Traffic grooming and rerouting in multi-layer WDM network

Elias Doumith

École Nationale Supérieure des Télécommunications
École Doctorale d'Informatique, Télécommunications et Électronique de Paris

Thèse

Pour obtenir le grade de

Docteur de l'École Nationale Supérieure des Télécommunications
Spécialité: Informatique et Réseaux

Présentée par

Elias A. DOUMITH

Agrégation et Reroutage de Trafic dans les Réseaux WDM Multi-couches

(Traffic Grooming and Rerouting in Multi-layer WDM Network)

Soutenue le 14 Mai 2007 devant le jury

Président et Rapporteur	
Prof. Achille Pattavina	Politecnico di Milano, Italy
Rapporteur	
Prof. Biswanath Mukherjee	University of California, Davis, USA
Examinateurs	
Dr. Olivier Audouin	Alcatel-Lucent Research & Innovation, France
Prof. Mario Pickavet	Universiteit Gent, Belgium
Directeur de thèse	
Prof. Maurice Gagnaire	École Nationale Supérieure des Télécommunications, France

To my family with love,
To my father, my best friend in the whole wind world
To my mother who encourages me every step of the way
To my brother who makes my life much more interesting
To my friends who have always been a support
To all those who helped me accomplish this thesis

To the One
whose eyes have always been my inspiration
whose smile always makes my day
whose heart always guides my way
who bears the pain and showers me with care
whom I love the most and will love forever

Acknowledgement

“ Managing risk is easier when

you have good people on board, . . .

both team members and outside advisors.”

“ Life is easier when

you are part of a neighborhood, . . .

both family and friends.”

This thesis is the result of three and half years of work whereby I have been accompanied and supported by many people. It is a pleasure to have now the opportunity to express my gratitude for all of them.

My hearty gratitude goes to my outstanding research advisor Professor Maurice Gagnaire who not only guided and encouraged me throughout my thesis, but also greatly influenced me on the attitude towards my career and life. He has been everything that one could want in an advisor. I feel privileged to have such experience.

I am deeply indebted to my committee members Professor Achille Pattavina and Professor Biswanath Mukherjee for their time and effort in reviewing this work. I convey special acknowledgement to Professor Mario Pickavet for serving on my thesis committee. I also gratefully thank Doctor Olivier Audouin at Alcatel-Lucent Research and Innovation for his collaboration and valuable guidance during the two-year project I was involved in.

I would also like to express my sincere thanks to Doctor Nicolas Puech for his help and valuable advice. He always kindly grants me his time for answering some of my questions. Many thanks go to Doctor Mohammed Koubaa for the fruitful discussions. Working together was a great experience and a great pleasure.

I thank all of the members of the Networks and Computer Science Department of the École Nationale Supérieure des Télécommunications (ENST) for creating a dynamic and collaborative environment for research and study.

I would like to acknowledge my best friends Mireille Sarkiss and Sawsan Al Zahr for reviewing my thesis and providing valuable feedback. Many thanks for the kind help in the preparation and the

administration of the successful ceremony following the thesis defense. Thank you for the delicious Lebanese food. Thanks to Rosy Aoun for sharing various thoughts during the coffee breaks and for creating such a great friendship at the office. I am extraordinarily fortunate to have Charbel Abdel Nour and Ihsan Fsaifes as friends. They were always there for me and always helped me through my most difficult days. I would like also to thank Chadi Abou Rjeily and Rawad Zgheib for their friendship and for all the great times we shared together. Good luck to each of you in your future endeavors.

Last but certainly not the least, I am deeply and forever indebted to my parents for their never-ending love, constant support and immeasurable encouragement throughout my entire life. I am also very grateful to my brother Charbel. I am proud to acknowledge the generous and enduring prayers of all my family throughout the years of efforts toward this dissertation.

Finally, I would like to thank everybody who contributed to the realization of this thesis and I apologies if I could not mention all of them.

Résumé

Bien avant l'invention du téléphone par Antonio Meucci (1860) et son brevetage par Alexander Graham Bell (1876), les premiers systèmes de télécommunications, tel que le télégraphe (1838), utilisaient déjà les fils de cuivre comme support de transmission. Plus tard, grâce à Maxwell (1873) et Hertz (1887), la transmission des messages a emprunté la voie radio (ondes électromagnétiques). Dans les années 1970, le principe de la fibre optique a été développé dans les laboratoires de l'entreprise américaine Corning Glass Works. Dès lors, la transmission d'un signal lumineux à travers un milieu transparent est devenue possible. Entourée d'une gaine protectrice, la fibre optique peut être utilisée pour conduire de la lumière entre deux lieux distants de plusieurs centaines, voire milliers, de kilomètres. Le signal lumineux codé par une variation d'intensité est capable de transmettre une grande quantité d'information. Dans une première phase (1984 à 2000), la fibre optique s'est limitée à l'interconnexion des centraux téléphoniques qui étaient les seuls à nécessiter de forts débits. Grâce aux performances avantageuses qu'elle permet, la fibre optique s'est répandue de plus en plus à l'intérieur des réseaux de télécommunications. On estime qu'aujourd'hui plus de 80% des communications à longue distance sont transportées le long de plus de 25 millions de kilomètres de câbles à fibres optiques partout dans le monde. De nos jours, avec la baisse des coûts entraînée par sa fabrication en masse et les besoins croissants des particuliers en très haut débit, on envisage le déploiement de la fibre optique dans les réseaux d'accès jusqu'aux particuliers.

Tour d'Horizon sur les Réseaux Optiques

Établir un réseau de télécommunications par fibre optique fait appel à de nombreuses compétences. Optimiser un tel réseau tant au niveau des performances qu'au niveau du coût demande des évolutions et des avancées notables dans des domaines aussi variés que la fabrication, la physique, l'informatique, l'optique, ... Depuis ses débuts, plusieurs études ont permis l'amélioration de l'efficacité et des performances des réseaux optiques pour répondre aux besoins croissants des usagers en bande passante. Les premières études se sont focalisées sur le routage et l'affectation de longueurs d'onde dans les réseaux optiques, le dimensionnement de réseau, l'ingénierie de trafic, le reroutage et la reconfiguration de la topologie logique du réseau, la protection et la restauration des demandes, et les réseaux d'accès optique.

De la planification à l'ingénierie de trafic dans les réseaux optiques: Un réseau peut être défini comme un ensemble de ressources mises en place afin de fournir un ensemble de services. Ces ressources se présentent principalement sous la forme de nœuds comprenant un routeur et des émetteurs-récepteurs; les nœuds étant connectés entre eux par des fibres optiques. La première question qui se pose est comment router des flux de données entre ces différentes ressources. Dans ce but, il faut définir pour chaque couple de nœuds source-destination un ou plusieurs chemins formés par l'aboutement au niveau physique d'un certain nombre de fibres optiques. Cependant, contrairement à leurs homologues cables classiques qui ne transportent qu'un seul flux de données, les fibres peuvent avoir autant de flux que de longueurs d'onde supportées. En conséquence, il faut aussi associer à chaque fibre d'un chemin donné une longueur d'onde. Si les nœuds intermédiaires le long de ce chemin n'autorisent pas la conversion de longueur d'onde, la longueur d'onde associée doit être la même sur toutes les fibres empruntées.

Du point de vue d'un opérateur, le problème du routage des flux de données revient à mettre en place un réseau en adéquation avec l'utilisation future de ses clients tout en maintenant un faible coût de déploiement et de maintenance. Une fois le réseau déployé, l'opérateur est amené à utiliser au mieux les ressources disponibles de ce réseau pour satisfaire ses clients. En d'autres termes, l'objectif de l'opérateur consiste à maximiser la quantité de requêtes satisfaites ou à minimiser le coût lié à la capacité utilisée au niveau physique tout en maintenant une certaine qualité de service et de fiabilité dans le réseau. Pour cela, différentes techniques d'optimisation existent:

a) **Dimensionnement/Planification de réseau:** La classe des problèmes de dimensionnement de réseau englobe tous les problèmes dont l'objectif est l'installation d'un ensemble de liens de capacités données, afin de pouvoir écouler un ensemble de demandes connu. C'est un processus visant le long terme et définissant le programme de l'investissement et du déploiement des équipements du réseau. Les demandes de trafic prises en compte sont habituellement considérées comme statiques et correspondent aux prévisions, à long terme, de la charge du réseau. Les problèmes de planification de réseau sont généralement formulés comme des problèmes d'optimisation.

b) **Ingénierie de réseau:** La classe des problèmes d'ingénierie de réseau concerne les problèmes de l'attribution efficace des ressources existantes du réseau aux demandes de trafic. Les demandes de connexion sont connues et le réseau est installé mais les ressources doivent être configurées et affectées aux demandes. Parmi toutes les configurations possibles, on cherche celle qui maximise l'efficacité de l'utilisation des ressources (par exemple, en termes de rentabilité du déploiement des ressources). Ce problème est généralement formulé comme un problème d'optimisation. Dans ce cas, les demandes sont dynamiques avec une durée de vie allant d'une heure à une journée.

c) **Ingénierie de trafic:** Le problème d'ingénierie de trafic est similaire à celui de l'ingénierie de réseau mais pour des demandes plus dynamiques de durée allant de quelques minutes à plusieurs heures et dont la fréquence d'arrivée est élevée. Dans ce cas, les demandes sont imprévisibles et le traitement des demandes se fait d'une manière séquentielle. Les applications les plus courantes concernent le routage des flux autour des points de congestion connus dans le réseau et le contrôle précis du reroutage de trafic affecté par un incident sur le réseau. D'une manière générale, l'ingénierie de trafic a pour objectif l'usage optimal de l'ensemble des liens physiques du réseau en évitant la surcharge de certains liens et la sous-utilisation d'autres.

Le reroutage et la reconfiguration de la topologie logique: Lors du déploiement d'un réseau de communication de type WDM, les opérateurs dimensionnent les liens du réseau en fonction de la connaissance du trafic actuel et d'une estimation de sa croissance. Cette évolution du trafic correspond à l'ajout ou au retrait de connexions. Chaque requête retirée libère des ressources qui pourront être utilisées pour l'ajout de nouvelles requêtes. Lors de l'ajout de nouvelles connexions, il s'agit de trouver une route et une longueur d'onde dans le réseau sans modifier les connexions déjà établies. Dans certaines configurations, il est impossible d'établir une connexion pour une requête bien que le problème de trouver un chemin entre le nœud source et le nœud destination de la requête en question admet une solution. Pour contourner cette difficulté, il faut développer des stratégies de reconfiguration du réseau (déplacement de connexions sur d'autres routes) permettant une meilleure utilisation des ressources. Cette reconfiguration de la topologie logique et son adaptation au niveau de la topologie physique sont des points clés des réseaux de télécommunications modernes. La reconfiguration peut être déclenchée à intervalle régulier ou la conséquence d'un certain phénomène comme la coupure d'un lien physique ou le blocage d'une demande. Dans ce paragraphe, nous nous limitons au cas de la reconfiguration de la topologie logique suite à un blocage d'une certaine demande. Cela a pour but de réarranger les connexions courantes et ainsi de libérer de l'espace pour la demande qui a initié cette reconfiguration. Le cas de la reconfiguration de la topologie logique suite à une coupure d'un lien ou à une panne d'un certain commutateur sera traité dans le paragraphe suivant sur la protection et la restauration du trafic.

Plusieurs algorithmes de reconfiguration existent et tous cherchent à minimiser la période où le trafic est perturbé. On cite à titre d'exemple l'approche "Move to vacant wavelength retuning" (MTV-WR) qui cherche à calculer une nouvelle topologie logique qui ne soit différente de la topologie actuelle que par un nombre réduit de demandes à rerouter. Cette approche cherche à rerouter certaines demandes existantes sans changer leurs chemins physiques mais en leur affectant des nouvelles longueurs d'onde. Elle se caractérise par une période de perturbation du réseau très courte sans aucune perte de donnés. Elle peut être implémentée en représentant le réseau par un graphe et en associant des coûts adéquats aux différents arcs du graphe. Il est à noter que cette approche est sans aucun avantage dans le cas où les nœuds du réseau sont équipés de convertisseurs de longueur d'onde.

La protection et la restauration: Le trafic sur un réseau de cœur peut atteindre des débits de plusieurs terabits par secondes. La panne d'un lien du réseau, par exemple une coupure physique d'une fibre lors de travaux de voirie ou la panne d'un équipement optique, résulte en la perte d'un volume important de données. Pour un opérateur l'enjeu économique est crucial, et les conséquences d'interruptions de connexions peuvent être dramatiques lorsqu'elles se traduisent par le non respect de contrats et des dédommagements financiers importants. Deux façons opposées mais complémentaires d'aborder une panne coexistent dans les réseaux: la "protection" et la "restauration". Le rôle de ces mécanismes est d'éviter l'interruption des services en redirigeant les demandes par d'autres chemins que ceux prévus initialement.

a) **Restauration:** La restauration consiste à rerouter dynamiquement des connexions lorsqu'une panne survient sur le réseau. Il faut alors déterminer, en fonction des ressources encore disponibles dans le réseau au moment de la panne, un nouveau chemin pour chaque connexion interrompue. Les approches utilisées sont basées sur des algorithmes "en ligne" puisque les pannes ne sont pas connues à l'avance. En règle général, elles ne garantissent pas que toutes les connexions puissent être rétablies après la panne. On retrouve ce mécanisme de survie au niveau de la couche IP: les chemins suivis par les paquets sont calculés en fonction de l'état du réseau au fur et à mesure de leur progression d'un nœud à l'autre. Tout équipement indisponible est tout simplement non pris en compte dans le calcul de ces chemins.

b) **Protection:** L'idée essentielle des méthodes de protection est de prévoir, au moment d'établir une connexion entre deux nœuds, plusieurs chemins de sorte qu'il en reste toujours au moins un chemin qui soit opérationnel en cas de panne. À partir de ce principe, de nombreuses variantes ont vu le jour afin d'atteindre le meilleur compromis entre le délai de rétablissement du service après une panne et les ressources requises. Par exemple, on distingue la protection de bout-en-bout, ou « *path protection* », de la protection autour de la panne, ou « *span protection* ». La protection dite de bout-en-bout consiste à rerouter les connections interrompus par la panne sur des chemins de secours calculés entre les nœuds sources et les nœuds destinations de connexions considérées. Par contre, la protection autour d'une panne consiste à rerouter localement les connexions affectées par la panne en contournant uniquement l'équipement en panne. On distingue aussi la protection dédiée de la protection partagée. La protection dédiée nécessite d'allouer un chemin de secours qui ne peut être réutilisé dans aucun autre contexte. À l'inverse, la protection partagée permet d'utiliser une même ressource par les chemins de secours de deux ou plusieurs connexions qui ne peuvent pas tomber en panne en même temps. Il est à noter que, dans les réseaux implémentant la protection, certaines connexions peuvent être refusées au moment de leur routage s'il est impossible de leur garantir la disponibilité d'un chemin de secours.

En résumé, on parle de "protection" lorsque l'on propose des stratégies préventives, et de "restauration" lorsque l'on s'intéresse à des stratégies curatives qui agissent une fois que la panne est survenue. Il faut noter que le préventif est coûteux en termes économiques alors que le curatif est plutôt coûteux en termes de qualité de service.

Les réseaux d'accès optique: Il y a 10 ans, les usagers se satisfaisaient de modems à 56 Kbit/s. Le transfert de fichiers volumineux comme des images ou de la vidéo, était encore très peu répandu. Cependant, au fur et à mesure de la pénétration des techniques de communication dans la société contemporaine, de nouveaux services tels que la sauvegarde distante des données, la visioconférence, la vidéo à la demande, la téléphonie sur IP, le très haut débit mobile, etc. sont proposés aux usagers et la demande de débit subit une croissance exponentielle. En réponse à cette demande croissante en débit, les supports câblés ont évolué afin d'offrir toujours plus de capacité de transmission. Mais, ces solutions câblées ont atteint aujourd'hui leur limite en termes de débit et ne peuvent garantir une bonne qualité de service que pour les usagers assez proches des centrales de distribution. La fibre optique semble alors être le seul média capable de transporter des débits bien plus importants. De plus, la fibre optique est vue comme le prolongement naturel des réseaux à grande distance qui l'utilisent largement aujourd'hui.

Les technologies des réseaux optiques passifs, ou « *Passive Optical Network* » (PON), constituent aujourd'hui une référence en matière de réseaux d'accès très haut débit dans la mesure où elles concilient très forte capacité de transport et minimisation des infrastructures fibres nécessaires. La technologie PON permet de distribuer à tous les abonnés d'un secteur géographique donné un flux descendant sur une longueur d'onde et de faire remonter vers le réseau les flux des abonnés sur une longueur d'onde différente. Afin de réduire les coûts d'infrastructure, poste prépondérant dans les investissements, les réseaux PON innovent par l'utilisation de composants passifs sur le parcours de la fibre optique. Ces composants ou coupleurs permettent de diviser le signal optique sur plusieurs branches secondaires, également en fibre optique, pour constituer un arbre de transmission passif. Les informations seront donc démultipliées dans leur trajet vers l'utilisateur final à la manière classique des réseaux d'acheminement d'eau, de gaz ou encore d'électricité. Dans le sens utilisateur-réseau, les signaux optiques issus des différentes branches sont additionnés. Des mécanismes d'allocation permettent d'éviter les collisions entre informations émises par les utilisateurs en partageant l'usage de la ressource optique dans le temps (multiplexage temporel).

Les réseaux PON ont fait l'objet de procédures de normalisation au niveau international par les principaux organismes de normalisation: l'ITU (International Telecommunication Union), le FSAN (Full Service Access Network), et l'IEEE (Institute of Electrical and Electronics Engineers). Ils sont d'ores et déjà significativement déployés dans le monde, notamment en Amérique du Nord et en Asie.

De nos jours, la recherche dans le domaine des réseaux optiques est beaucoup plus poussée et elle couvre des sujets plus pointus tel que l'agrégation de demandes de trafic, la prise en compte de la qualité de transmission (QoT) dans les réseaux optiques WDM, les réseaux grilles, la radio sur fibre, la protection et la restauration multi-couches, et le codage réseau.

L'agrégation de trafic: La bande passante disponible dans les réseaux modernes est gigantesque, mais les consommateurs de cette ressource restent des équipements électroniques (serveurs web, visio-conférence, téléphonie, télévision, $\cdots$) qui sont loin de l'exploiter complètement. Attribuer à chaque requête/demande de trafic un chemin optique et une longueur d'onde entière entraîne une sous-utilisation des ressources du réseau. Une alternative est d'avoir recourt à l'agrégation/groupage de trafic ou « *Traffic Grooming* ». Le groupage de trafic est le terme générique utilisé pour le problème qui consiste à agréger deux ou plusieurs flux de faible débit en un flux de débit supérieur. Au niveau physique, l'agrégat des flux élémentaires est considéré comme une seule requête et est alors routé sur un chemin unique et une longueur d'onde unique. On retrouve cette idée en particulier dans les réseaux GMPLS et WDM. Par exemple, dans les réseaux SONET/WDM, on cherchera à grouper plusieurs OC-3 (155 Mb/s) dans un même OC-48 (2.5 Gb/s) qui sera transporté par une longueur d'onde. Cependant, à chaque insertion ou extraction de trafic sur une longueur d'onde, il faut placer dans le nœud du réseau un multiplexeur à insertion/extraction (ADM). De plus, il faut un ADM pour chaque longueur d'onde utilisée dans le nœud, ce qui représente un coût d'équipements important. Les objectifs de l'agrégation de trafic sont d'une part le partage efficace de la bande passante et d'autre part la réduction du coût des équipements de routage en réduisant la charge de travail des routeurs et par conséquent leur complexité.

