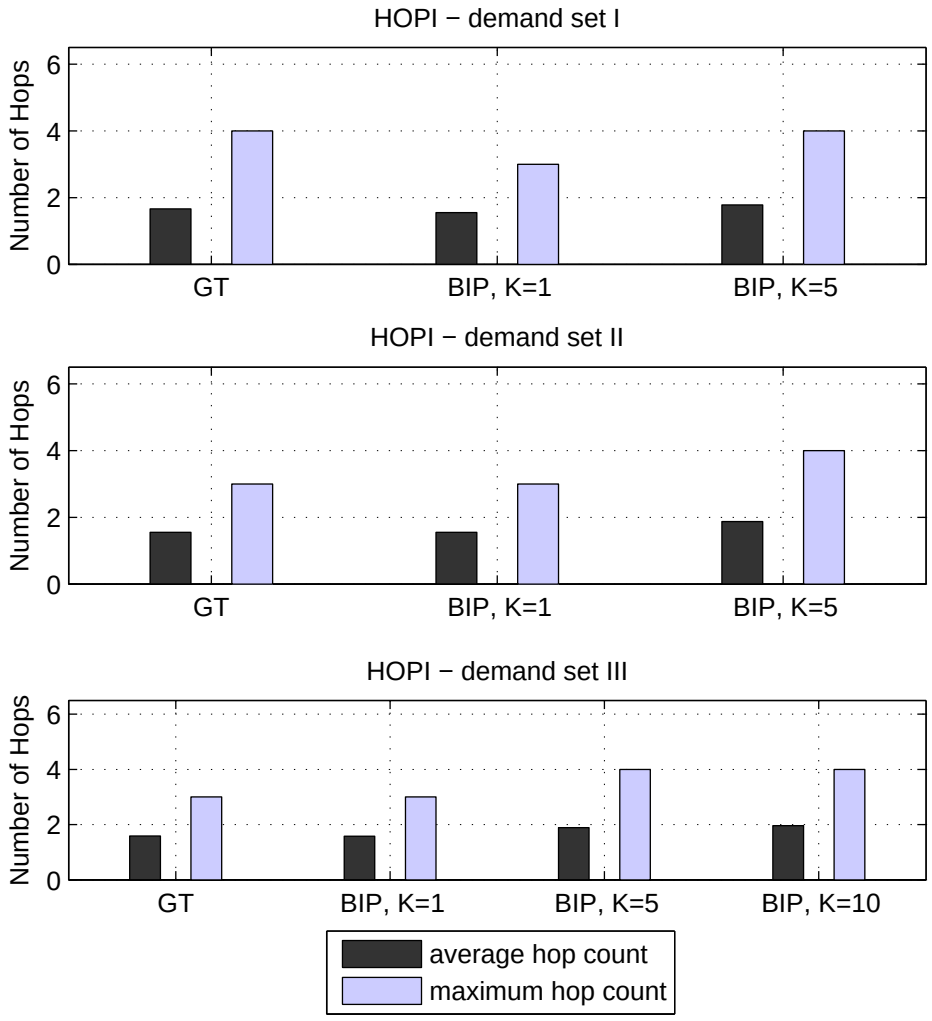


Figure 5.10 Hop count for various demand sets in HOPI

paths are found based on the number of hops. Thus, with bigger K values, the probability of the minimum objective function of the BIP to be found on a longer path is increased.

The effect of K on performance

Finally, Figures 5.12 and 5.13 study further the effect of K on the overall results on the HOPI and the NSFNET networks using their respective heavier demand set I. It is interesting to note that for both networks, increasing K does not give results that follow a certain trend. This confirms the mentioned previously randomness in the relation between the best value for K and the end results. These results are used to suggest the following guidelines in the selection of the value of K.