L'étude de l'agrégation de trafic à plusieurs niveaux de granularité dans les réseaux de télécommunications a débuté dans les années 1970 pour le développement des réseaux de téléphonie fixe. Ce sujet a trouvé un nouveau développement dans les années 1990 avec le déploiement des réseaux SONET/WDM. Plusieurs travaux ont été développé dans l'objectif de minimiser le coût de déploiement du réseau tout en satisfaisant un ensemble donné de demandes de trafic. Ces travaux proposent des solutions optimales (modèle ILP) et sous-optimales (heuristique) mais ne considèrent que des topologies de réseau très simples tel que les réseaux en anneau et des demandes de trafic permanentes. Ce n'est que vers la fin du 20$^{\text{ème}}$ siècle que les réseaux maillés ont été considéré. Les demandes considérées sont toujours des demandes permanentes mais l'objectif de ces études est maintenant la minimisation du nombre de transpondeurs utilisés. Dans ce cas, les approches considérées sont soit optimales soit séquentielles. Plus récemment, l'agrégation du traffic a été abordée pour des demandes de trafic dynamiques dans le contexte de réseaux optiques maillés. Les approches proposées se sont limitées à des approches séquentielles où l'on cherche à minimiser le taux de rejet/blocage dans le réseau.

Comme on le verra dans la suite, cette thèse s'intéresse au problème d'agrégation de trafic dans des réseaux multi-couches optiques. Des approches optimales (modèle ILP), sous-optimales (heuristique) et séquentielles seront introduites. On utilisera pour cela un mélange de trafic dynamiques déterministes et dynamiques aléatoires.

La prise en compte de la qualité de transmission: Les générations de systèmes WDM à base de 2.5 et de 10 Gbit/s ont aujourd'hui atteint une certaine maturité. La prochaine étape est la mise sur le marché de systèmes à base de 40 Gbit/s et leur introduction dans les réseaux par les opérateurs. Cependant, transmettre à un débit plus élevé tout en garantissant la même qualité de transmission (à performances égales) impose de disposer d'un rapport signal à bruit accru (par exemple, augmenté de 6 dB pour un débit multiplié par quatre). Dans ce but, plusieurs approches ont été proposées. Une première approche consiste à accepter un rapport signal à bruit inférieur tout en ayant recours à des codes correcteurs d'erreurs. Ces codes sont utilisés dans le but de corriger les erreurs introduites à cause de la dégradation du signal. Une autre approche est de réduire l'espacement entre amplificateurs. Cependant cette approche n'est pas viable économiquement. Une troisième approche est d'augmenter la puissance du signal utile. Dans ce cas, le signal optique subit dans la fibre des altérations tant au niveau de sa composition que de sa structure et de sa puissance, qu'il faut s'efforcer de minimiser et de compenser. Parmi les effets induits dans la fibre, on distingue les effets linéaires dus à la dispersion chromatique ou la dispersion de polarisation de la fibre, et les effets non linéaires essentiellement induit par l'effet Kerr (dépendance de l'indice de réfraction de la fibre par rapport à l'intensité lumineuse qui la traverse).

Au milieu des années 1990, les travaux en matière de planification de réseau tout-optique ont commencé à prendre en compte les dégradations subies par le signal optique tout au long de son parcours. Ces travaux se sont focalisés principalement sur la dispersion chromatique (Chromatic Dispersion, CD), la dispersion modale de polarisation (Polarization Mode Dispersion, PMD) et l'émission spontanée amplifiée (Amplified Spontaneous Emission, ASE) qui a un retentissement direct sur le rapport signal optique à bruit (Optical Signal to Noise Ratio, OSNR). Des travaux récents prennent en compte les effets non-linéaires comme, l'auto-modulation de la phase (Self

Phase Modulation, SPM), la modulation de la phase croisée (Cross Phase Modulation, XPM), le mélange à quatre ondes (Four Waves Mixing, FWM) ou encore l'interférence entre canaux optiques (Cross Talk). Certaines de ces études modélisent l'évolution des paramètres physiques dans le but de proposer des algorithmes de routage qui minimisent le taux de rejet des connections tout en respectant la qualité du signal requise. D'autres autorisent de placer des régénérateurs dans certains nœuds du réseau pour régénérer des signaux optiques qui présentent de fortes dégradations. Ces algorithmes cherchent à déterminer l'emplacement optimal des régénérateurs pour garantir une bonne qualité de transmission, donc un bon rapport signal à bruit à la réception, tout en minimisant le nombre de régénérateurs et/ou les sites de régénération.

Les réseaux grilles: Par définition, une grille est un ensemble de ressources distribuées qui peuvent être vu par les utilisateurs comme une seule et unique ressource. Les réseaux grilles, ou « *Grid Networks* », trouvent leur origine dans la grille d'électricité. Vers le début du $20^{\text{ème}}$ siècle, la génération personnelle d'électricité était technologiquement possible et de nouveaux équipements utilisant la puissance électrique avaient déjà fait leur apparition. Paradoxalement, le fait que chaque utilisateur possède son propre générateur d'électricité était un obstacle majeur à la généralisation de son utilisation. La véritable révolution a nécessité la construction d'une grille électrique, c'est-à-dire la mise en place de réseaux de transmission et de distribution d'électricité. C'est l'accessibilité et la baisse des coûts qui ont finalement permis de généraliser l'exploitation industrielle de cette énergie.

Les réseaux grilles ont fait leur apparition dans le monde académique au milieu des années 1990. Ils permettent de mettre en partage de façon sécurisée les données et les programmes de multiples ordinateurs, qu'ils soient de bureau, personnels ou supercalculateurs. Ces ressources sont mises en réseau et partagées grâce à des solutions logicielles dédiées. Elles peuvent ainsi générer, à un instant donné, un système virtuel doté d'une puissance gigantesque de calcul et une capacité de stockage en rapport pour mener à bien des projets scientifiques ou techniques requérant une grande quantité de cycles de traitement ou l'accès à de gros volumes de données. Grâce à sa capacités à transporter des données à haut débit sur de longues distances avec des faibles délais de propagation, à sa flexibilité d'allocation de ressources, et à la simplicité de son interface, le réseau optique est de loin le réseau de transport idéal contribuant au succès des réseaux grilles. Á leur tour, les réseaux grilles contribuent au progrès scientifique et particulièrement dans le domaine médical où les besoins en calcul et en stockage sont considérables.

La protection et la restauration multi-couches: Un réseau de communication n'est pas constitué de la seule couche physique. Généralement, une entreprise fournit le support physique et une seconde, via un protocole de communications, exploite le support physique pour véhiculer des informations. D'autre part l'intégration de nouveaux services et applications, comme la voix, les données, la vidéo, et même le triple play, sur une même infrastructure de réseau a conduit à des empilements complexes comme IP/ATM/SDH/WDM. En conséquence, il serait intéressant de considérer les problèmes de routage et de protection dans les différentes couches simultanément.

Dans les réseaux multi-niveaux actuels, chaque niveau dispose de son propre mécanisme de protection. Or lorsque la protection est planifiée indépendamment pour chaque couche, elle induit souvent une baisse des performances du réseau suite à une réservation de bande passante de protection désorganisée (redondances, etc.). De plus des connexions disjointes à un niveau donné sont parfois routées sur un même lien à un niveau inférieur, ce qui n'est pas nécessairement pris en compte par le mécanisme de protection. Les conséquences peuvent être fâcheuses si ce lien tombe en panne, pouvant aller jusqu'à l'isolement complet d'une partie du réseau et l'impossibilité de satisfaire les requêtes des utilisateurs. D'autre part, concernant la restauration, il peut être intéressant de considérer des stratégies qui agissent simultanément au niveau des différentes couches pour rétablir une communication en partageant les moyens. Cependant, dans certains cas, on s'aperçoit qu'intervenir dans telle ou telle couche est plus économique et plus rapide. Cet aspect multi-couche se traduit généralement par un problème de dimensionnement couplé à un problème de routage.

Depuis l'introduction de MPLS, des mécanismes de protection unifiés où les différents niveaux coopèrent sont étudiés. Outre le type de mécanisme de protection à adopter parmi les multiples possibilités pour chaque niveau, il est aussi nécessaire de déterminer les interactions entre les niveaux et le rôle de chacun en cas de panne. Plusieurs stratégies existent dans la littérature:

a) La protection peut être assurée au niveau le plus proche de l'origine de la panne, si possible dans le niveau de la panne. Le routage est simple car le trafic est agrégé à un niveau de granularité proche de celui de la panne. Le nombre de connexions à rerouter est donc d'un ordre de grandeur gérable par le niveau qui met en oeuvre la solution de secours.

b) La protection peut être assurée au niveau le plus proche de l'origine du trafic, c'est-à-dire au plus haut niveau. La communication entre les niveaux est réduite puisque c'est toujours le niveau supérieur qui prend les pannes en charge. De plus, il est beaucoup plus facile de mettre en oeuvre une protection spécifique suivant les types de trafics et leurs degrés de priorité, puisqu'à ce niveau ils ne sont pas encore agrégés.

c) La protection peut être distribuée sur plusieurs niveaux, pour combiner les avantages des deux solutions précédentes. Lorsqu'aucune coordination entre les niveaux n'est prévue, il est possible que plusieurs d'entre eux mettent en oeuvre leur stratégie de protection simultanément ce qui peut conduire à une mauvaise utilisation des ressources, ou pire à un routage instable des connexions. Les mécanismes de protection des différents niveaux doivent donc être déclenchés séquentiellement. Il existe deux façons de procéder:

- Le mécanisme de protection du niveau le plus bas est déclenché en premier. S'il échoue à restaurer tout le trafic, le niveau supérieur prend la relève.
- Le mécanisme de protection du niveau le plus haut est déclenché en premier. S'il échoue à restaurer tout le trafic, le niveau inférieur prend la relève.

La radio sur fibre: Pour accompagner la demande accrue en mobilité et en interopérabilité dans les réseaux locaux, les opérateurs cherchent à étendre et à densifier la fourniture de débits pouvant atteindre 1 à 2 Gbit/s. Une solution repose sur l'utilisation d'un lien radio à des fréquences élevées pour desservir les usagers sur les derniers mètres. L'utilisation d'une liaison radio assure une flexibilité et facilité d'utilisation sans équivalent. La translation des porteuses radio vers les fréquences élevées permet de répondre à la demande en bande passante. Cependant elle entraîne aussi une hausse des coûts technologiques, amplifié par la réduction de la couverture et donc la nécessite de multiplier le nombre de stations de base. Le réseau d'accès radio devient donc un réseau pico-cellulaires où les problèmes d'interférence entre cellules et la gestion seront similaires à leurs équivalents dans le monde des réseaux mobiles.

La technologie « *Radio sur Fibre* » tente de répondre à la problématique de coût des systèmes pico-cellulaires en déplaçant la complexité technologique présente dans chacune des stations fixes locales vers une station centrale. Le lien entre les stations de base locales et la station centrale s'effectue par fibre optique. La station centrale se charge d'alimenter les stations locales en porteuses optiques modulées par un signal radio à très haute fréquence contenant les données. Elle élimine ainsi la présence d'oscillateurs et de modulateurs micro-ondes dans chacune des cellules. Outre cette simplification technologique des stations de base, la transmission sur fibre optique permet de s'affranchir des contraintes environnementales telles les parasites électriques ou les interférences radio. Les courants induits, les radiations ou les effets électromagnétiques ne peuvent pas générer de bruit parasitant le signal lumineux sur la fibre et donc ne peuvent absolument pas perturber le trafic. En plus, la fibre optique présente une résistance face à la foudre incomparable au regard des autres supports. Cela combiné au fait d'utiliser des coupleurs entièrement passifs assure une fiabilité sans équivalent, des coûts de maintenance extrêmement réduits et la possibilité de déployer le réseau d'accès dans des endroits où l'acheminement d'énergie, les contraintes en température, l'humidité ou les interférences électriques ne l'auraient pas permis. Ces réseaux optiques sont caractérisés aussi par une très faible atténuation dans la fibre, ce qui permet (de par la norme) de couvrir une superficie de 20 kilomètres de rayon sans aucun répéteur.

Les intérêts de la radio sur fibre sont donc l'unification du réseau d'antennes en un seul "macro-réseau" radio, la mutualisation des éléments intelligents, la flexibilité d'évolution et d'allocation dynamique grâce à un contrôle centralisé, la simplification de la conception des antennes déportées, et finalement le fait que ces points d'accès soient transparents vis-à-vis des systèmes radio utilisés. Les applications potentielles de la technologie radio sur fibre couvrent les réseaux de téléphonie mobile, les réseaux satellitaires, la distribution vidéo, la communication automobile, ainsi que l'échange de données informatiques.

Le codage réseau: Le codage réseau, ou « *Network Coding* », est un domaine de recherche émergeant, à la confluence des réseaux de communication, de la théorie de l'information, et de la théorie de codage. Dans les réseaux de transport traditionnels, les flux d'information sont routés de leur nœud source à leur nœud destination sans aucune modification de leur contenu au niveau des nœuds intermédiaires. Par opposition à cette méthode de routage, le codage réseau autorise aux nœuds intermédiaires du réseau à modifier l'information reçue. En d'autres termes, les nœuds du réseau peuvent envoyer des paquets qui sont des combinaisons linéaires des différents paquets reçus. Les premiers résultats obtenus ont montrés l'efficacité de cette approche à augmenter la capacité du réseau aussi bien pour des flux multi-cast que pour des flux uni-cast. Le codage réseau permet ainsi d'offrir des débits de transmission élevés avec des délais d'acheminement très courts. Il présente également des avantages en termes de robustesse lorsque les connexions réseau présentent des pertes de paquets. Le codage réseau a trouvé son application dans plusieurs domaines tel que la distribution des données dans les réseaux point-à-point et les réseaux ad-hoc sans fils.

Motivation

Après ce tour d'horizon sur les principaux sujets de recherche, passés et actuels, dans le domaine des réseaux optiques, nous arrivons au sujet de cette thèse proprement dit. Le but de cette thèse consiste à associer deux types d'optimisation: ingénierie de réseau et ingénierie de trafic. Pour cela, nous avons considéré deux types de demandes de trafic: soit pré-planifiées (SLD pour "Scheduled Lightpath Demand"), soit aléatoires (RLD pour "Random Lightpath Demand"). Une SLD ou une RLD requiert une capacité en nombre entier de canaux optiques. Les SLD sont des demandes de trafic déterministes connues longtemps avant leur occurrence. Le routage de telle demande peut donc se faire sur la base d'une optimisation du coût global du réseau. Cependant, les RLD sont des demandes de trafic aléatoires dont les caractéristiques ne sont connues qu'à l'instant de leur arrivée. Elles ne peuvent être traitées qu'une à une au fur et à mesure de leurs arrivées. Il s'agit par la suite de considérer le cas de réseaux multi-couches dans lesquels un brasseur électrique (SDH, MPLS, ATM) est couplé à un brasseur optique (OXC). Nous avons introduit le concept de SED ("Scheduled Electrical Demand") et de RED ("Random Electrical Demand"). Elles correspondent à des demandes de trafic pré-planifiées ou aléatoires dont la capacité requise est une fraction du canal optique. La principale innovation a consisté à proposer en parallèle aux règles de routage et d'affectation de longueurs d'onde une règle d'agrégation des demandes du type SED ou RED. Le problème global du routage et de l'agrégation a été formulé mathématiquement en utilisant l'algèbre matricielle. Cette formulation mathématique nous a permis de résoudre des instances du routage et de l'agrégation des SxD (SxD = SLD + SED) à l'aide de résolveurs conventionnels. En parallèle, nous avons proposé pour ce même problème une approche globale qui présente deux avantages: elle assure un temps de convergence nettement plus rapide et une optimisation accrue de l'utilisation des ressources du réseau. Quant aux demandes aléatoires du type RxD (RxD = RLD + RED), le routage et l'agrégation de telles demandes repose sur le concept de graphe auxiliaire auquel nous avons associé des règles originales de gestion dynamique de coût des arcs.

Cependant, l'apport principal de cette thèse est sans doute la possibilité de considérer simultanément des demandes déterministes et des demandes aléatoires. Nos travaux sont les seuls, à notre connaissance, qui traite de ce problème. Les études antérieures considéraient ou l'un ou l'autre des types de trafic. D'un côté, dans le cas des demandes déterministes, les approaches proposées sont des optimisations globales ou itératives qui cherchent à déterminer un routage optimal des demandes connaissant la corrélation spatio-temporelle entre elles. D'un autre côté, le routage des demandes aléatoires se fait de manière séquentielle en fonction des ressources disponibles du réseau à l'instant d'arrivée de ces demandes. Toutefois, lorsqu'on considère simultanément les deux types de trafic, le routage des demandes est plus compliqué. Pour illustrer cela, considérons un lien du réseau avec 2 longueurs d'onde seulement (Figure 1). Supposons aussi qu'à l'issue de la phase de routage des demandes déterministes, il a été planifié qu'une demande D_1 va utiliser ce lien à partir de l'instant t_2. Á ce stade, nous pouvons router davantage des demandes aléatoires sur les ressources libres du réseau. Par exemple, une demande aléatoire d_1 arrivant dans le réseau à l'instant t_0 ($t_0 < t_2$) peut utiliser ce lien du réseau indépendamment de sa durée d'activité. En effet, cela est possible vu que ce lien a au moins une longueur d'onde disponible à n'importe quel instant. Avec les algorithmes de routage existants, si une autre demande aléatoire d_2 arrive dans le réseau à l'instant t_1 ($t_0 < t_1 < t_2$), cette nouvelle demande peut être routée en ce servant de ce lien car ce dernier dispose d'une longueur d'onde libre à l'instant t_1. Cette approche est valide si au moins l'une des deux demandes d_1 et d_2 se termine avant l'instant t_2, mais elle résulte en un conflit si toutes les deux restent actives après l'instant t_2. Par convention, les demandes déterministes tels que D_1 sont prioritaires et par conséquent elles doivent bénéficier de leurs ressources garanties. Comme nous ne connaissons pas la durée des demandes aléatoires à l'instant de leur arrivée (mais une fois elles se terminent), nous devons supposer que ces demandes ont, dans le pire des cas, une durée de vie infinie. Dans le cas où le réseau ne permet pas le reroutage des demandes, la demande aléatoire d_2 doit être rejetée/bloquée à son instant d'arrivée t_1 en raison du manque de ressources libres à partir de l'instant t_2.

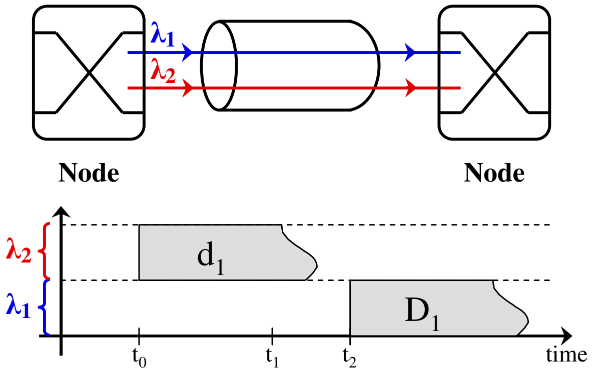


Fig. 1. Évaluation de la capacité libre d'un lien

Dans l'example précédent, on peut souligner deux idées importantes. La première consiste à évaluer les performances du réseau dans le cas où la date de fin des demandes aléatoires serait connue à l'instant d'arrivée de ces demandes. Une telle connaissance peut réduire le taux de blocage des demandes aléatoires en tirant un meilleur profit des ressources du réseau. Dans ce but, nous avons introduit un troisième type de trafic: les demandes semi-aléatoires (sRxD pour "semi-Random Demand"). Comme leurs homologues RxD, les demandes sRxD sont aussi des demandes aléatoires mais dont leur durée de vie est connue à leur instant d'arrivée. Elles seront routées séquentiellement au fur et à mesure de leur arrivée dans le réseau. Á leur tour, les demandes sRxD peuvent être classées en demandes sRLD ("semi-Random Lightpath Demand") et demandes sRED ("semi-Random Electrical Demand"). Les sRLD se caractérisent par un débit égal à celui d'un canal optique, alors que les sRED se caractérisent par une fraction de ce débit. Le deuxième point clé à soulever est l'impact d'un algorithme de reroutage sur les performances du réseau. Un tel algorithme permettrait de router les demandes RxD en utilisant les ressources libres du réseau à leur instant d'arrivée, et de les rerouter lorsqu'elles sont préemptées par les demandes déterministes considérées plus prioritaires. Cette approche a aussi pour effet de réduire le taux de blocage des demandes aléatoires.

Travaux de Recherche

La surveillance du trafic est un outil important pour un opérateur, elle lui permet d'approfondir sa connaissance du réseau. Le dimensionnement de réseau, l'ingénierie de trafic, et les applications offrant une qualité de service garantie dépendent fortement de cette connaissance des caractéristiques du trafic. Dans ce but, nous avons observé l'évolution de la charge du trafic dans des réseaux optiques de transport opérationnels. Les mesures effectuées montrent clairement que le trafic a des composantes périodiques journalières et hebdomadaires, aussi bien que des variations à plus long terme. A ces composantes se superposent des variations à une échelle beaucoup plus courte. Ces informations collectées ont servi au développement d'un modèle simple et précis du trafic dans les réseaux optiques de cœur. Dans le modèle proposé, la composante périodique du trafic est représentée par des demandes déterministes "*Scheduled Demands*" (SxD). Les fluctuations du trafic autour de cette composante périodique sont représentées par des demandes aléatoires. Selon que la date d'extinction des demandes aléatoires est donnée ou non à leur instant d'arrivée, nous en distinguons deux catégories: les "*semi-Random Demands*" (sRxD), et les "*Random Demands*" (RxD). Les sRxD englobent les demandes dont la date de fin est connue à leur instant d'arrivée alors que les RxD englobent les demandes dont la date de fin n'est connue qu'au moment où elles se terminent. Les sRxD peuvent être vue comme des demandes de trafic aléatoires plus prioritaires que les RxD et exigeant une meilleure qualité de service.

L'aspect périodique du trafic dans les réseaux observés dépend principalement de l'activité humaine. En conséquence, le débit de données déterministes entre deux nœuds du réseau sera directement lié à la densité de la population entourant les deux nœuds en question. En outre, le trafic est variable dans le temps. Ce caractère dynamique du trafic est pris en compte par un modèle représentant les heures de pointe. Pour rendre le modèle encore plus réaliste, nous avons pris en compte le décalage horaire qui peut avoir lieu entre les régions entourant deux nœuds différents. Á ce flux périodique vient se greffer des changements mineurs d'un jour à l'autre. Ces variations, représentées par les demandes aléatoires, affectent d'une manière plus ou moins homogène tous les flux de données. En conséquence, les demandes aléatoires arrivent dans le réseau suivant un processus poissonnien et seront réparties uniformément entre tous les couples de nœuds source-destination. Le débit requis par les demandes, qu'elles soient déterministes ou aléatoires, peut s'exprimer comme la somme d'un nombre entier de canaux WDM et d'une composante fractionnaire de la capacité d'un tel canal. Chaque composante entière unitaire fait référence à une "*Lightpath Demand*" ou xLD alors que la composante fractionnaire est associée à une "*Electrical Demand*" ou xED. En récapitulant les différents types de demandes évoquées, nous distinguons six catégories:

Scheduled Lightpath Demand (SLD): des demandes de débit unitaire (égal à la capacité d'une longueur d'onde) et dont les caractéristiques sont connues longtemps avant leur occurrence.

Scheduled Electrical Demand (SED): des demandes de débit fractionnaire (inférieur à la capacité d'une longueur d'onde) et dont les caractéristiques sont connues longtemps avant leur occurrence.

semi-Random Lightpath Demand (sRLD): des demandes de débit unitaire et dont la date de fin est connue à l'instant d'arrivée.

semi-Random Electrical Demand (sRED): des demandes de débit fractionnaire et dont la date de fin est connue à l'instant d'arrivée.

Random Lightpath Demand (RLD): des demandes de débit unitaire et dont la date de fin n'est pas connue au préalable.

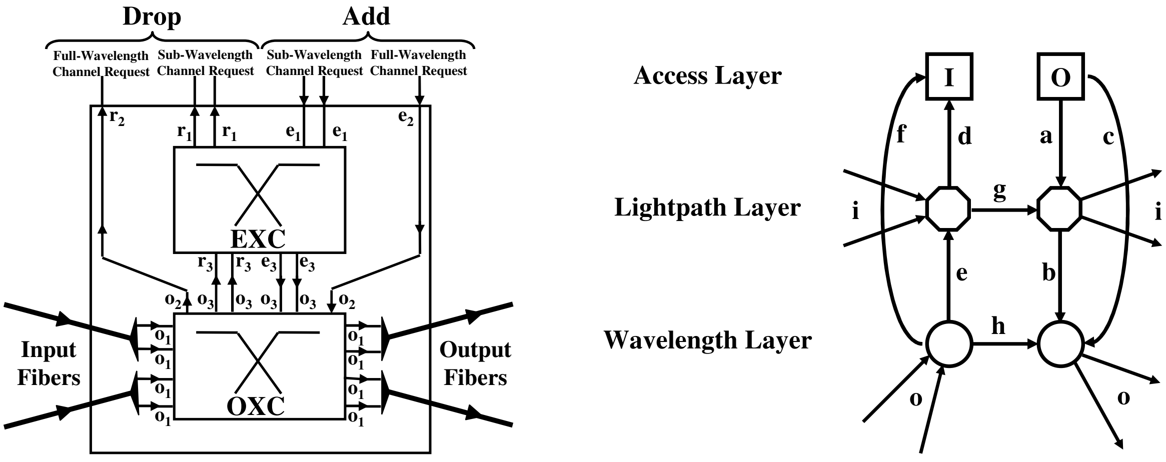
Random Electrical Demand (RED): des demandes débit fractionnaire et dont la date de fin n'est pas connue au préalable.

Tous ces types de demandes prennent en compte l'aspect dynamique du trafic et seront représentées par un quintuplet (nœud source, nœud destination, date de début, date de fin, débit souhaité). Á chaque ensemble de demandes, nous associons un débit moyen, un débit maximum, ainsi qu'une correlation temporelle entre les demandes.

La particularité des composantes xED réside dans le fait que leur débit est nettement inférieur à la capacité d'une longueur d'onde. Si on associait à chaque composante xED un chemin tout optique ou "*lightpath*" entre son nœud source et son nœud destination, une importante bande passante serait gaspillée. Par conséquent, il peut être avantageux de grouper sur le même lightpath plusieurs composantes xED simultanées dans le temps et ayant un trajet commun. Ainsi, un nombre réduit de lightpaths peut être associé aux demandes groupées. Cette technique consiste en un multiplexage au niveau électrique des composantes xED en question et est connue sous le nom d'agrégation ou "*Grooming*". Dans ce but, toutes les composantes xED en question doivent passer au niveau électrique au nœud amont du trajet commun où elles seront multiplexées ensemble. Une opération inverse a lieu au nœud aval du trajet commun dans le but de reconstituer les différentes composantes xED. On observe alors en ce nœud un nouveau passage au niveau électrique et un démultiplexage de toutes les composantes. Ces différents passages au niveau électrique requièrent un ensemble de ports optiques et de ports électriques additionnels au niveau de chaque nœud. Cela affecte considérablement le coût de déploiement du réseau.

Pour qu'un nœud du réseau puisse être capable de gérer des demandes de granularité entière (xLD) ainsi que des demandes de granularité fine (xED), il doit être composé d'un commutateur électrique (EXC) couplé à un commutateur optique (OXC). L'OXC est responsable d'ajouter, d'extraire et de commuter des xLD alors que l'EXC est responsable d'ajouter, d'extraire, d'agréger et de commuter des xED. Une telle architecture à double étages permet une utilisation efficace des ressources du nœud en assurant l'établissement et la reconfiguration de connexions point-à-point d'un bout à l'autre du réseau de transport. En cas de pannes de liens ou de nœuds, elle permet aussi de répondre aux besoins de restauration du réseau. Enfin, grâce à la fonction de commutation de données en mode circuit, cette architecture facilite l'ingénierie de trafic des couches clientes et permet à l'opérateur d'offrir des services de transport de données haut débit avec qualité de service garantie.

Chaque nœud du réseau peut être schématisé par un graphe auxiliaire équivalent ("*auxiliary graph*") inspiré de la littérature. Ce graphe est formé d'un ensemble de sommets connectés par des arcs dans le but d'implémenter les différentes fonctions du nœud. La schématisation d'un nœud diffère selon que ce dernier est capable ou non de conversion de longueurs d'onde. Mais en règle générale, le modèle équivalent d'un nœud est composé de 3 couches. La couche supérieure est la couche d'accès ("*Access Layer*"). C'est au niveau de cette couche que commencent et se terminent les demandes. La couche du milieu est la couche implémentant l'agrégation électrique ("*Grooming Layer*"). Finalement, la couche inférieure ("*Wavelength Layer*") est la couche responsable de la commutation des longueurs d'onde. Dans le cas où le nœud ne possède aucune capacité de conversion de longueurs d'onde, la dernière couche est séparée en w sous-couches où w est le nombre de longueurs d'onde par fibre. Chacune de ses sous-couches correspond à la capacité de commutation du nœud à une longueur d'onde donnée. Chaque couche/sous-couche est représentée par deux sommets: le premier agissant en port d'entrée à cette couche et le second en port de sortie. Des arcs intra-nodaux relient les différents sommets d'un même nœud. Ces arcs représentent les différents types de ports physiques du nœud. D'autres arcs, inter-nodaux, relient les sommets de deux nœuds différents appartenant à la même couche. De tels arcs au niveau de la couche de longueurs d'onde représentent la topologie physique du réseau, alors que les arcs au niveau de la couche d'agrégation représentent la topologie logique du réseau. À chaque arc, nous associons un couple $P(C_t, y_t)$ où C_t et y_t représentent respectivement l'évolution en fonction du temps de la capacité et de la charge de cet arc. La Figure 2 représente l'architecture d'un nœud ainsi que sa représentation en graphe équivalent dans le cas où il possèderait une capacité de conversion totale de longueurs d'onde. Nous soulignons le fait que le modèle du nœud et sa représentation en graphe auxiliaire sont complètement identiques. Dans la suite, nous nous servirons de ce modèle pour le dimensionnement de réseau et du graphe auxiliaire pour l'ingénierie de trafic.



(a) Architecture du nœud

(b) Graphe équivalent du nœud

Fig. 2. Architecture à double étages du nœud et sa representation en graphe équivalent

Les demandes de trafic déterministes (SxD = SLD + SED) sont des demandes planifiées dont on connaît les caractéristiques longtemps avant leur occurrence. Ce modèle de trafic dynamique et déterministe s'adapte parfaitement aux problèmes de dimensionnement de réseau. Ce problème est généralement formulé comme un problème d'optimisation. L'objectif est de satisfaire toutes les demandes tout en minimisant le coût de déploiement du réseau. La topologie physique du réseau étant fixée, le coût du réseau est exprimé en fonction du nombre des ports optiques et des ports électriques nécessaires en chaque nœud pour satisfaire toutes ces demandes. Pour cela, nous avons tout d'abord formulé mathématiquement le problème de routage et d'agrégation des demandes SxDs en utilisant l'algèbre matricielle. Les matrices définies nous permettent de calculer le nombre des différents types de ports physiques utilisés et permettent de distinguer entre les ports en émission et ceux en réception. Elles permettent aussi de distinguer entre les ports utilisés pour l'ajout et le retrait du trafic, ceux utilisés pour l'agrégation du trafic, et ceux utilisés pour la commutation optique. Basé sur cette formulation, nous avons résolu le problème de dimensionnement de réseau par deux approches différentes. La première approche est une solution exacte utilisant la programmation linéaire (ILP: Integer Linear Programming). Dans cette approche, nous avons considéré trois cas particuliers:

1) Dans le premier cas, nous considérons uniquement des demandes du type SLD. Pour chaque demande, nous calculons en avance un ensemble de k-plus courts chemins. Le rôle de l'algorithme d'optimisation est alors de déterminer sur quel chemin chaque demande sera routée afin de minimiser le coût total du réseau.

2) Dans le deuxième cas, nous considérons toujours des demandes du type SLD, mais nous ne pré-calculons aucun chemin de routage. Dans ce cas, en se servant de la propriété de conservation de flux de données dans les réseaux, l'algorithme d'optimisation doit déterminer pour chaque demande un aboutement de liens entre son nœud source et son nœud destination; le but global étant toujours de minimiser le coût total du réseau.

3) Dans le dernier cas, nous considérons uniquement des demandes du type SED avec k-plus courts chemins pré-calculés pour chaque demande. L'algorithme d'optimisation choisit alors le chemin physique emprunté par chaque demande et détermine son routage au niveau de la topologie logique.

Cette approche, basée sur un modèle ILP, nous a permis d'obtenir la solution optimale de routage et d'agrégation des SxDs pour des réseaux de taille réduite et un nombre limité de demandes. Cependant, le temps de calcul explose pour des réseaux de taille réelle et/ou un nombre important de demandes. Dans ce dernier cas, une approche heuristique a été adoptée. L'optimisation du routage des SxD est effectuée a l'aide d'un algorithme Recuit Simulé exploitant la corrélation espace-temps entre les demandes, alors que l'optimisation de l'agrégation des SED est effectuée à l'aide d'un algorithme Greedy itératif.

Le Recuit Simulé proposé construit une solution initiale en attribuant à chaque demande le plus court chemin entre son nœud source et son nœud destination. Dans le cas du routage des demandes SED, cette solution ne prend pas en compte l'agrégation des demandes au niveau électrique. En changeant à chaque itération le chemin de routage d'un nombre limité de demandes, l'algorithme Recuit Simulé parcourt l'espace des solutions à la recherche de la meilleure solution possible. L'algorithme se termine lorsqu'aucune amélioration de la solution courante n'est observée à l'issue d'un certain nombre de modifications successives. Quant à l'algorithme Greedy itératif, il part d'une solution de routage donnée par l'algorithme Recuit Simulé. Sans modifier le chemin de routage, il groupe les demandes SED par paires dans des sous-ensembles caractérisés par une longueur fixe du chemin commun. Il parcourt par la suite les sous-ensembles de demandes du plus long trajet commun ou plus court, et il applique un certain nombre d'opérations à l'intérieur de chacun d'eux. Ces opérations consistent à essayer d'agréger ensemble une paire de demandes choisie aléatoirement dans le sous-ensemble en question. L'algorithme Greedy proposé a montré sa capacité à achever des solutions d'agrégation dont le coût est inférieur au coût obtenu par d'autres algorithmes trouvés dans la littérature et en un temps nettement plus court.

En générale, nous pouvons conclure que l'utilisation de chemins alternatifs pour router les demandes SxD permet une meilleure utilisation des ressources du réseau. Plus le nombre de chemins alternatifs est grand, moins est le nombre de ports requis, meilleur est le gain obtenu par réutilisation des ressources. Toutefois, utiliser 4 chemins alternatifs semble un bon compromis entre le gain obtenu et le temps de calcul. Pour un nombre de chemins alternatifs fixés, le gain le plus élevé est obtenu par le modèle ILP au prix d'un très long temps de calcul. En outre, notre approche heuristique permet d'obtenir des solutions sous optimales assez proches de celles obtenues par le modèle ILP en un temps de calcul significativement réduit par rapport à ce dernier. En moyenne, en modifiant uniquement le chemin de routage des demandes, l'algorithme Recuit Simulé permet une réduction du coût global du réseau d'environ 4%. De même, en optimisant l'agrégation des demandes de granularités fines, l'algorithme Greedy itératif permet une réduction du coût global du réseau d'environ 15%. Finalement, en permettant aux demandes SLD et SED de partager des ressources communes, nous pouvons achever une réduction du coût global du réseau d'environ 1% supplémentaire.

À l'issue de cette étape de dimensionnement, le routage des demandes SxD est réalisé, et l'emplacement des ports optiques et des ports électriques est déterminé. Cependant, la grande variété des applications utilisées de nos jours fait que le trafic n'est pas tout à fait périodique. En effet, des variations inattendues de trafic peuvent survenir et ainsi mener à la saturation des ressources du réseau. Afin de réduire ce risque de blocage, le réseau peut être surdimensionné au prix d'un investissement substantiel dans son infrastructure. Ces ressources additionnelles serviront pour router les demandes aléatoires.

Concernant les demandes de trafic aléatoires, leurs caractéristiques ne sont connues qu'à leur instant d'arrivée. En conséquence, elles sont traitées une à une à l'instant de leur arrivée et sont routées par le plan de contrôle en fonction de l'état du réseau à cet instant. Le choix du chemin de routage d'une demande aléatoire se fait au niveau du nœud source à partir d'information sur la charge du réseau disponible en ce nœud. Comme le routage des demandes SxD est confié au plan de gestion, tous les nœuds du réseau sont informés des chemins affectés à ces demandes. Outre les informations sur le routage des demandes SxD, chaque nœud doit disposer aussi d'information sur le taux d'utilisation de tous les liens du réseau. De telles informations sont diffusées régulièrement sur le réseau à l'aide d'une signalisation distribuée compatible avec le protocole " *Generalized Multi-Protocol Label Switching*" (GMPLS). À l'aide de ces informations, le nœud source établit sa propre vision de l'état/la charge du réseau à chaque fois qu'une nouvelle demande de connexion arrive dans le réseau. Cette vision diffère selon que la demande à router est une demande sRxD ou une demande RxD. En effet, l'état d'un lien du réseau est déterminé, par exemple, à partir de son utilisation maximale durant la durée d'activité de la demande à router. Pour les demandes du type sRxD, cette durée d'activité est connue, alors que pour les demandes RxD, elle est inconnue, et par la suite supposée infinie. Le but de cette approche est de ne router les demandes aléatoires que sur la partie du réseau qui reste vacante durant toute leur durée d'activité. Cette condition est indispensable dans les réseaux ne disposant pas de fonction de reroutage.

Pour router les demandes aléatoires, nous nous sommes servis de la représentation en graphe équivalent d'un nœud du réseau dans le but d'implémenter les différentes fonctions du nœud. En ce basant sur cette représentation, nous avons construit une représentation en graphe de tout le réseau et nous avons développé de nouvelles métriques pour les arcs de ce graphe prenant en compte l'état actuel du réseau. En utilisant l'algorithme de Dijkstra, nous choisissons le chemin approprié dans la topologie logique capable de satisfaire la nouvelle demande ainsi que son chemin correspondant dans la topologie physique. La possibilité de modifier dynamiquement les métriques associées aux différents arcs du réseau nous permet à la fois de router et d'agréger efficacement les demandes RxD et sRxD. L'objectif de ces métriques est de minimiser les ressources sollicitées par chaque demande. Cet objectif vise implicitement à minimiser la probabilité de blocage des demandes futures. Les simulations ont permis de souligner l'intérêt de pouvoir spontanément agréger les nouvelles demandes aléatoires, l'efficacité des nouvelles métriques proposées, ainsi que la capacité de l'algorithme de gérer simultanément les informations concernant les demandes déterministes et celles concernant les demandes aléatoires. Nous avons aussi évalué le taux de rejet des demandes RxD et celui des demandes sRxD. Dans ce contexte, nous avons souligné l'impact de la connaissance de la durée d'activité des demandes aléatoires sur les performances du réseau. En effet, pour le même scénario de trafic, l'utilisation globale des ressources du réseau par les demandes RxD est inefficace en comparaison avec celle des demandes sRxD.