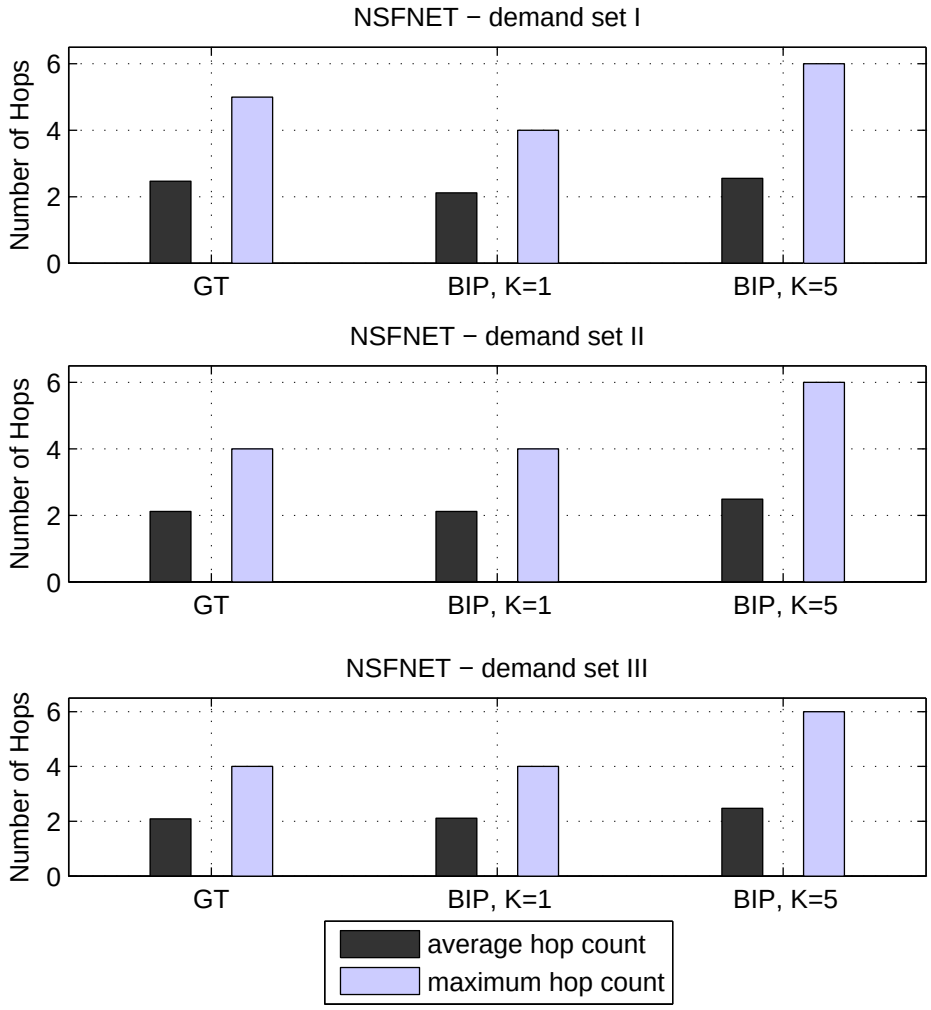


Figure 5.11 Hop count for various demand sets in NSFNET

If demand predictions are available, it is suggested to perform simulations to get a certain insight on the outcome of using different values of K. Then the right value for K can be selected based on these results given the traffic engineering goals that need to be achieved (e.g. better throughput, better path cost, smaller hop count, etc.). If demand predictions are not available, it is better to always use the shortest path, that is the first path in the K shortest paths. Then if a set of switch types and adaptations that satisfy all constraints does not exist on this path (i.e., no solution can be found), try with the second shortest path, and so on.

Nevertheless, as it will be discussed later and left as future work, assigning optimal costs to the ω_{ij}^{st} and $\Omega_{n,(st_i,st_j)}^{adapt}$ matrices has its influence. There is a relationship between the

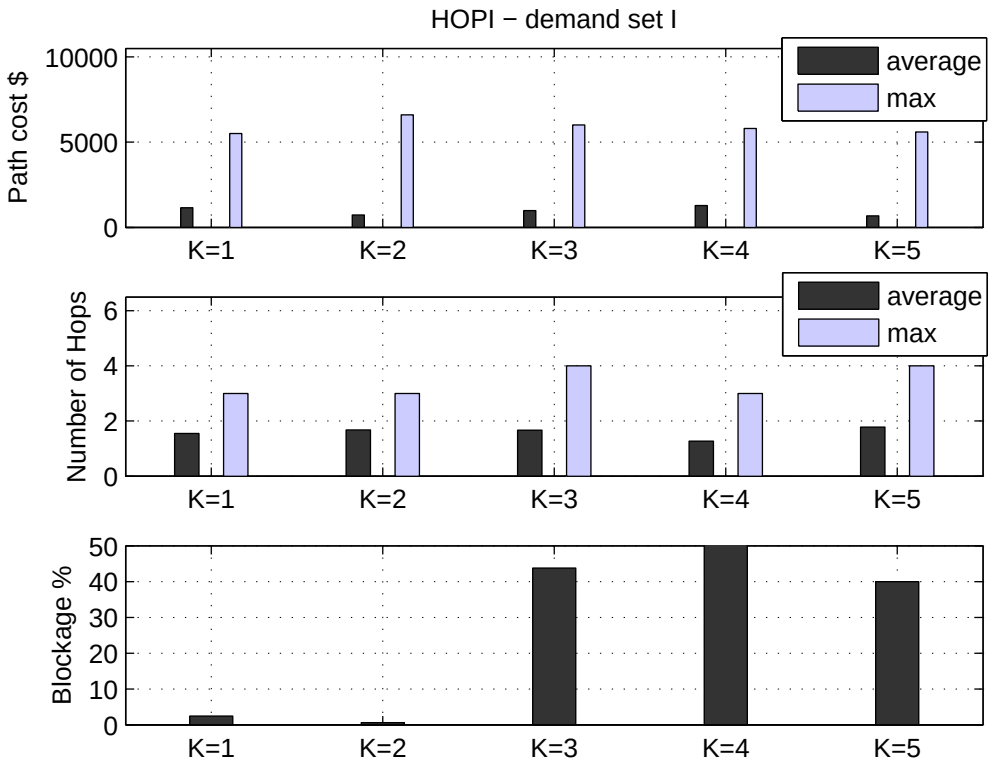


Figure 5.12 HOPI results for different K values

optimal K and the cost matrices that needs to be studied.

5.4 Summary

This chapter first redefines the complete problem of GMPLS inter-layer/inter-region path computation. Then it proposes a novel binary integer program that solves the constraints problem associated to switch type and adaptation action(s) assignations. The BIP is incorporated in the proposed algorithm which finds the optimal path for each *source-destination-bandwidth-switch type* path request. The algorithm will determine the optimal path (based on a cost function and the number of hops) with the assignations of switch type per link and adaptation actions(s) per node. The algorithm can be used to answer on-demand path computation requests and can be implemented in a PCE for dynamic path request and LSP deployment. It can also just be implemented on a separate node that can be used, for example, by the OAM group of an operator before signalling a LSP.