Dans le but d'améliorer les performances du réseau et de réduire la probabilité de blocage, deux techniques de reroutage ont été abordées. La première méthode de reroutage consiste à rerouter certaines demandes existantes à fin de satisfaire une nouvelle demande de connexion. Dans ce cas, l'algorithme de reroutage doit déterminer un chemin pour la nouvelle demande ainsi que des chemins alternatifs pour les demandes à rerouter. Dans un réseau implémentant l'agrégation électrique, la couche optique et la couche électrique sont impliquées dans le problème de reroutage. Ainsi, les techniques de reroutage peuvent être appliquées à deux niveaux différents, à savoir reroutage au niveau optique physique et reroutage au niveau électrique logique. L'algorithme proposé est dérivé d'un autre algorithme déjà existant et connu sous le nom de "*Move to Vacant-Wavelength Retuning*" (MTV-WR). Ce dernier algorithme consiste à rerouter certaines demandes existantes sans changer leurs chemins physiques mais en leur affectant de nouvelles longueurs d'onde. L'intérêt de cette approche est qu'elle cherche à minimiser la période où le trafic est perturbé. Cependant, elle est sans aucun avantage dans le cas où les nœuds du réseau sont capables de conversion de longueurs d'onde. En plus, elle s'avère impuissante dans le cas de réseau représenté par un graphe utilisant des arcs unidirectionnels. Le fait que notre représentation de nœud en graphe équivalent est basée sur des arcs unidirectionnels et que les demandes considérées ne sont pas symétriques nous a motivé à étendre l'algorithme existant au cas de représentation en graphe unidirectionnel. Le nouveau algorithme est en plus capable de changer le chemin de routage associé à une demande. Ces modifications consistent principalement en l'ajout de nouveaux types de "*crossover edges*". "*Path Adjusting Rerouting*" (PAR) est le nom que nous avons attribué à l'algorithme résultant. Comme les demandes du type SxD sont considérées prioritaires, l'algorithme proposé n'affecte que les demandes du type RxD ou sRxD, et se déroule en deux phases:

Phase 1: À l'arrivée d'une nouvelle demande aléatoire, nous cherchons tout d'abord à satisfaire cette demande sans avoir recours au reroutage. Cette phase se déroule comme décrite précédemment en représentant le réseau par un graphe basé sur la représentation en graphe auxiliaire de chaque nœud et en pondérant chaque arc par un poids adéquat déterminé à partir de la charge du réseau. En appliquant l'algorithme de Dijkstra, nous pouvons déterminer pour la nouvelle demande un chemin de routage à coût fini. Si cette tentative échoue (le coût du chemin est infini), on passe à la phase suivante.

Phase 2: Dans cette phase, nous cherchons tout d'abord les points d'étranglement du réseau. Ces points sont en général des arcs auxquels nous avons associés des poids infinis. Pour chaque circuit existant qui traverse un tel lien, nous cherchons à le rerouter sur d'autres chemins alternatifs. Une fois que nous avons identifié les connections qui peuvent être reroutées, nous modifions le graphe du réseau en ajoutant des arcs spéciaux (les "*crossover edges*"). Ces arcs spéciaux connectent les différents sommets d'une connection reroutable et seront pondérés par le prix de reroutage de cette demande. Nous appliquons une dernière fois l'algorithme de Dijkstra sur le graphe résultant. Si le chemin retourné par cet algorithme a un coût fini, la nouvelle demande peut être routée après le reroutage de certaines demandes existantes, sinon elle sera bloquée.

Une autre méthode de reroutage a été proposée et s'avère être très utile quand nous routons des demandes avec des durées de vie imprévisibles comme c'est le cas des RxDs. Cette stratégie de reroutage "*Time Limited Ressource Reservation*" (TLRR) est basée sur le principe de réservation de ressources pendant une période de temps limitée Δ pour fournir une sorte de services garantis aux RxDs. Cette approche consiste à supposer que la durée d'activité des demandes RxD est égale à Δ et elles seront routées par la suite comme des demandes sRxD dont on connaît la durée d'activité. Si la demande RxD se termine avant la fin de la période Δ, les ressources du réseau seront libérées, sinon elle continuera à utiliser les mêmes ressources jusqu'à sa fin ou jusqu'à ce qu'elle soit préemptée par une demande SxD. Dans ce denier cas, nous chercherons à rerouter la demande RxD sur un nouveau chemin pendant une nouvelle période Δ. En choisissant convenablement la valeur de la fenêtre de réservation, il est possible d'atteindre des taux de rejet pour les RxDs comparables à ceux obtenus pour les sRxDs. L'efficacité du reroutage nécessite d'examiner une nouvelle métrique autre que la probabilité de rejet. En effet, durant la procédure de reroutage, un ensemble de messages de signalisation doit être échangé entre les nœuds du réseau. Plus le nombre d'opérations de reroutage est faible, plus courte sera la période de perturbation du réseau, et meilleure sera la qualité de service QoS. Le choix du paramètre Δ est donc un compromis entre la sporadicité des demandes et le nombre d'opérations de reroutage.

Théoriquement, la meilleure solution de routage des RxD est obtenue lorsque nous connaissons exactement leur durée d'activité. Dans ce cas, le taux de rejet des RxD est égal à celui de leur homologue sRxD. Cependant, cette connaissance de la durée d'activité des RxDs est impossible du fait de la nature de ces demandes. Donc le rôle de la fenêtre temporelle Δ est d'estimer une valeur approchée de cette durée. Dans ce but, nous avons proposé une version dérivée de l'algorithme TLRR qui sera connue sous le nom de "*Adaptive Time Limited Ressource Reservation*". Dans cette version de l'algorithme, la valeur du paramètre Δ n'est pas fixe mais elle varie au cours de l'algorithme. Dans ce cas, Δ est une estimation de la durée d'activité des RxD et est calculée comme étant la moyenne des durées d'activité des demandes RxD déjà routées dans le réseau. Cette valeur moyenne sera mise à jour chaque fois qu'une demande RxD se termine. Les résultats de simulation montrent que l'algorithme de reroutage TLRR permet d'obtenir des taux de rejet légèrement inférieurs au cas où les demandes RxD seront routées en se basant uniquement sur la connaissance de l'état du réseau à leur instant d'arrivée ($\Delta = 0$) avec un nombre d'opérations de reroutage significativement réduit. En plus, l'algorithme ATLRR permet de s'adapter dynamiquement à la sporadicité du trafic et d'assurer par la suite une bonne estimation du paramètre Δ.

Conclusion et Perspectives

Les réseaux de transport haut-débit doivent être capable de pouvoir transporter simultanément différents types de trafic; permanent, planifié, sporadique, et même aléatoire. En général, de tels réseaux sont déployés en utilisant une infrastructure optique offrant des canaux de transmission permettant des débits de plusieurs Gigabits par seconde. Cependant la plupart des applications actuellement existantes exigent une bande passante inférieure à celle-ci, d'où la nécessité de pouvoir agréger plusieurs demandes sur le même canal optique. Il est donc important que ces réseaux soient conçus de manière optimale en termes de coût tout en permettant le transport efficace de ces types de trafic.

Cette thèse porte sur la conception et l'analyse de réseau optique permettant l'ingénierie de trafic et englobant les fonctions d'agrégation et de reroutage. L'originalité de cette thèse est de traiter à la fois des demandes planifiées et des demandes aléatoires. Elle associe par la suite deux types d'optimisation: planification de réseau et ingénierie de trafic. Le problème traité n'est pas évident sachant que les ressources libres du réseau sont fluctuantes dans le temps. En effet, selon les instants d'établissement et de libération des chemins pré-calculés pour les demandes planifiées, les ressources du réseau seront occupées pendant certaines périodes de temps et demeureront libres pendant d'autres. En fonction de la date d'arrivée de la nouvelle demande aléatoire et de sa période d'activité, ces ressources peuvent être utilisées ou non pour satisfaire la nouvelle demande. Dans ce but et dans un premier temps, nous avons développé des approches exactes et d'autres heuristiques pour le problème de dimensionnement de réseau multi-granulaire dans le cas de demandes déterministes. Dans un second temps, nous avons examiné le problème de routage et d'agrégation dynamique des demandes aléatoires sous réserve de la disponibilité de ressource libre dans le réseau. Plusieurs algorithmes sont développés pour faciliter ce problème de provisionnement. Enfin, des techniques de reroutage de trafic sont adressées. Nous avons étudié différentes stratégies de reroutage et leur application en vue d'améliorer l'efficacité du réseau quand ce dernier transporte à la fois des demandes de trafic planifiées et d'autres aléatoires.

Certains des aspects originaux développés dans cette thèse ouvrent de nouvelles pistes de recherche. Toutes les techniques de routage et d'agrégation de demandes déterministes et aléatoires méritent d'être étendues et validées dans le cadre de réseau optique avec des contraintes sur la qualité de la transmission (QoT). Dans ce cas, les algorithmes de routage des demandes doivent prendre en compte la dégradation du signal optique due à la traversée des divers équipements installés le long du chemin optique. Cela a pour but de minimiser le taux de rejet des demandes dû à une mauvaise qualité du signal à la reception. Il serait aussi intéressant d'approfondir l'étude sur le mécanisme de reroutage TLRR avec des services garantis pendant une fenêtre temporelle Δ. Certes, il existe une relation entre la valeur optimale de Δ d'un côté et les caractéristiques des demandes de trafic déterministes et aléatoires de l'autre. Cette relation reste à déterminer.

Paris,
Mai 2007

Elias A. DOUMITH

Abstract

Understanding Internet backbone traffic is crucial for Internet architecture evolution. Capacity planning, traffic engineering, and meeting service level agreements are strongly dependent on the knowledge of the traffic evolution. By observing aggregate link statistics collected on a large backbone network, we have collected the data necessary to develop an accurate model of backbone traffic. From these statistics, it is clear that the traffic has both daily and weekly periodic components, as well as longer-term variations. Shorter time-scale stochastic variations are superimposed on top of these components. Given these characteristics, we have developed a simple, but realistic model for backbone traffic. The periodic traffic is represented by a set of dynamic but deterministic traffic requests referred to as *Scheduled Demands* (SxDs) while the stochastic traffic is represented by a set of dynamic and completely random traffic requests referred to as *Random Demands* (RxDs). An SxD is a preplanned demand with pre-determined date of arrival, life duration, and capacity. Optical Virtual Private Network (OVPN) is an example of such demands. Conversely, an RxD corresponds to a connection request that arrives randomly and, if accepted, holds the network for an arbitrary period of time. Best-effort traffic is an example of such demands. As this thesis is partially dedicated to the impact of traffic predictability on network efficiency, we have been driven to introduce a third class of traffic requests referred to as *semi-Random Demands* (sRxDs). An sRxD can be viewed as a class of random traffic requests that requires higher priority services than an RxD. An sRxD corresponds to a connection request that arrives randomly in the network but asks for network resources for a given period of time. All three classes of traffic requests are said to be multi-granular demands and do not necessarily require the whole capacity of an optical channel. To allow a better link optimization, low-speed traffic streams generated in the electrical domain should be merged together onto high-capacity lightpaths.

The aggregation of these electrical traffic flows onto optical lightpaths is known as the traffic grooming problem. Traffic grooming enables to increase network capacity utilization, minimize network cost, and optimize potential network revenue. The traffic grooming problem can be formulated as follows. Given a network configuration (including physical topology, number of transceivers at each node, number of wavelengths on each fiber-link, and the capacity of each wavelength) and given a set of connection requests with different bandwidth granularities, we need to determine how to set-up lightpaths in order to satisfy these connection requests. Because of the sub-wavelength granularity of the connection requests, one or more connections can be multiplexed on the same lightpath. The set of connection requests may be given either in advance (*e.g.,* SxDs), or sequentially (*e.g.,* RxDs and sRxDs). Traffic grooming with dynamic scheduled traffic demands is a dual optimization problem. In a non-blocking scenario where the network has enough resources to carry all the connection requests

(Network Planning), the objective is to minimize the network cost. The network cost can be modeled as the total number of Optical Cross-Connect/Electrical Cross-Connect (OXC/EXC) ports used in a Wavelength Division Multiplexing (WDM) mesh network, and/or the total number of wavelengths used in the fiber-links of this network. In a blocking scenario where some connections can be blocked due to resource limitations (Traffic Engineering), the objective is to maximize the network throughput. With dynamic random traffic demands, the objective is to minimize the network resources required for each request. This objective implicitly attempts to minimize the blocking probability for future requests. In our approach, we have first developed different optimization strategies in order to dimension a network able to carry a given set of SxDs. For this purpose, we exploit the space-time correlation between the traffic demands. As a second step, once the network's nodes are dimensioned, the network can be over-dimensioned to carry additional random requests. To the best of our knowledge, this is the first work which considers simultaneously preplanned demands and random demands. The compound problem is not trivial knowing that the free resources of the network are fluctuating. In fact, according to the scheduled set-up and tear-down of the lightpaths already computed to satisfy the SxDs, some of the network resources are occupied during some period of time and remain free during other. Depending on the arrival date of a new random demand and on its holding time, these network resources could be used, or not, to satisfy this new request. Using an existing auxiliary graph model for the network nodes, we have developed new metrics for the graph edges that enable us to efficiently route and groom both RxDs and sRxDs. Then, by means of the well-known Dijkstra algorithm, we choose the appropriate lightpath in the logical topology and its corresponding path in the physical topology. In this context, we have evaluated the rejection ratio of RxDs and sRxDs. We have highlighted that, when routing RxDs, the overall network resources utilization is inefficient compared to the case where sRxDs are considered under the same traffic scenario.

In order to enhance the network performance and to reduce the blocking probability, rerouting techniques are addressed. Many rerouting schemes can be developed. One scheme may determine which existing lightpaths should be rerouted in order to accommodate a new connection request. When some connections need to be rerouted, the rerouting algorithm should determine the new paths to be assigned to these rerouted connections. In a traffic grooming capable network, the optical layer and the electrical layer are involved in the routing problem. Thus, the rerouting techniques can be applied at two different levels, namely *Rerouting at the Physical Optical Level* (RPOL) and *Rerouting at the Logical Electrical Level* (RLEL). Another rerouting scheme can be very useful when we route demands with unpredictable life duration such as RxDs. This rerouting strategy is based on a *Time Limited Resource Reservation* (TLRR) to provide a form of guaranteed services to RxDs. By choosing the suitable value of the transient reservation window Δ, it is possible to achieve rejection ratios for RxDs comparable to those obtained for sRxDs. The efficiency of rerouting needs to investigate another metric than the rejection probability. Indeed, in order to proceed to rerouting, a set of signaling messages must be exchanged between the network nodes. The lesser the number of rerouting operations, the shorter the disruption period, the better the offered Quality of Service (QoS). The choice of the parameter Δ is a trade-off between the burstiness of the requests to be routed and the number of rerouting operations.

Paris,
May 2007

Elias A. DOUMITH

Contents

Part I Introduction

Part II Traffic Grooming and Rerouting in Multi-layer WDM Network

Part III End Matter

List of Figures

List of Tables

List of Algorithms

List of Examples

Part I

Introduction

1

Introduction

Abstract: *High-performance transport networks are expected to support applications with various types of traffic, e.g., permanent, scheduled, bursty, and noisy traffic flows. Since high-performance networks usually employ optical network infrastructures, and since most applications require sub-wavelength bandwidth, several streams are usually groomed on the same wavelength. It is therefore important that such networks are designed in an optimal way in terms of cost while efficiently supporting these types of traffic. This thesis deals with the design and analysis of optical networks allowing for traffic engineering including grooming and rerouting functionalities. Both deterministic and random traffic scenarios are considered. As a first step, optimal as well as accurate heuristic approaches are developed for network design and operation under deterministic traffic conditions. As a second step, under random traffic conditions, the dynamic routing and grooming problem is considered subject to the availability of free network resources. Several algorithms are developed to facilitate this provisioning problem. At last, rerouting techniques are addressed. We investigate different rerouting strategies and their implementation in order to enhance network efficiency under specific traffic scenarios.*

1.1 Introduction

In current transport networks, Wavelength Division Multiplexing (WDM) divides the large bandwidth available in optical fibers into multiple parallel channels operating at different wavelengths and at data rates of around 2.5 Gbps (OC-48) or 10 Gbps (OC-192). According to the current state of the technology, a significant gap exists between the huge transmission capacity of WDM fibers and the electronic switching capacity. Consequently, these networks are limited more by the electronic switches and routers at the nodes than by the link transmission bandwidth. Wavelength-routing technology using Optical Add/Drop Multiplexers (OADMs) and Optical Cross-Connects (OXCs) enables wavelength reuse and helps to overcome the electronic bottleneck. OADMs allow wavelengths to be selectively dropped at the nodes or optically passed through the node unaffected. In addition, OXCs enable space-switching of wavelengths from one port to another and help in establishing circuit-switched connections called lightpaths between the nodes. This enables optical pass-through at the WDM layer and eliminates the need to electronically process all the traffic at the nodes.

In WDM core networks where the optical channel has transmission capacity of several gigabits per second (*e.g.,* OC-48, OC-192 or OC-768 in the future), the full wavelength capacity corresponds to a coarse grain granularity whereas the traffic connections may require lower data rates. These low-rate traffic connections can vary in range from a few megabits up to the full wavelength channel capacity. In addition, for networks of practical size, the number of available wavelengths is still lower by a few orders of magnitude than the number of source to destination connections that need to be established. The solution for this multi-granularity problem lies in efficiently grooming data flows onto wavelengths. Traffic grooming in WDM networks can be defined as the act of multiplexing, demultiplexing, and switching low rate traffic streams onto high capacity lightpaths. Switching can be performed in space, time, and wavelength so that a traffic stream occupying time-slots on a wavelength on a fiber can be switched to different time-slots on a different wavelength on another fiber. Traffic grooming improves the amount of optical pass-through and wavelength utilization in the network at the cost of increased cross-connect complexity. In addition, wavelength conversion may be provided in the network in order to lighten the wavelength continuity constraint. Currently, all-optical wavelength conversion and all-optical time-slot interchange devices are still not commercially available, and it is more attractive to use electronic techniques to incorporate these features into the network. In the future, it may be possible to perform all-optical traffic grooming using Optical Packet Switches (OPS). Such all-optical traffic grooming networks may prove to be more cost-effective and manageable than their electronic counterparts.

Apart from traffic grooming and wavelength conversion, there is yet another way, called rerouting, to improve resources utilization in WDM networks. Rerouting techniques may be used when a new connection request arrives and cannot be accepted due to the lack of network resources. Rerouting aims at rearranging a certain number of existing lightpaths to free some network resources for the incoming request. A rerouting scheme can change the wavelength as well as the original path (called basepath in the following) of the existing lightpaths to satisfy a new connection. However, when wavelength conversion is deployed in each node of the network, wavelength retuning is inefficient. Thus, the basepath of some existing lightpaths needs to be changed in order to accept a new request. A comprehensive survey of rerouting can be found in [J54].

Rerouting can be very expensive in terms of service disruption and network signalling overhead. Indeed, depending on the dynamics of the signaling messages necessary to proceed to rerouting, a certain bufferization of the rerouted connection is necessary at the ingress node. The faster this signaling, the shorter the bufferization, the better the offered Quality of Service (QoS). Rerouting runs in two phases. The first phase, known as the rerouting algorithm in the literature, aims at determining existing lightpaths or connection demands to be rerouted in order to accommodate a new request. The second phase, also called the configuration migration technique, defines the sequence of steps executed in the network to migrate the rerouted lightpaths or connections to their new paths. The first phase should be simple (*i.e.,* run in polynomial time) and should minimize the weighted number of rerouted lightpaths in the network while satisfying a new connection request. The second phase is a key function of the control plane and largely determines the rerouting disruption time which should be minimal [J55].

1.2 Traffic Grooming

Traffic grooming in networks refers to grouping low rate traffic streams into higher speed streams, which can be considered as single entities. Historically, grooming has been investigated with special attention in the context of packet-switched data networks. For instance, the concept of Virtual Path (VP) in Asynchronous Transfer Mode (ATM) networks has been motivated by the necessity to group individual Virtual Circuits (VCs) in order to prevent large routing tables at the switching nodes. With the increase of Internet traffic, the concept of VPs has been generalized in Multi-Protocol Label Switching (MPLS) networks. Indeed, in MPLS networks, it is possible to define multiple levels of aggregation thanks to the label stacking procedure.

New forms of grooming have been introduced with the emergence of all-optical networks where it is possible, for example, to aggregate lightpaths into wavebands in the optical domain. Such an all-optical aggregation does not only reduce the size of the routing tables but also the cost of the switching nodes. This cost strongly depends on the number of input/output ports.

As WDM technology advances, the capacity of wavelength channel continues to increase. However, the bandwidth requirement of a typical connection usually corresponds to a small fraction of the WDM channel capacity. In order to efficiently use the bandwidth provided by an optical channel, hybrid grooming has been introduced. Hybrid grooming consists in merging multiple electrical flows (*e.g.,* VCs and VPs from ATM, and Label Switched Path (LSP) from MPLS) into optical lightpaths. The hybrid grooming scheme cumulates the benefits of electrical and all-optical grooming which lead to a better utilization of the optical channels. In this thesis, we focus exclusively on hybrid grooming. Through the remaining of this dissertation, we refer by "grooming" to the "hybrid grooming" scheme in WDM networks.

Traffic grooming started to become a relevant research topic at the end of the 1990's. The early research work on the problem of traffic grooming has been carried out in the context of Synchronous Optical NETwork/Wavelength Division Multiplexing (SONET/WDM) ring networks. In traditional SONET network, an Add Drop Multiplexer (ADM) is needed for each wavelength at every node to perform an add/drop operation on that particular wavelength. With the progress of WDM technology, over a hundred wavelengths can now be supported simultaneously by a single fiber. Consequently, it is too costly to put the same amount of ADMs at every network node since a lot of traffic is only bypassing an intermediate node.

Compared to the wavelength channel resources, ADMs form the dominant cost in a SONET/WDM ring networks. Several optimal and near-optimal algorithms have been proposed in the literature to solve the traffic grooming and wavelength assignment problem in ring networks for single-hop and multi-hop connections. The objective of such algorithms is to minimize the number of required wavelengths and ADMs [J56, J57].

It has been proven in [J56, J58] that the general traffic grooming problem in WDM ring networks is *NP*-Complete. Wang *et al.* [J57] formulate the grooming optimization problem as an Integer Linear Program (ILP) for ring networks considering static traffic. The limitation of the ILP approach is that the number of variables and the number of equations increase explosively as the size of the network increases. In order to handle the problem in large networks, several heuristic approaches have been proposed. In [J58, J59], a lower bound analysis is provided for different traffic criteria (uniform and non-uniform) and different network models (unidirectional and bi-directional rings). These lower bound results characterize the performance of traffic grooming heuristic algorithms. In most heuristic approaches, the traffic grooming problem is considered as twofold. As the first step, the traffic demands are assigned to circles. As the second step, a traffic grooming algorithm is employed to reorganize the circles or connections on wavelengths. Greedy approaches and Simulated Annealing (SA) approaches are used in these heuristic algorithms [J56, J57, J58, J59].

Later on, traffic grooming in WDM mesh networks started to get more attention [C1, J60]. In the context of mesh networks, the objective of traffic grooming algorithms becomes to reduce the number of transceivers (electrical ports used in the Electrical Cross-Connect (EXC)). The authors in [C1] formulate the problem of static traffic grooming as an ILP which enables to analyze some regular topologies. For the general case, the authors propose a heuristic approach that relies on an iterative greedy scheme where the traffic grooming problem is solved for one connection request at a time. For each request, this heuristic tries to simultaneously route and groom the considered request on a suitable path.

Recent research works [C2, J61, J62] started to consider dynamic traffic patterns in WDM mesh networks. The work in [J61] proposes a Connection Admission Control (CAC) scheme to ensure that the network will treat every connection fairly. It has been observed in [J61] that, when most of the network nodes have grooming capability, the high-speed connection requests will have higher blocking probability than the low-speed connection requests in the absence of any fairness control. The CAC is needed to guarantee that every class of connection requests will have similar blocking probability performance. The work in [C2] proposes a theoretical capacity correlation model to compute the blocking probability for WDM mesh networks with constrained grooming capability. In [J62], the authors propose a generic graph model for traffic grooming in heterogeneous WDM mesh networks. The originality of this model is that, by only manipulating the edges of the auxiliary graph constructed by the model and the weights of these edges, the model can achieve various objectives using different grooming policies while taking into account various constraints such as the number of transceivers at each node, the number of wavelengths on each fiber-link, as well as the wavelength conversion capability and the grooming capability of each node.

Considering dynamic traffic scenarios, four routing and grooming solutions are possible in order to satisfy a new incoming request:

1. route the traffic request onto an existing lightpath directly connecting the source node to the destination node
2. route the traffic request using multiple existing lightpaths
3. set-up a new lightpath directly connecting the source node and the destination node, and route the traffic request onto this lightpath
4. set-up one or more new lightpaths, and route the traffic onto these lightpaths and some already existing lightpaths

The first solution, if feasible, is the best method to set-up the connection. In some situations, all solutions are feasible, while in other situations, only some of them can be used. Considering a particular situation where multiple operations can be applied, combining the various solutions in different orders can achieve different grooming policies. If none of the aforementioned schemes can be applied, other traffic engineering schemes such as rerouting may be addressed.

1.3 Reconfiguration Techniques

Usually, when connection requests are added or removed from the network, the routing solution of older connections is not modified. Consequently, after some traffic additions and removals, the overall use of network resources is likely to be far from optimal. Assuming the wavelength continuity constraint, a new connection request may be rejected if no common wavelength is available along all the links of its candidate paths. The wavelength continuity constraint can be eliminated by means of wavelength converters. As a result, the use of wavelength converters can significantly reduce the connection rejection ratio and thus, increase network efficiency. Even with wavelength conversion capability, resources contention between various connection requests may result in the use of inefficient long path for some connections and/or in the blocking of other connection requests. A new connection request can be blocked because neither a new lightpath can be established nor an existing lightpath has enough bandwidth to accommodate the new connection. Either with or without wavelength conversion capabilities, logical topology reconfiguration and rerouting techniques are effective ways to increase the network resources utilization and thereby improve the network performance at the cost of additional signaling processes.

1.3.1 Logical Topology Reconfiguration

A key feature of WDM networks is the inherently high degree of configurability owing to tunable transceivers. More specifically, the flexibility of WDM networks lies in the ability to admit a large number of lightpaths within the network. Indeed, this is a feature that distinguishes WDM networks and drives a significant portion of the optical communication-theoretic research community.

One important area of current research implying network configurability is the problem of dynamic scheduling of logical topology reconfigurations. The reconfiguration management scheme changes a logical topology in response to changing traffic patterns in the higher layer of a network or in response to a network failure. There are two possible approaches for the logical topology reconfiguration, namely the “global reconfiguration” option and the “local reconfiguration” option.

Global Reconfiguration

The goal of the global reconfiguration is to have at each moment the most optimal logical topology with respect to the new traffic pattern. The logical topology is completely recomputed from scratch to obtain a new optimal topology with respect to the new traffic flows. The new configuration is independent of the current configuration and does not take into account the configuration migration sub-problem which can result in large network disruption.

In [J63], the reconfiguration problem is formulated using Integer Linear Programming (ILP). The proposed reconfiguration algorithm proceeds in two steps. As the first step, it solves the new virtual topology problem under a new demand. The goal of the optimization is to minimize the average physical hop length of the lightpaths in the virtual topology. Using this optimal average hop distance as a constraint, the virtual topology design problem is reformulated, as a second step, to minimize the number of changes required in order to migrate to the new virtual topology. The authors in [C3] extend the objective in [J63] to minimize the total number of lightpaths, as well as the number of physical links.

While this method is guaranteed to find a virtual topology that is optimal for the new conditions, it does not achieve a balance between finding this optimal virtual topology and minimizing the traffic disruption or maximizing the network availability. It is possible that a very costly reconfiguration will be undertaken for only a slight gain in network performance. More balanced formulations of this problem are possible, and heuristics designed on such formulations are likely to perform better in practice. In addition, since the reconfiguration problem is proved to be an *NP*-hard problem, the linear programming approaches do not scale well for large networks.

Local Reconfiguration

In order to minimize the degree of network disruption, the local reconfiguration improves the configuration of the network while scarifying the optimality of the new connection. More specifically, such schemes focus on a smaller number of more beneficial configuration changes rather than considering the reconfiguration of the entire network.

One approach tries to create lightpaths to relieve congestion and to proactively delete under-utilized lightpaths. In this approach, the algorithm identifies a congested link and attempts to create a lightpath between the endpoints of the greatest contributing multi-hop traffic flow. In the absence of congested links, the algorithm attempts to delete an under-utilized lightpath without disconnecting the logical topology. Congested and under-utilized links are defined in terms of high and low load thresholds, which are specified as parameters of the algorithm [C4, J64].

Another approach explores a set of neighboring configurations and the most optimal of these configurations is adopted. We define the set of neighbors as the set of configurations that are reachable from the current configuration using a single reconfiguration operation [J65].

Reconfiguration Policies

Whereas reconfiguration algorithms are concerned with the task of updating the configuration of the network, reconfiguration policies are concerned with the process of deciding when such algorithms should be executed. The simplest approach lies in executing the reconfiguration algorithm at regular time intervals [C5, C6, J66]. In that case, the interval length must be chosen to reflect the rate of changes in the traffic pattern with respect to the degree of adaptation achieved at each execution of the algorithm. Other approaches take into account the current traffic request. Reconfiguration can be performed at every traffic change [J67] or only when an important change is detected, for example on the basis of load thresholds [C4, J64]. A more sophisticated reconfiguration approach [C7, J65] considers both the benefit of such reconfiguration and its associated cost. The "reconfiguration benefit" is a function of the degree of load balancing achieved by the new configuration, while the "reconfiguration cost" is a function of the number of configuration changes necessary to migrate to the new logical topology. As a result of this formulation, a decision is made to either perform one of these possible reconfigurations or to remain in the current configuration.

Configuration Migration Techniques

A configuration migration technique for multi-hop broadcast networks is explored in [J68]. Because each logical link requires a dedicated wavelength in this architecture, links must be reconfigured carefully in order to avoid resource conflicts. To this end, the "Branch-Exchange" operation is proposed, where the logical links between a pair of transmitters and a pair of receivers are rearranged as illustrated in Figure 1.1. In this Figure, solid arrows indicate unidirectional logical links, each using a dedicated wavelength.

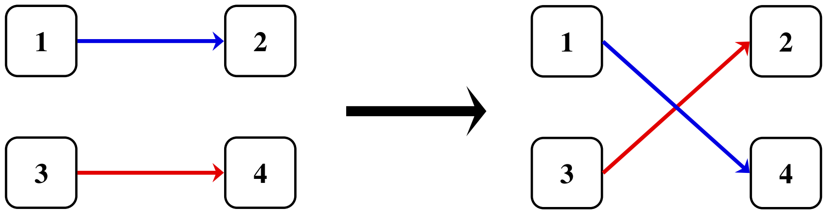


Fig. 1.1. Branch-Exchange operation

It can be shown that a sequence of Branch-Exchange operations can transform an initial logical topology to any target logical topology with the same number of links. Naturally, the problem of finding the shortest such sequence is important in minimizing the network disruption period during the configuration migration phase. Two key properties that simplify this problem are identified in [J68]. First, the shortest sequence only contains Branch-Exchange operations that decrease the number of links that remain to be established in order to reach the target configuration. For comparison, in the general case, a Branch-Exchange operation can also preserve or increase this number. Second, two Branch-Exchange operations that do not operate on the same logical link can be applied in any order without affecting the final result. These properties can be exploited in constructing heuristics that produce approximate solutions to the shortest Branch-Exchange sequence problem.

Another configuration migration technique, called "Move-To-Vacant" (MTV), is introduced in [J69]. Move-To-Vacant technique reroutes a lightpath to a vacant route whose links are free and are not used by any of the other lightpaths. This scheme has many desirable features. First, Move-To-Vacant does not interrupt other circuits since there is no other lightpaths on the new route of the rerouted lightpath. Second, it preserves transmission on the old route during the set-up of the new route. Therefore, there is no data lost and the disruption period is reduced.

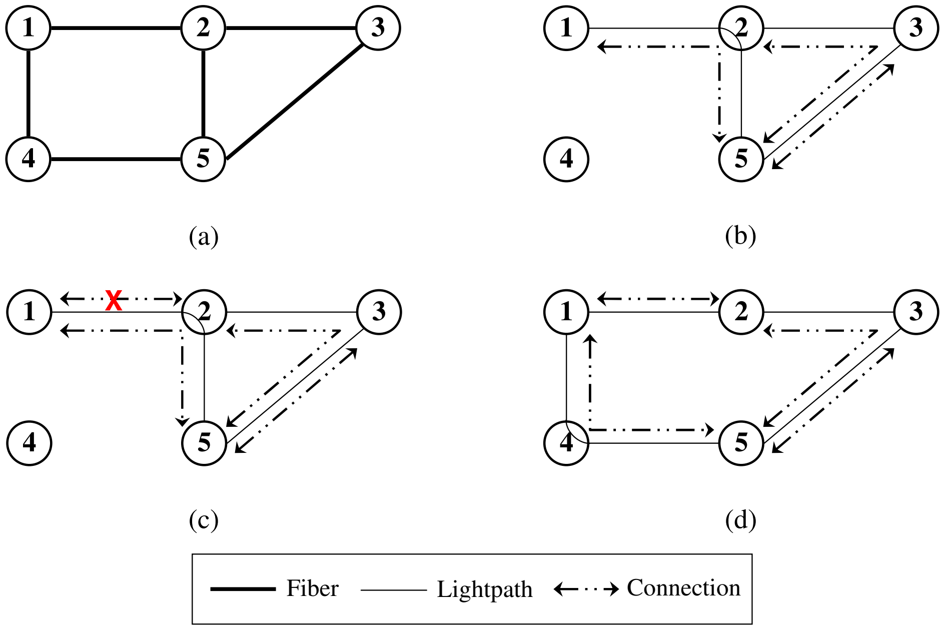


Fig. 1.2. Move-to-vacant reconfiguration technique

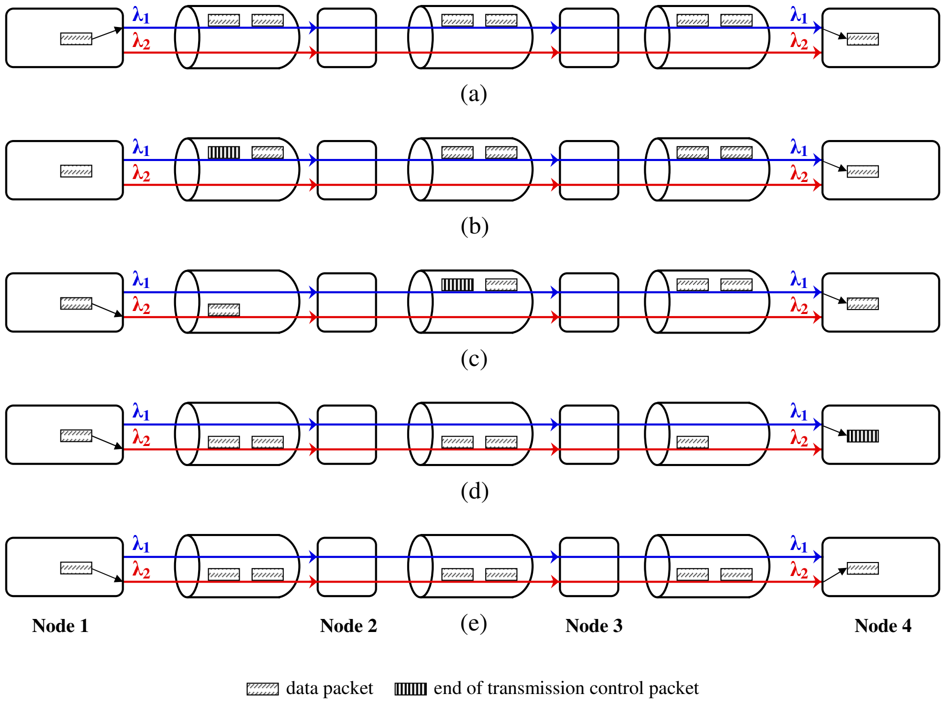
Figure 1.2 presents an example of the move-to-vacant configuration technique. We consider a small network whose physical topology is given by Figure 1.2(a). Each fiber of this network has only one wavelength λ_0 and can carry two connections. Each connection is assumed to be bidirectional in the sense if a connection exists between node 1 and node 2, another connection exists between node 2 and node 1. Figure 1.2(b) shows the current logical topology configuration and the connections carried by this logical topology. If a connection request $1-2$ arrives (Figure 1.2(c)), it cannot be accepted because neither does an existing lightpath $1-2$ exist nor a new lightpath $1-2$ can be established. Indeed, the wavelength λ_0 on link $1-2$ has been allocated to the lightpath $1-5$ which uses the route $1-2-5$. However, if this lightpath $1-5$ is rerouted to the route $1-4-5$, as shown in Figure 1.2(d), then a new lightpath $1-2$ can be established and the new connection request $1-2$ can be satisfied using this newly established lightpath.

1.3.2 Rerouting

As a new connection request arrives, the network should select an appropriate path to route the new demand without rerouting any existing circuit. If the initial attempt fails, the routing controller will determine which existing circuits should be rerouted in order to accommodate the new connection request. When some connections need to be rerouted, the routing controller should also determine the new paths to be assigned to these rerouted connections. In general, rerouting may change the path and/or the wavelength of the existing circuit.