The results presented in this chapter are from the simulation of the algorithm on two

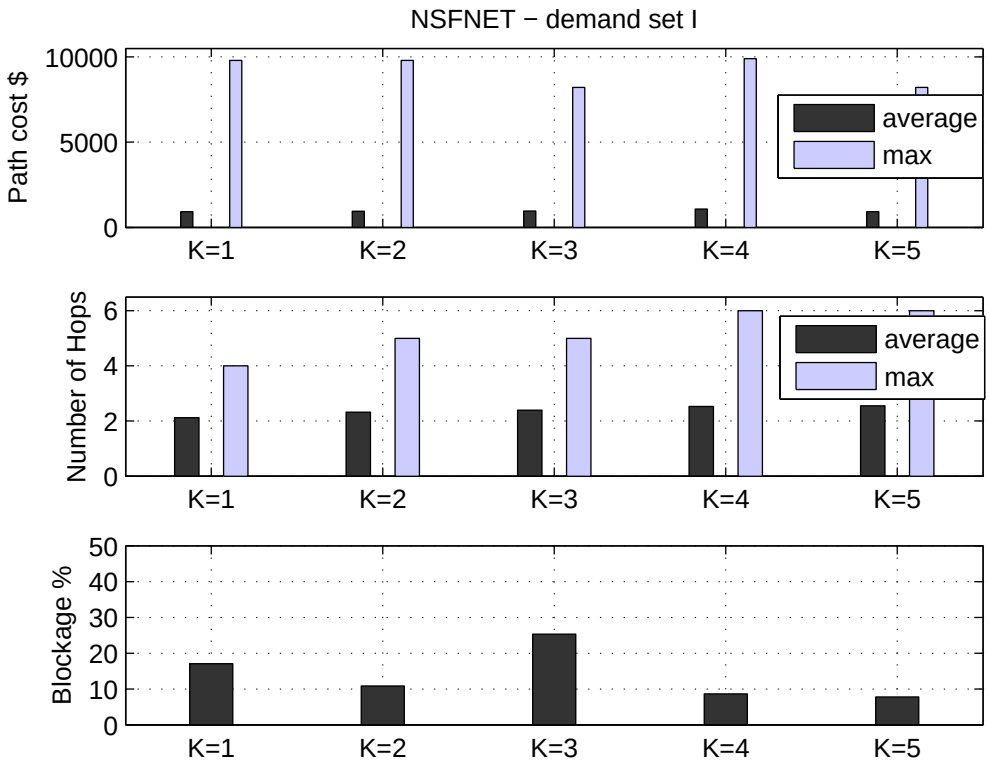


Figure 5.13 NSFNET results for different K values

real world networks. In terms of performance, the proposed algorithm does better than the existing graph transformation method. However its main strength remains the fact that it dynamically finds the assignations of switch type per link and adaptation action(s) per node that respect multi-layer/multi-region technological constraints as well as multi-layer traffic engineering best practices.

The binary integer program can be improved. Presently it has the drawback that it does not allow the scenario of nesting for example TDM(L2SC), then convert TDM to LSC to get LSC(L2SC) and then do un-nest of LSC(L2SC). This is not a real issue because this scenario is much stretched and is not yet confirmed to be allowed from a technological point of view. Another minor detail is that when more than one adaptation actions are to be performed, the model does not return their order. But, the order can easily be determined by looking at the incoming and outgoing links' switch types. A small script can be written to determine this order.

Finally, optimally assigning the costs to the ω_{ij}^{st} and $\Omega_{n,(st_i,st_j)}^{adapt}$ matrices can by itself be the subject of subsequent research.

CHAPTER 6

CONCLUSION

In the dawn of the all-IP Next Generation Networks (NGNs), the Internet is to be transformed and relatively large portion of the Internet traffic is to become Quality of Service (QoS) reliant. Traffic engineering is the only solution for QoS provisioning. However, assuring QoS in NGN is challenged not only by the well known weaknesses of the best effort IP networks, but also by the multi-domain/multi-layer facet of the world wide Internet and its transport networks. End-to-end QoS implies that the service level shall be sustained, not only across a single domain, but also often across multiple autonomous networks. Moreover, for each individual domain, end-to-end QoS implies that the service level shall be sustained across multiple technological layers. Then, there is also the typical scenario where both multi-layer/multi-domain problems occur concurrently.

This thesis tackled the problem of traffic engineered path computation in the context of inter-domain, inter-layer, and mixed inter-layer/inter-domain scenarios. The work in this thesis is subdivided into these three contexts, all falling under the umbrella of the Path Computation Element (PCE) architecture and GMPLS technology. The PCE architecture has been proposed by the Internet Engineering Task Force (IETF) to allow the computation of end-to-end inter-domain/inter-layer paths in a distributed manner among different PCEs. The PCE architecture proposes the use of GMPLS technology for the deployment of the traffic engineered paths, as inter-domain/inter-layer LSPs. This is due to the worldwide success of MPLS technology for QoS provisioning inside a single domain. MPLS was extended to its general form, GMPLS, and then for inter-domain and inter-layer reachability.

6.1 Review of Main Contributions

This thesis dealt with the end-to-end QoS provisioning problem under three separate parts: a distributed inter-domain scheme, a joint inter-layer/inter-domain path computation scheme and traffic engineering, and an inter-layer path computation algorithm for GMPLS networks. For clarity reason, the contributions for each of these parts are highlighted separately below, even though their dependence is trivial, i.e., all three aspects must be considered for true end-to-end QoS provisioning.

Inter-domain scheme

Chapter 3 presented a distributed pre-reservation based procedure for inter-domain path computation of LSPs within a PCE based architecture. The proposed approach is parallel to the IETF's BRPC standard (RFC5441) and uses the concept of resource pre-reservation during the computation of the end-to-end path. Simulation results were performed on a real world network. The scheme was shown to be effective regarding the overall dilemma between deployment blockages due to resource fluctuations and PCE path computation failures due to numerous pending pre-reservations. This chapter resulted in numerous patents and publications (Shirazipour et Pierre, 2009a; Shirazipour et Pierre, 2009b; Shirazipour et Lemieux, 2009; Shirazipour et Lemieux, 2006).

Inter-domain/inter-layer scheme

The work in chapter 4 is pioneer in the consideration of the joint inter-layer/inter-domain problem. This work presented in a novel way the actual real world situation where inter-layer/inter-domain scenarios occur simultaneously. It then provided a distributed PCE based path computation scheme to perform traffic engineering by adapting the inter-domain scheme of chapter 3 to the inter-layer environment. This part also considered the benefits of using traffic predictions, as opposed to existing works which mostly focus on the prediction algorithms. The overall traffic engineering scheme was evaluated by simulations on a real world network. Moreover, the benefits of using traffic predictions were studied. The obtained results showed that in fact there is no need for 100% accurate predictions and that the same benefits can be obtained by less accurate predictions. This is revelatory for the numerous works performed on traffic prediction. This chapter of the thesis resulted in a publication pending acceptance (Shirazipour et Pierre, 2010c), which presents the end-to-end traffic engineering technique applied in the backhaul of next generation mobile networks.

Inter-layer path computation algorithm

Chapter 5 presented a novel multi-layer path computation algorithm for GMPLS networks. This algorithm allows the consideration of specific adaptation constraints when computing multi-layer/multi-region LSPs. The algorithm was tested on two real world networks and was shown to outperform existing graph transformation based path computation techniques. The algorithm was also analyzed for a better understanding of its performance under different conditions. The findings in this chapter resulted in a publication pin press (Shirazipour et Pierre, 2010b) and another one pending acceptance (Shirazipour et Pierre,

2010a).

A general outcome of the findings of chapters 4 and 5 is in the form of an overall conclusion on traffic engineered admission control and path computation. The finding is that their outcome always depends on the order the requests arrive. This is analogous to the falling blocks in a Tetris game (Pajitnov, 1985). Depending on the size, shape, and order the blocks fall, it may be more difficult to optimally arrange them in a way that does not waste space. Therefore, demand and traffic forecasts are always useful to predict in advance which route and resource assignation scheme will result in the overall best results, in terms of utilization as well as performance predictability. However, given the specific application, the required accuracy of the forecasts should be determined prior to investigate on any prediction algorithm. This is important because, very often, the inability of obtaining very precise predictions has discouraged their use in traffic engineering techniques. For example, in the case of this work, average-like marketing forecasts are sufficient.

6.2 Limitations

The limitations were mentioned in the different chapters, but for sake of clarity and completeness, they are discussed again in detail in this section. Most of these limitations inspired the future works proposed in section 6.3.

Inter-domain scheme

The work in chapter 3 has a few limitations that can lead to interesting future projects. It must be recalled that the proposed method finds the optimal AS path if configured to consider all possible neighbouring ASes. However, if inter-domain PCEs are densely connected, this technique may not scale as it is similar to flooding. Moreover, the proposed scheme does not define any means for differentiating between usual intra-domain path optimization criteria versus inter-domain specific optimization criteria. Finally, the results obtained are from simulation because a test-bed of real world scale is not available. This could add some imprecision to the actual timing values collected, however it does not affect the relative comparisons made in this chapter.

Inter-domain/inter-layer scheme

Chapter 4 presented a joint inter-layer/inter-domain scheme for end-to-end path computation, which was tested on a real world network using simulations. It would be interesting

to implement this solution on a test bed network and measure the actual setup time of the inter-layer LSPs. Moreover, the results considered the IP packet switched layer on top of the optical layer. It would be interesting to test the proposed scheme on more than two layers. Another limitation is that the inter-domain aspect was considered through the inter-layer setting, i.e., the lower layer was considered to belong to another administration. It would be interesting to test real world situations where the scheme of chapter 3 would be implemented on top of the scheme of this chapter. The former could also be considered in cases where the transport layer triggers another transport layer for end-to-end connectivity. Another limitation is that the algorithm for obtaining prediction results was assumed to be existent; however, this is still an open issue in the literature. But, given the obtained results, the accuracy of such algorithm is not an issue, and thus any average-like forecasts shall work.

Inter-layer path computation algorithm

The work in chapter 5 has some limitations that could also lead to interesting future projects. Presently, the binary integer program (BIP) does not allow the scenario of, for example, nesting switch type STa to STb, then converting STb to STc and then un-nesting STc back to STa. This is not a real issue though, because the conversion performed by real world hybrid nodes may not stretch to such extreme scenarios. However, it remains an interesting challenge to propose another BIP to consider this possibility. Moreover, the cost matrices ω_{ij}^{st} and $\Omega_{n,(st_i,st_j)}^{adapt}$ were assigned by increasing values when going up the switching hierarchy. This chapter did not investigate ways to relate specific traffic engineering goals with cost assignations.