The work in [J69] proposes a Move-To-Vacant Wavelength Retuning (MTV-WR) rerouting scheme. On one hand, as introduced earlier, MTV technique achieves a small rerouting disruption time. On the other hand, Wavelength Retuning (WR) retunes the wavelength of the lightpath to another wavelength on the same physical path. By combining these two schemes, MTV-WR moves a lightpath to a vacant wavelength on the same path. It has the advantages of both MTV and WR. However, WR is unable to reduce the resource inefficiency caused by lightpaths using extremely long paths. Figure 1.3 illustrates how the MTV-WR is implemented. This figure illustrates the chronological events that occur when retuning the request from the wavelength λ_1 (*c.f.* Figure 1.3(a)) to the wavelength λ_2 (*c.f.* Figure 1.3(e)). In (a), the network controller sends a control message to the intermediate switches on the path of the routed circuit to set-up a new route on a vacant wavelength. Then, the origin node prepares to switch data transmission from the old to the new wavelength. In (b), the origin node appends an end-of-transmission control packet after the last packet on the old wavelength and holds the first packet on the new wavelength for a guard time (this guard time will determine the length of the disruption period). The end-of-transmission packet is used to inform the destination node that data transmission via the old wavelength has ended and that data will soon arrive via the new wavelength. The guard time prevents data from being lost during the transient period of circuit migration. In (c), (d), and (e), data transmission is switched from the old to the new vacant wavelength. The information source switches transmission by retuning the wavelength of the optical transmitter to the new wavelength. At the destination node, the information sink switches reception by retuning the wavelength of the receiver to the new wavelength upon detecting the end-of-transmission control packet.

The work in [C8] proposes a routing and traffic grooming scheme. It includes two phases: a normal routing phase and a rerouting phase. The rerouting phase is only executed when the normal routing phase fails to find a path for the arriving connection request. It aims at rerouting some lightpaths or connections so that the arriving connection request can be accepted. At the end of this rerouting phase, if these lightpaths or connections are rerouted successfully, the new connection request is provisioned using the routing algorithm of the normal routing phase. In this work, the normal routing phase uses the Least Virtual Hop First (LVHF) routing algorithm. LVHF aims at minimizing the



(a)

(b)

(c)

(d)

(e)

Fig. 1.3. Implementation of the MTV-WR operation of circuit migration

number of virtual hops (grooming lightpaths) to carry a connection on the fixed alternate paths. As for the rerouting phase, the rerouting operation is restricted to one lightpath or connection for each incoming/new connection request. This restriction not only reduces the complexity of the rerouting algorithm but also reduces the amount of traffic affected by the rerouting operation. The rerouting operation is also based on the set of K-shortest paths computed off-line for each source-destination pair. Such technique does not take into consideration the current state of the network and reduces furthermore the complexity of the rerouting algorithm.

In traffic grooming, the optical layer and the electronic layer are involved in the routing problem. Hence, rerouting techniques can be applied at two different levels, namely Rerouting at the Physical Optical Level (RPOL) and Rerouting at the Logical Electrical Level (RLEL). Generally, RPOL is a coarse-granularity rerouting scheme which operates at the high-rate lightpath (aggregate) level whereas RLEL is a fine-granularity rerouting scheme which operates at the low-rate connection (per-flow) level. Illustrations of rerouting at lightpath and connection levels are given in Figures 1.2 and 1.4, respectively. Please refer to the section on MTV for an example on lightpath rerouting. Here, we consider only an example on connection rerouting. For this task, we consider a small network whose physical topology is given by Figure 1.4(a). Each fiber of this network has only one wavelength λ_0 and can carry two connections. Each connection is assumed to be bidirectional in the sense if a connection exists between node 1 and node 2, another connection exists between node 2 and node 1. Figure 1.4(b) shows the current logical topology configuration and the connections established over this logical topology. A new connection request between nodes 4 and 3 is blocked because all the bandwidth capacity on lightpath $4-5$ has been used by two other connections. However, if the connection $1-5$ can be rerouted to lightpath $1-5$ on route $1-2-5$, as shown in Figure 1.4(d), then the connection $4-3$ can be carried over a two-hop path $4-5-3$ using lightpaths $4-5$ and $5-3$.

As a coarse-granularity scheme, the RPOL rerouting algorithm is relatively simple. It only needs the global information of the lightpaths in the network to decide the lightpaths to be rerouted with

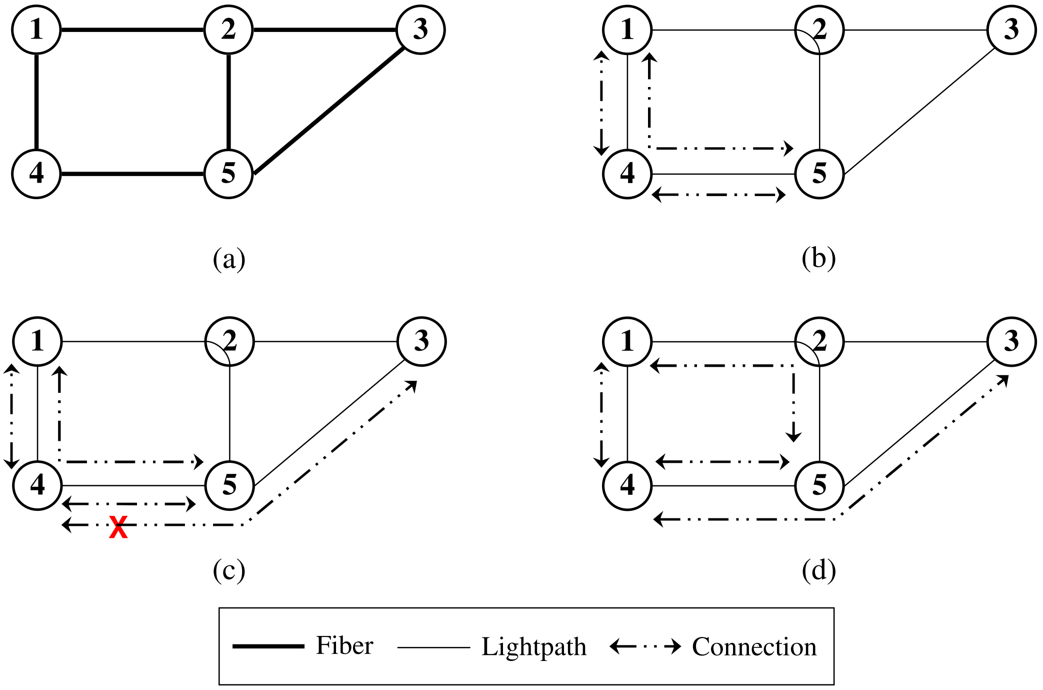


Fig. 1.4. Illustration of rerouting scheme at connection level

their new paths. The path transitions of the underlying lightpaths are transparent to the upper sub-wavelength connections. On the other hand, RLEL needs the global information of all the lightpaths and all the connections to compute the connections to be rerouted with their new paths. As the number of connections in the network is larger than the number of lightpaths, the RLEL rerouting algorithm usually has a higher complexity. However, as a fine-granularity scheme, RLEL is more flexible in terms of selecting rerouted connections with their new paths.

Example 1.1.

Let us consider an example of the parallel MTV-WR rerouting algorithm for a simple network scenario described in Figure 1.5. The maximum number of available wavelengths per fiber-link is 3. This network is decomposed into three subgraphs corresponding to each of the three wavelengths. These subgraphs are disjoint if the network does not allow for wavelength conversion. Let M denotes a number chosen larger than the cost of the longest possible lightpath (M represents the cost of retuning a single lightpath of unit cost). Three requests δ_1, δ_2, and δ_3 are already satisfied. For instance, δ_1 is already routed on wavelength λ_1 between node 2 and node 5.

We assume that a new request δ_4 arrives and needs to be routed between node 1 and node 4. In the initial routing phase, an active wavelength is assigned an infinite weight, while a free wavelength is assigned a small weight ϵ. No path with finite weight can be found for this new connection request. Thus, the rerouting phase is initiated. This rerouting phase is decomposed into three steps:

- First, we identify all the retunable lightpaths in the network. The cost of each link of a retunable lightpath is replaced by the cost of retuning the totality of this lightpath. This cost is equal to M times the cost of this lightpath when it was created. However, the

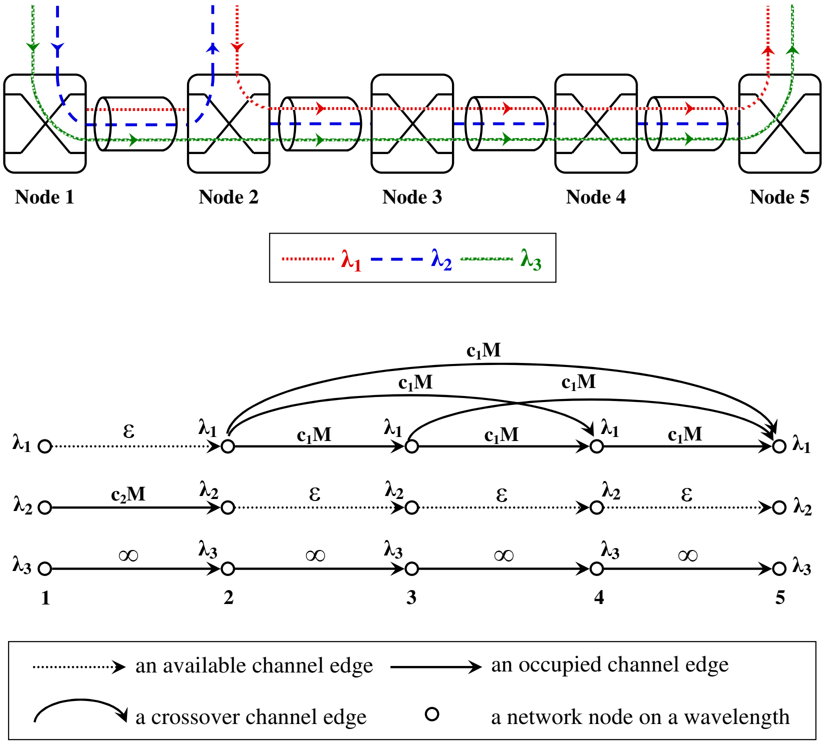


Fig. 1.5. Illustration of the graph transformation of the parallel MTV-WR rerouting algorithm

cost of a non-retunable lightpath remains infinity. For instance, the edge between node 1 and node 2 on wavelength λ_1 is free and will be assigned a cost ϵ. However, the edge between node 1 and node 2 on wavelength λ_2 is active and corresponds to a retunable lightpath. Let us assume that the cost of this lightpath was c_2 at its establishment. Consequently, the weight assigned to the edge between node 1 and node 2 on wavelength λ_2 is $M \times c_2$. Finally, the edge between node 1 and node 2 on wavelength λ_3 is active, but its corresponding lightpath is not retunable. Hence, its assigned weight remains equal to infinity.

- Second, the graph is slightly modified by introducing crossover edges for every retunable lightpath. For instance, let us consider the request δ_1 between node 2 and node 5 routed along the path $2 - 3 - 4 - 5$ on wavelength λ_1. As this request can be retuned on wavelength λ_2, crossover edges are established between any two non-adjacent nodes in the set $\{2, 3, 4, 5\}$. A common weight is assigned to all these crossover edges. This weight is equal to the cost of retuning the totality of the lightpath (or sub-lightpath if wavelength conversion is used) used by δ_1. In our example, this cost is equal to $M \times c_1$.
- Finally, using conventional shortest path algorithm, the path with the minimum weight between node 1 and node 4 is selected. In our example, if c_2 is smaller than c_1, the path $1-2-3$ on wavelength λ_2 is selected since $(M \times c_2 + 2 \times \epsilon)$ is smaller than $(M \times c_1 + \epsilon)$. As a result, the request δ_2 must be retuned on wavelength λ_1 before satisfying the incoming request δ_4 on wavelength λ_2.

In general, if no path with a finite cost is possible at the end of the last step, the incoming connection request is rejected.

1.4 Motivation

Traditionally, two main approaches can be considered in order to solve the Routing and Wavelength Assignment (RWA) problem in WDM networks: static and dynamic.

- On the one hand, static RWA refers to network dimensioning and considers permanent or scheduled (predictable) dynamic traffic requests. The source, the destination, and the requested bandwidth associated to each traffic demand are known in advance. The set of these three parameters corresponds typically to long-term traffic estimations. Optimization tools used for optical network planning are not subject to stringent constraints in terms of computation duration. Indeed, even if it may be complex, the computation of these optimization tools is carried out, in general, offline on a centralized computer. Optimization tools used in network planning proceed to a global optimization in the sense they consider all the demands at once to compute the route and the wavelength(s) to assign to each demand. The objective of static RWA is to minimize the global network cost assuming a given traffic matrix. According to the state of the technology, optical networks with static routing and wavelength assignment are technically feasible in the short term.
- On the other hand, dynamic RWA refers to Traffic Engineering (TE) and considers random dynamic traffic requests. Since the year 2000, the Internet Engineering Task Force (IETF) [W159], the Optical Internetworking Forum (OIF) [W160], and the International Telecommunication Union-Telecommunication Standardization Sector (ITU-T) [W161] collaborate to specify the characteristics of agile hybrid optical networks. Agility assumes that intelligence has been implemented in the network nodes in order to enable real-time provisioning of optical circuits for random traffic demands. The objective of dynamic RWA is to minimize the rejection ratio of traffic demands assuming a given network capacity.

Most of these studies have assumed that the whole capacity of an optical channel is dedicated to each connection request during its life duration. Recent research works started to consider traffic requests that require a fraction of the optical channel capacity. These studies investigate the more practical problem of how to efficiently groom low-speed connection requests into high capacity lightpaths referred to as *"Grooming Lightpaths"* (GLs). Moreover, we need to assign paths and wavelengths to each grooming lightpath such that any two GLs passing through the same physical link are assigned different wavelengths. We assume a single fiber network system. As it is the case in current carriers network, two contra-directional fibers may be used to connect two adjacent nodes. The wavelength selection may be performed either after a path has been determined, or in parallel while finding a path. In general, if there are multiple feasible wavelengths between a source node and a destination node, different wavelength assignment algorithms can be adopted. A review of wavelength assignment approaches can be found in [J70, J71].

An example of a simple and effective wavelength assignment heuristic is First-Fit (FF). In FF, the wavelengths are indexed according to their increasing wavelength value expressed in nanometers. Wavelength assignment consists in assigning the first available wavelength with the lowest index. Another wavelength assignment approach consists in randomly selecting a wavelength from the set of available wavelengths. Other simple wavelength assignment heuristics include the Most Used Wavelength (MUW) heuristic and the Least Used Wavelength (LUW) heuristic [C9]. In most used wavelength assignment, the wavelength which is the most used in the rest of the network is selected. This approach attempts to provide maximum wavelength reuse in the network. The least used approach attempts to spread the load evenly across all wavelengths by selecting the wavelength which is the least used throughout the network. Both most used and least used approaches require global knowledge of the network state. Ten wavelength assignment approaches have been compared in [J70]. The performance of these approaches may vary depending on the working scenario.

To the best of our knowledge, all the studies in optical WDM networks consider one type of traffic request at a time; either deterministic or random. In order to investigate the impact of traffic predictability on network performance, we consider both types of traffic requests on the same infrastructure. Moreover, we distinguish two types of random demands: pure random demands and semi-random demands. In short, we consider three traffic classes: *Scheduled traffic Demands* (SxDs), *Random traffic Demands* (RxDs), and *semi-Random traffic Demands* (sRxDs). The SxDs are deterministic demands with predetermined date of arrival, life duration, and data rate, while sRxDs and RxDs correspond to connection requests with random instant of arrival and life duration. The life duration of an sRxD is known at its instant of arrival, while the life duration of RxDs remains unpredictable. Depending on the value of the required rate, we can distinguish between two classes of demands: *Lightpath Demands* (xLDs) and *Electrical Demands* (xEDs). The former are supposed to carry a traffic rate C_ω(in bps) equal to that of a WDM channel, while the latter are supposed to carry a traffic rate less than C_ω. As a result, we consider 6 types of requests: SLDs, SEDs, sRLDs, sREDs, RLDs, and REDs.

In this thesis, we investigate the routing and grooming problem for dynamic traffic requests in a multi-granular optical WDM network environment. The wavelength assignment problem is out of the scope of our study. For this task, we assume that each network node is composed of an Electrical Cross-Connect (EXC) coupled to an Optical Cross-Connect (OXC). This network node can also be represented by means of an equivalent “auxiliary graph”. This auxiliary graph has the same functions as the original node representation and enables easy handling and managing traffic grooming in multi-layer WDM mesh networks. As the SxDs (SLDs and SEDs) are known in advance, they can be routed by means of global optimization tools. The proposed algorithm determines the best routing for SxDs that minimizes the global network cost. Multiple SEDs can be electrically aggregated together if they share at least one common fiber-link and if they are active during a common period of time. In general, the global cost is expressed in terms of required electrical ports and optical ports. This is carried out by the management plane which operates in the long-term and corresponds to a centralized off-line computation. Once the SxDs are routed, the management plane informs each node of the availability of the various equipment (electrical ports and optical ports) within the network.

The control plane is in charge of routing sRxDs and RxDs that arrive one at a time at the network and need to be routed on the fly. This is achieved by a sequential optimization tool based on a shortest path algorithm combined with an appropriate dynamic cost assignment for the different edges of the network auxiliary graph. Note that sREDs and REDs are also groomed on the fly by the proposed algorithm. The control plane operates in real time and is distributed in all the nodes of the network. One assumes that Generalized Multi-Protocol Label Switching (GMPLS) signaling, still under specification[1], is used to share information between the two planes.

The impact of the unpredictable life duration of RxDs is best evaluated when the network carries a mix of scheduled traffic and random traffic. In this context, searching on the fly for free network resources is not trivial. For instance, let us consider a fiber-link with two free wavelengths as depicted in Figure 1.6. Let us assume that all the traffic requests to be routed on the network require a full

[1] [Draft IETF]: Generalized Multi-Protocol Label Switching (GMPLS) Protocol: Extensions for Multi-Layer and Multi-Region Networks (MLN/MRN) (November 2007). Available at: http://www.ietf.org/internet-drafts/draft-ietf-ccamp-gmpls-mln-extensions-00.txt

optical channel capacity. Once the SxDs are routed, it may happen that an SxD D_1 is scheduled to use this fiber-link starting from instant t_2. In addition let us suppose that an RxD d_1 uses this same fiber-link starting from instant t_0 ($t_0 < t_2$). According to the network state at t_1 ($t_0 < t_1 < t_2$), a new RxD d_2 arriving at t_1 can use the available wavelength of this fiber-link. This assumption is true if either d_1 or d_2 ends before the instant t_2, but it results in a conflict if both d_1 and d_2 remain active after t_2. By convention, an SxD such as D_1 must benefit from its guaranteed resources. In the worst case scenario where the RxDs d_1 and d_2 may have an infinite life duration, d_2 must be rejected at its instant of arrival t_1 because of the lack of available resources starting from t_2. As a result, RxDs must use distinct resources from those assigned for SxDs.

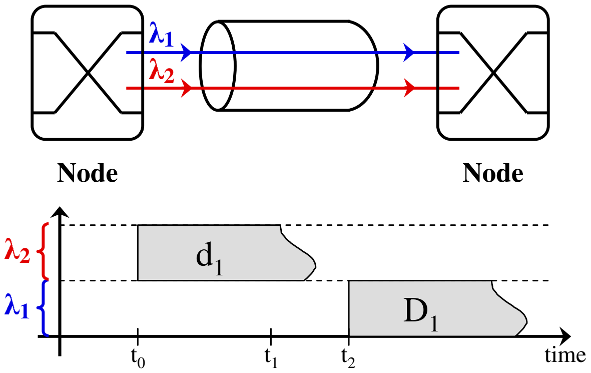


Fig. 1.6. Determining free network resources

As a first approach, assuming that the life duration of the RxDs is infinite, we can evaluate the rejection ratio of these demands. However, compared to the case where the random demands could be characterized by a predictable life-duration (sRxDs), the overall network resources utilization is still inefficient. Indeed, for a given network and a given traffic scenario, a huge gap can be noticed between the rejection ratio of RxDs and that of sRxDs. In order to enhance network performance and to reduce the request rejection ratio in the context of dynamic and random traffic demands assuming pre-established demands, several rerouting strategies are addressed.

1.5 Thesis Overview

This thesis is organized into two parts. The first part is a literature survey of the current and previous work that has been done in the field of WDM optical networks. It includes two chapters. The current chapter is a general introduction to the study. It reviews the relevant literature and provides background information needed to formulate study objectives. The second chapter explores some of the latest research topics in the WDM optical network area. The second part is our contribution to this area. It covers different problems varying from the network planning to the traffic engineering and can be decomposed into 6 chapters:

In chapter 3 titled “Backbone Traffic Characterization”, we propose an original traffic characterization based on a pragmatic approach. From the observations of real traffic traces, we propose to decompose the traffic load of a network into a set of scheduled demands represented by SxDs and a set of random demands represented by sRxDs and RxDs. An original aspect of our proposed traffic characterization consists in taking into account the impact of time-zone shifts on large scale networks.

A similar traffic model is considered by Gençata [C10, D155]; the author assumed that a typical traffic profile between any two nodes follow a sinusoidal shape, where the peak is around midday and the valley around early hours of the morning. Since different cities have different time-zone shifts, each node will create a traffic whose intensity changes according to its time-zone. The 24-hour traffic intensities between any two cities is then generated by randomly selecting one of the several sampled traffic profiles from real backbones. However, no additional details are given concerning the way this traffic is generated, its granularity, or its predictability/randomness. Thus, our approach is the first to give a fully detailed mathematical description of backbone traffic characterization including time-zone shifts, different traffic granularities, and covering the time predictability scale from the most deterministic to the most random in terms of arrival date and life duration of the requests. In this thesis, all our numerical evaluations are applied to the NSFNET North American backbone [W162, W163] since it is the widely adopted reference model in the literature.

In chapter 4 titled "Nodal Architecture", we describe into detail the EXC/OXC multi-layer node architecture that enables to groom electrical connections (MPLS-LSPs, ATM-VCs, ATM-VPs, Synchronous Digital Hierarchy (SDH) connections, ...) into grooming lightpaths. For such a node, we also propose an associated auxiliary graph partially inspired from a previous study in the literature [C11, B150].

In chapter 5 titled "Routing and Grooming of Scheduled Demands (SxD)", we investigate the problem of efficiently designing a WDM mesh network with grooming capability for a given set of traffic requests via theoretical formulation as well as simulation experiments.

Chapter 6 titled "Routing and Grooming of Random Demands (sRxD/RxD)" is dedicated to Traffic Engineering in the context of random demands. For this task, we define various weight assignment strategies to be applied to the auxiliary graph of the network.

In chapter 7 titled "Rerouting Strategies", we consider two strategies for traffic rerouting. In the first strategy, we extend the concept of rerouting using crossover edges for a multi-layer EXC/OXC network assuming unidirectional fiber-links. The second strategy deals with the concurrency between scheduled and pure random demands. By convention, an SxD must benefit from its guaranteed resources. In this context, for a given network and a given traffic scenario, a huge gap exists between the rejection ratio of RxDs and that of sRxDs. In order to narrow this gap, we propose an original approach called "Time Limited Resource Reservation" (TLRR) applied to RxDs. Our numerical results outline that the TLRR policy enables to achieve rejection ratios for RxDs comparable to those obtained for sRxDs.

Finally, Chapter 8 presents the conclusions of the thesis and some future tracks.

2

Operator's Network Evolution

Abstract:
In this preliminary chapter, we aim at providing a brief overview of some key architectural evolutions in telecommunication core and access networks. Operator's network evolution is always driven by the ever-increasing demands for new fixed and mobile broadband services as well as continuous advances in communication systems. The rapid evolution of network technologies presents to network operators many challenges. There is always the need to roll-out new services and exploit new technologies to maintain a competitive advantage, but there is also the continuing need to get the maximum return on past investments and current generation technologies and services. In order to face the competition and to meet the unprecedented demand for bandwidth capacity, most operators will migrate to next generation networks.

2.1 Introduction

In this chapter, we briefly discuss some new technologies and techniques, especially from the network perspective, that facilitate seamless communication across the network. We present five topics that are the subject of current investigations in WDM optical network area. These topics are not directly related to the subject of this thesis, but they are presented here to give a global overview on current optical technologies and researches. These topics are: *Multi-layer Recovery Approaches*, *Quality of Transmission Aware Networks*, *Grid Networks*[1], *Network Coding*[1], and *Radio Over Fiber Technology*[1].

2.2 Multi-layer Recovery Approaches

The integration of different network technologies, like for instance IP/MPLS and Optical Transport Network, into multi-layer transport networks leads to new opportunities and also new challenges concerning the survivability of such multi-layer networks. Those opportunities lie in the fact that, in such networks, recovery techniques carried out at different network layers can cooperate to recover more efficiently or faster from a network failure. This also brings new challenges and difficulties with

[1] The last three topics have been moved to the annex upon the request of the thesis jury.

regard to the coordination of those mechanisms in the different layers. A major challenge consists in defining new signaling protocols enabling fast communications between network layer elements.

Different control plane architectures assume different integration levels of the control planes of the different layers [B151].

Overlay Model: The first control plane model is the "overlay model" where each layer runs its own control plane. In order to couple the different layers, an appropriate protocol must be provided to allow the communication between the different control planes. This protocol must, for example, provide address resolution between the layers and/or initiate the connection request/release. The main drawback of this model is the duplication of the control functionalities; *e.g.,* separate routing protocols are running in different layers. Another disadvantage is the scalability problem; for each established lightpath, a corresponding routing adjacency has to be established. A final drawback of this model is the clear client/server relationship between the different layers; address resolution is required because of separate address spaces. On the other hand, due to the separation of the control plane instances, no confidential information from one layer can be disclosed to the other network layers. In addition, the overlay model is suited for the interconnection with legacy SDH networks.

Peer Model: In order to solve the drawbacks of the overlay model, the "peer model" was proposed. In this model, a single control plane ensures the control of all layers. As a result, the disadvantages of the client/server relationship between the different layers no longer exist and the duplication of functionalities is avoided. Moreover, no additional peering session is required per established lightpath, which may solve some scalability problems. However, the peer model is unable to be applied to all imaginable business models. For example, in an IP-over-WDM network scenario, the optical transport network operator may not accept that the Internet Service Provider (ISP) takes over the control of its transport network (or vice versa). In addition, the peer model is limited to a single domain or autonomous system.

Augmented Model: The third control plane model is the "augmented model". This model is a compromise between both the overlay model and the peer model. It is quite similar to the overlay model, in the sense that all layers may have their own control plane instance. However, all layers can exchange control information like reachability information. Note that although everyone agrees that the augmented model is situated somewhere in between the two extreme overlay and peer models, there is not yet a clear understanding or consensus about the definition of this augmented model. Nevertheless, it is clear that the augmented model tries to find a compromise between the advantages and the disadvantages of both extremes.

The survivability and recovery mechanism in multi-layer networks can be grouped into three generic categories:

1. **Single-layer recovery schemes in multi-layer networks:** In this case, one of the multiple stacked layers is chosen *a-priori* to proceed to the protection of the whole system.
2. **Static multi-layer recovery schemes:** In this situation, recovery mechanisms are activated at the different layers of the system. The main problem is then to coordinate these mechanisms in order to provide the best restoration efficiency.
3. **Dynamic multi-layer recovery strategies:** In the two previous approaches, the disrupted traffic is rerouted on the remaining operational lightpaths. By this way, we try, as best as possible, to maintain the logical topology seen by the highest layer. In this third approach, new lightpaths

may be established whereas others may be turned down, inducing in general a new logical topology at the highest layer.

In the following, the concepts and discussions focus on a two-layer network but are generic and can thus be applied to any multi-layer network.

2.2.1 Single-layer Recovery Schemes in Multi-layer Networks

A multi-layer transport network can be viewed as a stack of single-layer networks. Each of these network layers has its own (single-layer) recovery scheme which can effectively handle a large number of failure scenarios. For example, in an optical transport network, 1 + 1 optical protection is used against the failure of an OXC or a fiber cut. A simple approach to provide survivability in a multi-layer network is to deploy a recovery mechanism in only one single layer of the multi-layer network. This approach is known as single-layer recovery scheme [J72, B151]. Using this strategy, it remains to choose in which network layer, the recovery scheme should be deployed.

Survivability at the Bottom Layer

In the "survivability at the bottom layer" approach, the recovery of a failure is done at the bottom layer of the multi-layer network. For example, in an IP-over-WDM network, the 1+1 optical protection scheme (or any other recovery scheme deployed in the WDM optical transport layer) will attempt to restore the affected traffic when a failure occurs. By recovering a failure at the bottom layer (WDM), this strategy deals with a simple root failure. Thus, the number of required recovery actions is minimal as these actions are performed on the coarsest granularity. In addition, the signalling information related to failures does not need to propagate through multiple layers before it triggers any recovery action. However, one of the major drawbacks of this recovery strategy involves its inability to cope with problems or failures that occur in a higher network layer. In addition, there are also situations in which the recovery process in the bottom layer is not able to restore all the affected traffic (*c.f.* Example 2.1), whereas a higher layer recovery mechanism would be able to.

Survivability at the Top Layer

Another approach to recover from failures in a multi-layer network is the "survivability at the top layer" strategy. This strategy can easily cope with higher layer failures and can implement differentiated services depending on the priority of the traffic. For instance, the top layer may choose to recover the critical high priority traffic before it takes any recovery action for the low priority traffic flows. Such a service differentiation among traffic flows based on the order of the recovery action is not possible in lower layers because of traffic aggregation. The lower layers switch all the flows within an aggregated signal with one single recovery action discarding the priority of the individual traffic flows. Under certain conditions, this finer granularity of the flow entities in the top layer may also lead to a more efficient capacity usage. First, the aggregated signals that are poorly filled with working traffic have enough capacity left to transport spare resources. Second, the finer granularity allows distributing flows over more alternative paths. Conversely, because of this finer granularity, this strategy typically requires a lot of recovery actions and a large number of higher layer equipment. In addition, a single root failure in a lower layer can result in multiple simultaneous failures in the top network layer. Thus, the recovery scheme in the top layer has to recover from the typically complex multiple failures scenario. However, the lower layer recovery scheme only have to cope with the single failure scenario.

Survivability at the Lowest Detecting Layer

A slightly different variant of the survivability at the bottom layer strategy is the "survivability at the lowest detecting layer" strategy. The lowest detecting layer is the lowest layer in the layered network hierarchy that can detect the failure. In this strategy, multiple layers in the network will deploy a recovery scheme. However, this strategy is still considered as a single-layer recovery scheme in a multi-layer network since, for each failure scenario, the (single) layer that detects the root failure is still the only layer that takes any recovery actions. With this kind of strategy, the problem that the bottom layer recovery scheme does not detect a higher layer failure is avoided because the higher layer that detects the failure will recover the affected traffic. However, although this strategy can guarantee that the traffic transiting the failing equipment in the detecting layer is restored, it still suffers from the fact that it cannot restore any traffic transiting higher layer equipment isolated by a node failure in the detecting layer (*c.f.* Example 2.1).

Survivability at the Highest-possible Layer

A slightly different variant of the strategy that provides survivability at the top layer is the "survivability at the highest-possible layer" strategy. Since not all the traffic has to be injected (by the customer) at the top layer, with this strategy a traffic flow is recovered in the layer in which it is injected, or in other words in the highest-possible layer for this traffic flow. This means that this highest-possible layer is to be determined on a per traffic flow basis. This strategy is also considered as a single-layer recovery scheme for providing survivability in a multi-layer network, even though it deploys different recovery schemes in the different network layers. Indeed, the survivability at the highest possible layer may lead to recovery schemes in multiple layers, but these schemes will never recover the same traffic flow. Actually, this strategy deploys the survivability at the top layer strategy for each traffic flow individually, which implies that in essence, both strategies do not differ from each other.

2.2.2 Static Multi-layer Recovery Scheme

As stated earlier, not every failure in a particular network layer can be handled by the recovery mechanism in that same layer. This is depicted in the following example.

Example 2.1.

Let us consider a two-layer network where the lower layer provides transport functionalities to the upper layer (Figure 2.1). This network carries two traffic flows δ_1 and δ_2 between the two upper-layer nodes a and c. Both traffic flows are routed along the physical path $A-D-C$. However, the request δ_2 transits through the upper-layer node d. As a result, δ_1 uses a direct logical link $a-c$ from the upper-layer node a to the upper-layer node c, while δ_2 uses two logical links $a-d$ and $d-c$.

Let us assume that a failure occurs in the lower layer, for example the failure of the lower-layer node D. This failure is first detected in this layer and a recovery action is initiated. This recovery mechanism will only be able to restore the affected traffic δ_1 that transits the failed lower-layer node D. However, it cannot recover the traffic δ_2 since, from the lower-layer point of view, this traffic is considered as two separate connections $A-D$

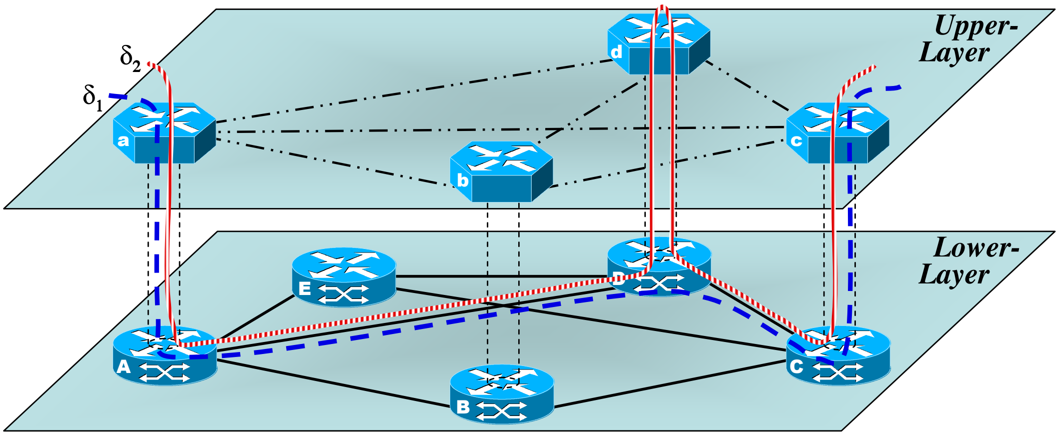


Fig. 2.1. A small example of a two-layer network

and $D-C$ originating from/terminating in the failing node. Subsequently, the upper layer has to recover this flow.

From the upper-layer point of view, multiple simultaneous failures (logical links $a-d$, $b-d$, and $c-d$) are detected, isolating the upper-layer node d. Upon detection of these faults, the upper layer could also initiate a recovery action. This will eventually lead to the use of the recovery path $a-c$ for instance.

The previous example illustrates that it is not simple to decide straightforwardly in which single layer of the multi-layer network to provide and deploy a recovery scheme. On one hand, recovery at higher network layers is desired since lower layers will not notice failures of higher layer equipment. In addition, recovery at higher layers is desired since higher layer equipment can become isolated due to a failure in the lower network layer (for instance the failure of the lower-layer node in the previous example). Only a recovery scheme in the higher layer is then able to recover the traffic that transits this isolated equipment. On the other hand, recovery at lower layers is desired since native traffic that is injected in lower layers cannot be recovered by higher layer recovery strategies. In addition, recovery at lower layers is desired to keep the number of recovery actions at minimum. A single failure in a lower layer can be detected as multiple simultaneous failures in higher network layers. The latter scenario is in general more complex. Implementing multi-layer recovery is desired to combine the advantages of the single-layer recovery approaches. Multi-layer recovery implies that recovery mechanisms will run in different layers of the network as a reaction to the occurrence of a single network failure. However, this requires some rules or coordination actions (called “escalation strategies”) in order to ensure efficient inter-working and coordination between the various network layers involved in the recovery process [J72, B151].

Uncoordinated Approach

The easiest way to provide a multi-layer recovery mechanism is to simply deploy recovery schemes in the multiple layers without any coordination at all. Upon a failure, all the layers that detect it will deploy their own recovery mechanism in order to recover the affected traffic. This results in parallel recovery actions at distinct layers. Since the recovery actions at the different layers are not coordinated, this solution is simple and straightforward from an implementation and operational point of view. For instance, no standardization of coordination signals between the layers is required. However, this approach suffers from excessive capacity utilization and network disruption during recovery. Indeed, each layer reserves spare resources in order to recover the affected traffic, even if only one recovery scheme occupying spare resources would be sufficient. This implies the reservation of more spare resources than necessary. Moreover, conflicts can occur between several layers when they try to reserve resources to recover from a failure. This can even lead to "destructive interference" where none of the recovery actions will be able to restore the traffic.

In summary, although simple and straightforward, just letting the recovery mechanisms in each layer run without any coordination strategy has its consequences on efficiency, capacity requirements, and even ability to restore the traffic.

Sequential Approach

A more efficient escalation strategy is the sequential approach where a chronological order on the recovery mechanisms is imposed. When the current network layer is unable to recover from the failure, this task is handed over to the next layer. Two approaches exist, the bottom-up escalation strategy and the top-down escalation strategy.

Bottom-up Escalation: The first strategy starts in the lowest detecting layer ensuring a fast activation of the recovery mechanism. All affected traffic that cannot be restored in this layer will be restored in a higher layer. The advantage of this approach is that recovery actions are taken at the appropriate granularity: first, the coarse granularities are handled recovering as much traffic as possible in the shortest delay, then the remaining finer granularity traffic flows are recovered. One question rises: How does a network layer know whether it is the lowest layer that detects the failure or not? In general, the fault signals, that are exchanged to indicate a failure, carry sufficient information to determine in which layer the failure occurred. However, if this is not the case, this problem can be overcome by using the mechanism of hold-off timers. Upon detection of a failure, each layer must wait until its built-in hold-off timer expires before it initiates any recovery process. These hold-off timers are set progressively higher as we move upwards in the stack of layers: 0 ms in the bottom layer, 20 ms in the first layer, 40 ms in the second layer, and so on. In this way, if the fault is fixed by a lower-layer recovery mechanism before the hold-off timer of a higher-layer expires, no recovery action will be taken by this higher-layer. However, if this hold-off timer expires and all or part of the traffic is not restored, then the higher-layer will take over the recovery actions. This is very straightforward; however, the main drawback of this approach is that the recovery actions in higher layers are always delayed, independently of the failure scenario, as the hold-off timer must expire first. Moreover, determining the optimal value for these hold-off timers is another challenge. On one hand, if the hold-off timer is set to a too small value, this may lead to a so-called "false-positive" recovery action, where the higher layer will trigger its recovery mechanism before the lower layer has completed its set of recovery actions. On the other hand, setting the hold-off timer to a too large value results in longer recovery time when the fault cannot be recovered in the lower layer. Such a compromise is not always straightforward notably because of the strong impact of the adopted technologies at the different layers. The recovery time objectives must imperatively be balanced with the network stability and performance in multi-layer recovery networks.