6.3 Future Research Directions

This section mostly builds on the limitations mentioned above and gives specific future research directions for each of the three contributions of this thesis as well as other related research topics.

Inter-domain scheme

To continue the work presented in chapter 3, a mechanism to optimally choose the pre-reservation timers could be defined and used to compare it with the proposed way of using hard pre-reservations with early tear down option. For this, prediction of optimal pre-reservation timers is a possible solution. Another possibility is to extend existing protocols

in order to carry the information needed across domains in order to optimally set the pre-reservation timers.

Future work shall also investigate the integration of request priorities into the proposed scheme to obtain overall utilizations and blocking rates of requests in different priority categories. It is also interesting to relate the request priorities to the solution for the correct setting of pre-reservation timers.

Another research direction is to investigate the consideration of QoS parameters other than the number of hops. This is different from the intra-domain scenario where each domain may give a different definition to various QoS parameters. Work in this area should set in motion the standardization of inter-domain QoS specifications. This way, each network operator could deploy its own techniques and definitions within its domain, and use conversion mechanisms at boundary nodes to allow the translation or mapping of inter-domain standard QoS requests into specific internal representations. Such solution requires scalability consideration at the inter-domain level.

Another important research direction is the optimal determination of the sequence of ASes in the computation of the inter-domain path. In addition, with a hierarchical solution, AS number exhaustion can be prevented. AS Number Translators (ANT) can be used within the proposed hierarchy to allow the existence of unpublished AS numbers.

A possibility for such solution is a distributed technique which relies on PCE hierarchies to compute the optimal AS path. Moreover, any solution should also propose a differentiation between optimal intra-domain versus optimal inter-domain paths. These differences shall serve to identify the criteria to be used in the optimization processes and algorithms. They can be implemented by the LSP differentiation scheme proposed in Shirazipour et Lemieux (2009). Some initial ideas on possible inter-domain path optimization metrics are enumerated below. These shall be satisfied on top of other intra-domain and inter-layer constraints.

1. minimize the total cost of the end-to-end inter-domain path ;
2. select the inter-domain path crossing the least number of domains (ASes) ;
3. maximize the available bandwidth on all the inter-domain links ;
4. select the inter-domain path crossing links with the lowest utilization ;
5. etc.

Another way to approach the optimal AS sequence determination is by considering the new routing architecture proposed at the IETF under the name of Locator/Identification Separation Protocol (LISP) proposed by Farinacci *et al.* (2010). LISP is a network based protocol which consists in the separation of IP addresses into Endpoint Identifiers (EIDs)

and Routing Locators (RLOCs). An AS sequence algorithm can be built using some of the new functionalities and databases introduced by the LISP standard.

Inter-domain/inter-layer scheme

The work and ideas presented in chapter 4 can naturally be extended to telecommunication cloud computing concepts and virtual network provisioning schemes. The idea is to offer virtual network services by looking at each service as a separate layer. Moreover, each layer shall be considered to ultimately belong to a separate provider, with top layers acting as clients to lower layers. This work proposed a way to compute traffic engineered paths among inter-layer/inter-domain networks, with the ultimate goal of minimum cost with best QoS and resource utilization. Thus, the adaptation of the distributed path computation schemes to new cloud computing provisioning schemes can be investigated as future work. In fact, the concept of network layers belonging to different providers is analogous to the concept of network-as-a-platform in the cloud computing paradigm.

Moreover, predicting traffic has always been subject of research interest. Depending on the type of network and traffic, the complexity of traffic prediction varies greatly. This is left for future work to investigate methods for effectively predicting the demand traffic, while considering the findings of this chapter which state that even 50% accurate predictions are beneficial and sufficient. Thus, this only leaves room for improving prediction algorithms in terms of time and resource efficiency as well as robustness.

Inter-layer path computation algorithm

As mentioned when discussing the limitations, the binary integer program in chapter 5 can be modified to allow the scenario of nesting switch type ST_a to ST_b, then converting ST_b to ST_c and then un-nesting ST_c back to ST_a.

Then, assigning the optimal values for the ω_{ij}^{st} and $\Omega_{n,(st_i,st_j)}^{adapt}$ cost matrices is a complete research subject by itself. This is comparable to the well investigated OSPF optimal weight assignment problem. There is a relationship that needs to be studied between different traffic engineering goals and the assignation of these costs.

Moreover, studying the effect of K in relation to the ω_{ij}^{st} and $\Omega_{n,(st_i,st_j)}^{adapt}$ cost matrices and various traffic engineering goals is left for future work. This is not to mention that the work in chapter 5 computed the K shortest paths based on the number of hops. This is also left for future investigation to see if the ω_{ij}^{st} or any other cost matrix should not be used when computing the K shortest paths.

On the other hand, it would be interesting to propose a search based method to perform the same path computation, considering the same set of constraints, by perhaps using the Tabu search heuristic. It would be interesting to compare the time taken between such algorithm and the one proposed in chapter 5.

Other end-to-end path computation challenges

Other than extensions to the three themes treated in this chapter, other research avenues for the provisioning of end-to-end QoS and path computation can be mentioned. One such important research direction is the computation of multicast trees and especially inter-domain multicast trees using the PCE architecture. This is an imminent problem for QoS provisioning for IPTV related services.

Another important research direction is the extension of the end-to-end model up to the access network. There must be hand off point where the GMPLS path terminates and the access network (cellular, WLAN, WiMAX, GPON, cable, etc.) takes charge of the QoS. At this point a mapping between core and backbone QoS parameter to access network QoS parameters needs to be defined. Also, the requirements of the access network in terms of QoS need to be defined in core and backbone (GMPLS) terms.

BIBLIOGRAPHY

ANJALI, T., SCOGLIO, C. et UHL, G. (2003). A new scheme for traffic estimation and resource allocation for bandwidth brokers. *Computer Networks*, 41, 761–777.

ASLAM, F., UZMI, Z. et FARREL, A. (2007). Interdomain Path Computation: Challenges and Solutions for Label Switched Networks. *IEEE Communication Magazine*, 45, 94–101.

AWDUCHE, D., BERGER, L., LI, T., SRINIVASAN, V. et SWALLOW, G. (2001). *RFC3209: RSVP-TE: Extensions to RSVP for LSP Tunnels*. IETF.

AYYANGAR, A., KOMPELLA, K., VASSEUR, J. et FARREL, A. (2008). *RFC5150: Label Switched Path Stitching with Generalized Multiprotocol Label Switching Traffic Engineering (GMPLS TE)*. IETF.

BERGER, L. (2003). *RFC3473: Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions*. IETF.

BLAKE, S., BLACK, D., CARLSON, M., DAVIES, E., WANG, Z. et WEISS, W. (1998). *RFC2475: An Architecture for Differentiated Services*. IETF.

BONAVENTURE, O., CNODDER, S. D., QUOITIN, B. et WHITE, R. (2003a). *Expired draft-ietf-grow-bgp-redistribution-00.txt: Controlling the redistribution of BGP routes*. IETF.

BONAVENTURE, O. et QUOITIN, B. (2003). *Expired draft-bonaventure-quoitin-bgp-communities-00.txt: Common utilization of the BGP community attribute*. IETF.

BONAVENTURE, O., QUOITIN, B., UHLIG, S., PELSSER, C. et SWINNEN, L. (2003b). Interdomain Traffic Engineering with BGP. *IEEE Communications magazine*, 41, 122–128.

BRADEN, R., CLARK, D. et SHENKER, S. (1994). *RFC1633: Integrated Services in the Internet Architecture: an Overview*. IETF.

BRADEN, R., ZHANG, L., BERSON, S., HERZOG, S. et JAMIN, S. (1997). *RFC2205: Resource ReSerVation Protocol (RSVP)*. IETF.

BRADFORD, R., VASSEUR, J.-P. et FARREL, A. (2009). *RFC5520: Preserving Topology Confidentiality in Inter-Domain Path Computation Using a Path-Key-Based Mechanism*. IETF.

BRYSKIN, I. et FARREL, A. (2006). *RFC4397: A Lexicography for the Interpretation of Generalized Multiprotocol Label Switching (GMPLS) Terminology within the Context of the ITU-T's Automatically Switched Optical Network (ASON) Architecture*. IETF.

COOK, S. (1971). The complexity of theorem-proving procedures. *In the Proceedings of the 3rd annual ACM symposium on theory of computing*, 151–158.

CORTEZ, P., RIO, M., ROCHA, M. et SOUSA, P. (2006). Internet Traffic Forecasting using Neural Networks. *International Joint Conference on Neural Networks*, 2635–2642.

DASGUPTA, S., DE OLIVEIRA, J. et VASSEUR, J.-P. (2007). Path-Computation-Element-Based Architecture for Interdomain MPLS/GMPLS Traffic Engineering: Overview and Performance. *IEEE Network*, 21, 38–45.

DIJKSTRA, E. W. (1959). A note on two problems in connection with graphs. *Numerische Mathematik*, 1, 269–271.

ELWALID, A., MITRA, D., SANIEE, I. et WIDJAJA, I. (2003). Routing and Protection in GMPLS Networks: From Shortest Paths to Optimized Designs. *Journal of Lightwave Technology*, 21, 2828–2838.

FARELL, A., VASSEUR, J.-P. et ASH, J. (2006). *RFC4655: A Path Computation Element (PCE)-based Architecture*. IETF.

FARINACCI, D., FULLER, V., MEYER, D. et LEWIS, D. (2010). Locator/id separation protocol (lisp). Internet-Draft (work in progress), draft-ietf-lisp-07.

FARREL, A., AYYANGAR, A. et VASSEUR, J.-P. (2008). *RFC5151: Inter-Domain MPLS and GMPLS Traffic Engineering - Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions*. IETF.

FARREL, A. et BRYSKIN, I. (2006). *GMPLS Architecture and Applications*. Morgan Kaufmann Publishers, San Francisco, première édition.

FARREL, A., PAPADIMITRIOU, D., VASSEUR, J. et AYYANGAR, A. (2006a). *RFC4420: Encoding of Attributes for Multiprotocol Label Switching (MPLS) Label Switched Path (LSP) Establishment Using Resource ReserVation Protocol-Traffic Engineering (RSVP-TE)*. IETF.

FARREL, A., SATYANARAYANA, A., IWATA, A., FUJITA, N. et ASH, G. (2007). *RFC4920: Crankback Signaling Extensions for MPLS and GMPLS RSVP-TE*. IETF.

FARREL, A., VASSEUR, J.-P. et AYYANGAR, A. (2006b). *RFC4726: A Framework for Inter-Domain Multiprotocol Label Switching Traffic Engineering*. IETF.

FU, B. (2009). *Traffic Engineering and Quality of Service in the Internet*. Bath, UK.

GONG, S. et JABBARI, B. (2008). Optimal and efficient end-to-end path computation in multi-layer networks. *ICC'08, IEEE International Conference on Communications*, 5767–5771.

HANCOCK, J. (2006). Sndlib - library of test instances for survivable fixed telecommunication network design. `http://sndlib.zib.de`.

HARHIRA, H. et PIERRE, S. (2008). A novel admission control mechanism in GMPLS-based IP over optical networks. *Computer Networks*, 52, 1281–1290.

HUANG, S., DUTTA, R. et ROUSKAS, G. (2006). Traffic grooming in path, star, and tree networks: complexity, bounds, and algorithms. *IEEE Journal on Selected Areas in Communications*, 24, 66–82.

HUSTON, G. (2009). As numbers - again. `http://www.potaroo.net/ispcol/2009-08/asagain.html`.

JABBARI, B., GONG, S. et OKI, E. (2007). On constraints for path computation in multi-layer switched networks. *IEICE Transactions on Communications*, E90-B, 1922–1927.

KING, D., LEE, Y., XU, H. et FARREL, A. (2008). Path computation architectures overview in multidomain optical networks based on ITU-T ASON and IETF PCE. *NOMS'08, IEEE Network Operations and Management Symposium Workshops*, 219–226.

KOMPELLA, K. et REKHTER, Y. (2005a). *RFC4202: Routing Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS).* IETF.

KOMPELLA, K. et REKHTER, Y. (2005b). *RFC4206: Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE).* IETF.

LEBLANC, L., CHIFFLET, J. et MAHEY, P. (1999). Packet routing in telecommunication networks with path and flow restrictions. *INFORMS J. on Computing*, 11, 188–197.

LEROUX, J.-L. (2007). *RFC4927: Path Computation Element Communication Protocol (PCECP) Specific Requirements for Inter-Area MPLS and GMPLS Traffic Engineering.* IETF.

LEROUX, J.-L., VASSEUR, J.-P. et LEE, Y. (2009). *RFC5541: Encoding of Objective Functions in Path Computation Element communication Protocol (PCEP).* IETF.

MANNIE, E. (2004). *RFC3945: Generalized Multi-Protocol Label Switching (GMPLS) Architecture.* IETF.

MANTAR, H., HWANG, J., OKUMUS, I. et CHAPIN, S. (2004). A Scalable Model for Inter bandwidth Broker Resource Reservation and Provisioning. *IEEE Journal on selected areas in communications*, 22, 2019–2034.

MARTINEZ, R., R.MUNOZ, SORRIBES, J. et JUNYENT, J. C. G. (2005). Experimental GMPLS-based dynamic routing in all-optical wavelength-routed networks. *ICTON'05, Proceedings of 2005 7th International Conference on Transparent Optical Networks*, 193-6, Tu.A1.5.

MOUFTAH, H. et NAAS, N. (2008). A novel MILP formulation for planning GMPLS transport networks with conversion and regeneration capabilities. *CCECE'08, IEEE Canadian Conference on Electrical and Computer Engineering*, 569–574.

NAKAGOME, Y. et MORI, H. (1973). Flexible routing in the global communication network. *ITC7'73, 7th International Teletraffic Congress*, 426.1–426.8.

OKI, E., TAKEDA, T., LEROUX, J.-L. et FARREL, A. (2009). *RFC5623: Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering*. IETF.

OKUMUS, T., HWANG, J., MANTAR, H. et CHAPIN, S. (2001). Inter-domain lsp setup using bandwidth management points. *Globecomm'01, IEEE Global Communications Conference*, 7–11.

PAJITNOV, A. (1985). The tetris game 1985-2010. `http://www.tetris.com`.

PAN, P. (2002). *Scalable Resource Reservation Signaling in the Internet, PhD Thesis*. New York, NY, USA.

PAN, P., SWALLOW, G. et ATLAS, A. (2005). *RFC4090: Fast Reroute Extensions to RSVP-TE for LSP Tunnels*. IETF.

PAOLUCCI, F., F.CUGINI, VALCARENGHI, L. et CASTOLDI, P. (2008). Enhancing backward recursive PCE-based computation (BRPC) for inter-domain protected LSP provisioing. *Conference on Optical Fiber Communication/National Fiber Optic Engineers Conference OFC/NFOEC 2008*, 1–3.

PAPAGIANNAKI, K., TAFT, N., ZHANG, Z. et DIOT, C. (2005). Long-Term Forecasting of Internet Backbone Traffic. *IEEE Transactions on Neural Networks*, 16, 1110–1124.

PELSSER, C. et BONAVENTURE, O. (2003). Extending rsvp-te to support inter-as lsps. *HPSR'03, Workshop on High Performance Switching and Routing*, 79–84.

PIÓRO, M. et MEDHI, D. (2004). *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann Publishers, San Francisco, première édition.

ROSEN, E., VISWANATHAN, A. et CALLON, R. (2001). *RFC3031: Multi-protocol Label Switching Architecture*. IETF.

SANCHEZ-LOPEZ, S., MASIP-BRUIN, X., SOLE-PARETA, J. et DOMINGO-PASCUAL, J. (2007). Fast setup of end-to-end paths for bandwidth constrained applications in an IP/MPLS-ATM integrated environment. *Computer Networks*, 51, 835–852.

SANGLI, S., TAPPAN, D. et REKHTER, Y. (2006). *RFC4360: BGP Extended Communities Attribute*. IETF.

SHIOMOTO, K., OKI, E., IMAJUKU, W., OKAMOTO, S. et YAMANAKA, N. (2003). Distributed Virtual Network Topology Control Mechanism in GMPLS-Based Multiregion

Networks. *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, 21, 1254–1262.

SHIOMOTO, K., PAPADIMITRIOU, D., LEROUX, J.-L., VIGOUREUX, M. et BRUNGARD, D. (2008). *RFC5212: Requirements for GMPLS-Based Multi-Region and Multi-Layer Networks (MRN/MLN).* IETF.

SHIRAZIPOUR, M. et LEMIEUX, Y. (2006). Inter-domain qos reservation, establishment and modification. International Patent No. PCT/IB2006/053271. Filing date: November 1, 2005. Publication date: September 13, 2006.

SHIRAZIPOUR, M. et LEMIEUX, Y. (2009). Inter-domain traffic engineering. US Patent No. 7593405B2. Filing date: October 14, 2005. Publication date: September 22, 2009.

SHIRAZIPOUR, M. et PIERRE, S. (2009a). Reducing TE-LSP setup time by minimizing blockage with the use of pre-reservations during the path computation process. *CCECE'09, IEEE Canadian Conference on Electrical and Computer Engineering*, 610–613.

SHIRAZIPOUR, M. et PIERRE, S. (2009b). Study of Resource Pre-Reservation for Label Switched Paths Computed within a Path Computation Element Environment. *ISMCN'09, International Symposium on Mobile Computing and Networking*, 30–34.

SHIRAZIPOUR, M. et PIERRE, S. (2010a). GMPLS Multi-Region Path Constraints. *Submitted to IEEE Communications Letters, Manuscript ID: CL2010-1082.*

SHIRAZIPOUR, M. et PIERRE, S. (2010b). Multi-Layer/Multi-Region Path Computation with Adaptation Capability Constraints. *To be published in the Proc. of GC'10, IEEE Global Communications Conference, Miami, USA, ref. No. 1569311558.*

SHIRAZIPOUR, M. et PIERRE, S. (2010c). Traffic Engineering for Next Generation Mobile Backhaul Networks with Joint Inter-Layer/Inter-Domain Considerations. *Submitted to the Proc. of WiMob'10, IEEE Conference on Wireless and Mobile Computing, Networking and Communications, Canada, ref. No. 1569325673.*

SONG, Q., HABIB, I. et ALANQAR, W. (2005). Performance evaluation of connection setup in GMPLS IP optical network. *OFC/NFOEC'05, Conference on Optical Fiber Communication*, 3, 169–171.

SZIGETI, J., TAPOLCAI, J., CINKLER, T., HENK, T. et SALLAI, G. (2004). Stalled information based routing in multi-domain multilayer networks. *11th International Telecommunications Network Strategy and Planning Symposium*, 297–302.

TOMIC, S. (2007). *Dynamic Virtualization and Service Provision in Multi-Provider GMPLS Networks, PhD Thesis.* Austria.

TSIRAKAKIS, G. et CLARKSON, T. (2009). Simulation Tools for Multilayer Fault Restoration. *IEEE Communications Magazine*, 47, 128–134.

VASSEUR, J.-P., AYYANGAR, A. et ZHANG, R. (2008). *RFC5152: A Per-Domain Path Computation Method for Establishing Inter-Domain Traffic Engineering (TE) Label Switched Paths (LSPs)*. IETF.

VASSEUR, J.-P. et LEROUX, J.-L. (2009). *RFC:5440: Path Computation Element (PCE) Communication Protocol (PCEP)*. IETF.

VASSEUR, J.-P., ZHANG, R., BITAR, N. et LEROUX, J.-L. (2009). *RFC5441: A Backward Recursive PCE-based Computation (BRPC) Procedure To Compute Shortest Constrained Inter-domain Traffic Engineering Label Switched Paths*. IETF.

VERCHERE, D., BROCKMANN, S. et VIGOUREUX, M. (2006). Prereservation of resources for connecting paths in a packet-address-switched or label-switched communication network. US Patent No. 20,060,274,658. Filing date: May 25, 2006. Publication date: December 7, 2006.

WANG, N., HO, K., PAVLOU, G. et HOWARTH, M. (2008). An Overview of Routing Optimization for Internet Traffic Engineering. *IEEE Communications Surveys 1st Quarter*, 10, 36–56.

X.YANG, LEHMAN, T., OGAKI, K. et OTANI, T. (2009). A study on cross-layer multi-constraint path computation for IP-over-optical networks. *ICC'09,IEEE International Conference on Communications*.

X.ZHANG, KIM, S. et LUMETTA, S. (2007). Reduced flow routing: Leveraging residual capacity to reduce blocking in GMPLS networks. *Proceedings of the 4th International Conference on Broadband Communications, Networks, Systems, BroadNets*, 394–403.

YEN, J. (1971). Finding the k shortest loopless paths in a network. *Management Science*, 17, 712–716.