Top-down Escalation: The recovery actions are initiated in the highest-possible layer, and the escalation goes downwards in the layered network. Only if the higher layer cannot restore all the traffic, actions are triggered in the lower network layer. An advantage of this approach is its capability to implement differentiated quality of service but this implementation is somewhat more complex. Indeed, a lower layer cannot detect easily whether a higher layer is able to restore the traffic or not. Thus, when the higher layer cannot recover all or a part of the traffic flow, it must send an explicit signal called “recovery token signal” to the lower layer. This recovery token needs to be included in the standardization of the interface between network layers. Upon receipt of this token, the lower layer initiates its recovery mechanism. In addition, this approach lacks of efficiency. For instance, let us assume that only 50% of the traffic flow carried by the failing higher layer lightpath can be restored by the recovery mechanism in that layer. Hence, protecting this failing higher layer lightpath in a lower layer is only useful for the other 50% of the carried traffic.

Integrated Approach

A more radical approach to ensure coordination between the recovery mechanisms in different layers is to combine the different recovery mechanisms in a single integrated multi-layer recovery scheme. This recovery scheme needs to have a full overview of all the network layers in order to decide when and in which layer (or layers) to take the appropriate recovery actions. While this approach is clearly the most flexible one from the recovery point of view, combining different technologies in one mechanism is often unrealistic from a practical point of view. Indeed, in order to profit from this high flexibility, one has to provide the necessary algorithmic intelligence and complexity.

Supporting Spare Resources for Multi-layer Recovery

Static multi-layer survivability involves more than just the coordination between all layers; it also involves the provision of spare resources to support the disrupted traffic. The simplest and rather expensive solution is the “double protection” where both the used and the spare capacities that are provisioned by a layer are simply protected again in the underlying layers. For instance, in a two-layer network, each working lightpath of the higher layer is first protected by means of a backup lightpath in this same layer. In addition, this higher layer working lightpath is implemented and protected in the lower layer. At its turn, the backup lightpath is also implemented and protected in this lower layer. Consequently, the required capacity is more than twice the capacity actually needed since that spare capacity is provisioned in both network layers. The added-value of protecting, and thus investing in an additional amount of spare capacity in the lower layer, is expected to be very low (*e.g.,* few exceptional multiple failure network scenarios).

A first possibility to save investment in physical capacity is to carry the backup lightpath of the higher layer network recovery scheme as unprotected traffic in the underlying network layers. This strategy, called “logical spare unprotected”, still allows protecting against any single failure. A prerequisite for such a scenario is that the lower network layer supports both protected and unprotected lightpaths. Another step towards reducing furthermore the cost of spare capacity implementation is the “common pool” strategy [J73]. This strategy consists in routing the spare capacity introduced by the higher layer in the spare capacity provided by the underlying network layer. For instance, in a two-layer network, a working lightpath of the higher layer is routed along a physical path in the lower layer. This physical path is protected in the lower layer by means of a backup path. The working lightpath is also protected by a backup lightpath in the higher layer. This backup lightpath is routed using the same spare capacity resources as the backup path. The common pool strategy provides a pool of physical spare capacity that can be used by the recovery technique in either the higher layer or the lower layer (but not simultaneously).

Restoration method may use a dedicated backup lightpath for a primary lightpath. This is known as "dedicated backup" reservation method [C12, J74]. This method has an advantage of shorter restoration time; however, this method reserves excessive resources. For better resource utilization, multiplexing techniques can be employed. In general, the probability that two primary lightpaths fail simultaneously is negligible. Consequently, their backup lightpaths can share a common wavelength channel [C13, J74]. This technique is known as "backup multiplexing".

2.2.3 Dynamic Multi-layer Recovery Scheme

Conversely to static recovery schemes that do not alter the logical network topology, dynamic multi-layer survivability strategies use logical topology modification for recovery purposes [J75, B151]. This requires real-time establishment and release of lower layer network connections that implement logical lightpaths of the higher network layer. For instance, optical networks will be enhanced with a control plane, which gives the client networks the possibility to initiate the set-up and tear-down of lightpaths through the optical network. This could be used to reconfigure the logical IP network when it is affected by a network failure.

This approach has the advantage that the spare resources in the higher network layer should not be established in advance. Thus, the underlying network layer should not care about how to treat these spare resources (double protection, logical spare unprotected, common pool). However, spare capacity still has to be provided in the lower network layer to deal with lower layer failures. Moreover, enough capacity is also needed in the lower layer to support the reconfiguration of the higher layer logical network topology and the traffic routed on that topology.

There are two possible approaches for the reconfiguration of the logical topology during a failure, namely the "global reconfiguration" option and the "local reconfiguration" option. For additional information, please refer back to Section 1.3.1 on "Logical Topology Reconfiguration".

Upon a failure, the network should be quickly reconfigured to limit the impact of traffic disruption and/or QoS degradation due to the congestion; then, disrupted connections are re-established between end nodes. The process of reestablishing communication through a lightpath between the endpoints of a failed lightpath is known as "lightpath restoration". To obtain a fast recovery mechanism, the backup lightpath should be pre-computed. Different backup lightpaths types exist depending on where they have originated or, on what types of failure/recovery notification are activated [J74]. A "link-based" method employs local detouring, while a "path-based" method employs end-to-end detouring. A link-based method reroutes traffic around the failed component. When a link fails, a new path is selected between the end nodes of the failed link. This path, in combination with the working segment of the primary path, will be used as the backup path. In the case of wavelength selective networks, the backup path must necessarily use the same wavelength as that of the primary path as its working segment is retained. In a path-based restoration method, a backup lightpath is selected between the end-points of the failed primary lightpath. This method shows better resource utilization than the link-based restoration methods. However, it requires excessive signaling and results in longer restoration time. In network scenarios where the traffic streams are very sensitive to packet losses, a "reverse backup" model is used. This method can reverse the traffic at the point of failure back to the source node via a reverse backup lightpath. Then, the traffic is carried by the recovery lightpath as in the path-based model [J76, J77].

2.3 Quality of Transmission Aware Networks

The ever-increasing demand for bandwidth capacity has witnessed a wide deployment of point-to-point WDM transmission systems which have emerged as a viable solution for the Internet's underlying technology. WDM technology has first been deployed for point-to-point transmission. Routing and switching functions were performed electrically at each network node. Hence, optical signals must go through Opto-Electrical (O/E) and Electro-Optical (E/O) conversions at each intermediate node. Consequently, a network node may not be able to process all the traffic carried by all its input signals, including the traffic intended for the node itself as well as the traffic that passes through the node to other destinations. This problem is referred to in the literature as electronic bottleneck. In order to overcome this problem, WDM systems are moving beyond point-to-point transmission systems to all-optical systems thanks to recent advances in optical technologies.

All-optical (*i.e., or transparent*) WDM networks are nowadays achievable owing to recent technology advances in optical amplifiers, optical multiplexers and demultiplexers, optical switches, and other optical devices. In all-optical networks, optical routing and switching, circuit set-up and tear-down, protection and rerouting functions are incorporated at the optical layer. All these operations are performed in the optical domain without undergoing any Optical-Electrical-Optical (O/E/O) conversion at intermediate nodes which results in potentially lower cost to build the network. Such a network is considered as a promising candidate for building the next-generation backbone network at low cost.

However, all-optical devices are not ideal. An optical signal undergoes many transmission impairments as it travels through several optical components along its route. Hence, the physical size of transparent network is mainly determined by impairments effects such as attenuation, noise, crosstalk, chromatic and polarization mode dispersions, nonlinear effects, and so on [C14]. Consequently, transmission impairments occurring in the physical layer should not be ignored in all-optical lightpath routing.

In *opaque* networks, the quality of the optical signal is always considered to be acceptable since 3R regeneration (Re-amplifying, Re-shaping, and Re-timing) is performed at each intermediate node along the route. However, the operating expenses of such a system may be quite high due to the large amount of regenerators required at the nodes of a national-scale network.

Today, *translucent* networks become a promising solution to meet the quality of transmission requirements by achieving performance measures close to those obtained by fully opaque networks at a much lesser cost. In such networks, 3R regenerators are used at intermediate nodes only when it is necessary to improve the signal budget.

2.3.1 Effects of Transmission Impairments on RWA

Many RWA problems have been investigated by assuming an ideal all-optical network where signal transmission is considered to be error-free. However, the transmission impairments may significantly affect the quality of a lightpath. Hence, they must be taken into consideration during lightpath assignment.

In particular, a lightpath may traverse through a number of non-ideal transmission devices, such as optical fibers, optical amplifiers, and optical cross-connects. Because of transparency of an optical network, noise and signal distortion due to linear and nonlinear effects accumulate along the lightpath and may cause significant signal degradation [J78]. At the destination node, the quality of the received signal may be so poor that the bit error rate can reach an unacceptable high value, and consequently the lightpath is unusable.

The transmission impairments induced by non-ideal transmission components can be classified into linear and nonlinear effects. Some important linear impairments are amplifier noise, chromatic and polarization mode dispersions, crosstalk, etc. Among nonlinear effects, we outline self-phase modulation, four-wave mixing, and scattering. Since the linear effects are independent of the signal power, their impact on end-to-end lightpath might be estimated from link parameters, and hence they may be handled as a constraint on routing [J79]. The nonlinear effects are significantly more complex because they depend on the signal power and the current wavelength allocation.

2.3.2 All-optical Impairment-Aware Routing

Without transmission impairment awareness, an RWA algorithm might provide a lightpath which cannot meet the signal quality requirements. Therefore, the control plane of a transparent optical network should incorporate the characteristics of the physical layer in setting-up a lightpath for a new connection.

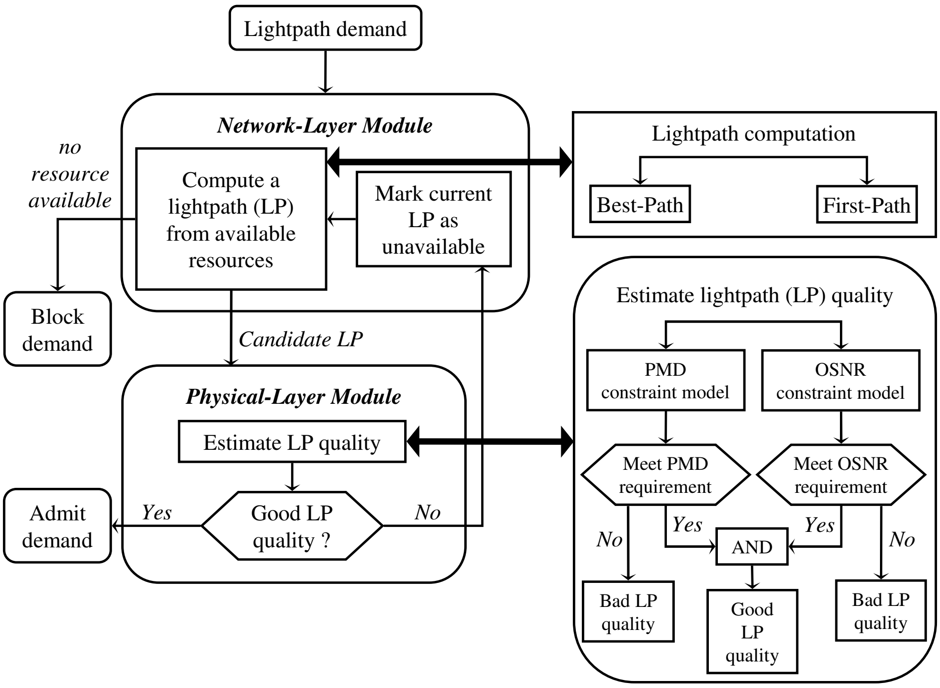


Fig. 2.2. Integrated model of impairment aware RWA algorithms

An approach to incorporate network layer and physical layer effects is to employ an impairment-aware algorithm consisting of two steps: *lightpath computation* and *lightpath verification*. Such an approach is referred to as cross-layer design. A high-level structure of such impairment-aware RWA algorithms is presented in Figure 2.2 [J80, B152]. The algorithm first uses a network layer module to look for a candidate lightpath without considering the transmission impairments for the lightpath demand. If no route or wavelength is available, the demand is blocked due to lack of resources in the network layer. This type of blocking is referred to as *network resource blocking*. If one or more routes and wavelengths are available for the candidate lightpath, the physical layer module is invoked to estimate the statistical signal quality of the candidate lightpath [J81]. If the lightpath quality is acceptable, *i.e.,* if it satisfies a certain signal quality requirement, the demand is admitted at the source node using this candidate lightpath. Otherwise, the physical layer module notifies the network-layer module to reject the candidate lightpath, and the network layer module tries to find another candidate lightpath from the available resources and wavelengths. In this way, if no available lightpath can achieve acceptable signal quality, the demand is blocked. This kind of blocking is due to poor signal quality in the physical layer and is called *physical layer blocking.*

2.3.3 Translucent Network Design

Recent researches support the idea of sparse regeneration in large-scale optical networks as an alternate architecture to the fully transparent and the fully opaque architecture. In such networks, referred to as translucent networks, sparse regeneration is used to overcome the severe physical layer blocking of a fully transparent network while using much less regenerators than in a fully opaque network. The network design and wavelength routing problem in translucent networks, also known in the literature as regenerator placement problem, has been addressed in [C16, J82, J83].

Generally, regenerators can be placed under two lightpath establishment schemes, namely the *Static Lightpath Establishment* (SLE) scheme and the *Dynamic Lightpath Establishment* (DLE) scheme [B153, J70]. In SLE schemes, all demands are fixed and should be provisioned at the same time whereas in DLE schemes, lightpath demands may arrive and leave in a dynamic manner.

The problem of regenerator placement is addressed in [C17] under the SLE scheme. The authors propose a heuristic algorithm to tackle the RWA problem while meeting the quality of transmission requirements for the established lightpaths. In their approach, the authors assume that it is possible to install a regenerator for a traffic request at any intermediate node if necessary. In this study, the quality of transmission on a given lightpath is evaluated using a prediction tool that takes into account the simultaneous effects of four transmission parameters, namely chromatic dispersion, polarization mode dispersion, nonlinear phase shift, and amplified spontaneous emission [C18].

In [J83], the authors deal with the regenerator placement problem under the DLE scheme. Due to the difficulty in precisely describing future traffic patterns, regenerator placement under DLE has to be carried out based on some kind of prediction. Furthermore, an optimal placement at one time may not remain optimal at another time. The authors propose several heuristic regenerator placement algorithms based on different knowledge of future traffic patterns.

Without specific knowledge on future network traffic, the most useful information available for regenerator placement is the network topology. Regeneration are most likely to be carried at two categories of nodes. The first category consists of those nodes that are located at the center of the network whilst the second category consists of those nodes that have a high nodal degree [J83].

When some information on future network traffic is available, a traffic-prediction based regenerator placement algorithm may be used. Considering predicted traffic pattern, a predefined wavelength routing algorithm is applied to a large number of lightpath demands. From this knowledge, the nodes where regeneration will be most likely necessary can be determined. In [J83], the authors propose two traffic-prediction based regenerator placement algorithms, which favor the nodes with heavy traffic loads and the nodes traversed by large amount of lightpaths, respectively.

Part II

Traffic Grooming and Rerouting in Multi-layer WDM Network

3

Backbone Traffic Characterization

Abstract: *In the field of network planning and traffic engineering, traffic characterization is a key parameter. This topic has been widely investigated in the context of packet-switched data networks. In this chapter, we propose a backbone traffic characterization well suited to dynamic circuit-switched optical WDM networks. From the traffic fluctuations observed on point-to-point WDM pipes in a real core network, a pragmatic approach is introduced in order to decompose this traffic into six basic traffic types. These traffic types cover the time predictability scale from the most deterministic to the most random in terms of arrival date and life duration of the requests. They also cover the different granularities of the traffic demands. An original aspect of our proposed traffic characterization consists in taking into consideration the time-zone shifts that may exist in large scale continental networks. We formulate the time correlation of the demands which has a strong impact on network resources utilization. In this chapter, we also define the various traffic scenarios that will be used in the following chapters of the thesis.*

3.1 Introduction

In recent years, the telecommunication backbone has experienced substantial growth; data traffic has increased at an unprecedented rate. Sustainable data traffic growth rate of over 100% per year has been observed since 1990. There were periods when a combination of economical and technological factors resulted in even larger growth rates, *e.g.,* 1000% increase per year in 1995 and 1996. This trend is likely to continue in the future. Simply put, more and more users are getting online, and those who are already online are spending more time online and are using more bandwidth-intensive applications. Market research[1] [2] shows that, after upgrading to a broadband connection, users spend about 35%

[1] J. P. Morgan Securities Incorporation, and McKinsey & Company, "Broadband 2001: A comprehensive analysis of demand, supply, economics, and industry dynamics in the U.S. broadband market" (New york, April 2001).

[2] J. B. Horrigan, "Home Broadband Adoption 2006: Home broadband adoption is going mainstream and that means user-generated content is coming from all kinds of internet users" (Washington, D.C.: Pew Internet & American Life Project, May 2006). Available online at: http://www.pewinternet.org [W164].

more time online than before. Voice traffic is also growing, but at a much slower rate[3]. According to most analysts, data traffic has already surpassed voice traffic. More and more subscribers telecommute and require the same network performance as they see on corporate LANs. More services and new applications become available as bandwidth per user increases. Understanding the variability of Internet traffic in backbone networks is crucial to fundamental operational tasks like capacity planning, traffic engineering, and peering management. Router vendors, third parties, academic researchers, and ingenious network engineers have devised multiple ways of collecting and estimating traffic matrices. A traffic matrix describes the volume of data traffic transmitted between every source-destination node pair in the network. The traffic matrices can refer either to instantaneous traffic flows or to averaged traffic flows. When used together with routing information, the traffic matrix gives the network operator valuable information about the current network state. Traffic engineering, network management, and network provisioning are based on the knowledge of such information.

3.2 Background

Depending on the problem being addressed, various design models can be used to represent the traffic in a network. In network planning problems, long term traffic forecasts are represented by traffic matrices. In this case, a traffic matrix is a deterministic static representation of the expected spatial traffic distribution in the network for a given time interval in the future. Instead of using a single static traffic matrix to characterize the traffic requirement, it is also possible to describe it by a set of traffic matrices. In this way, we can outline the dynamic evolution of the network load. The traffic pattern may change within this matrix set over a period of time, say throughout a day or a month, and is referred to, in the literature, as Multi-Hour Traffic Matrices (MHTM) [C19, C20, J84]. The network needs to be reconfigured when the traffic pattern transits from one matrix to another.

In traffic engineering problems, the traffic is usually characterized by stochastic processes that capture its dynamics and randomness. One of the first approaches to characterize such traffic flows goes back to 1937 [J85]. The suggested method estimates point-to-point traffic demands in a telephone network based on a prior traffic matrix and measurements of incoming and outgoing traffic. Erlangian traffic models have been widely adopted in telephone networks. With the emergence of Internet Protocol (IP) networks, new traffic models have been required since the mid 1990's.

An important statistical difference between the telephone network and the Internet comes in the distribution of the length of the connections. The exponential distribution has provided a useful workhorse model for the telephone network, perhaps because human choice determines the duration of a phone call. But this exponential distribution is very inappropriate for the durations of Internet connections. Many investigations have outlined the limits of the Poisson model especially for Transfer Control Protocol/Internet Protocol (TCP/IP) connections [J86, J87, J88]. In the context of packet switched networks, the concept of long-range dependent traffic (also called auto-similar traffic) has been introduced as a more accurate traffic characterization [J89, J90]. Thank to this representation, the congestion control efficiency has been considerably improved [J91, J92].

[3] The demand for voice traffic is increasing as the number of users of personal communications devices increases. The number of users of such devices (*i.e.,* cellular telephones, beepers, etc.) is increasing each year. In addition, many third world countries are beginning to modernize their aging communications infrastructures by replacing outdated systems and/or installing new systems.

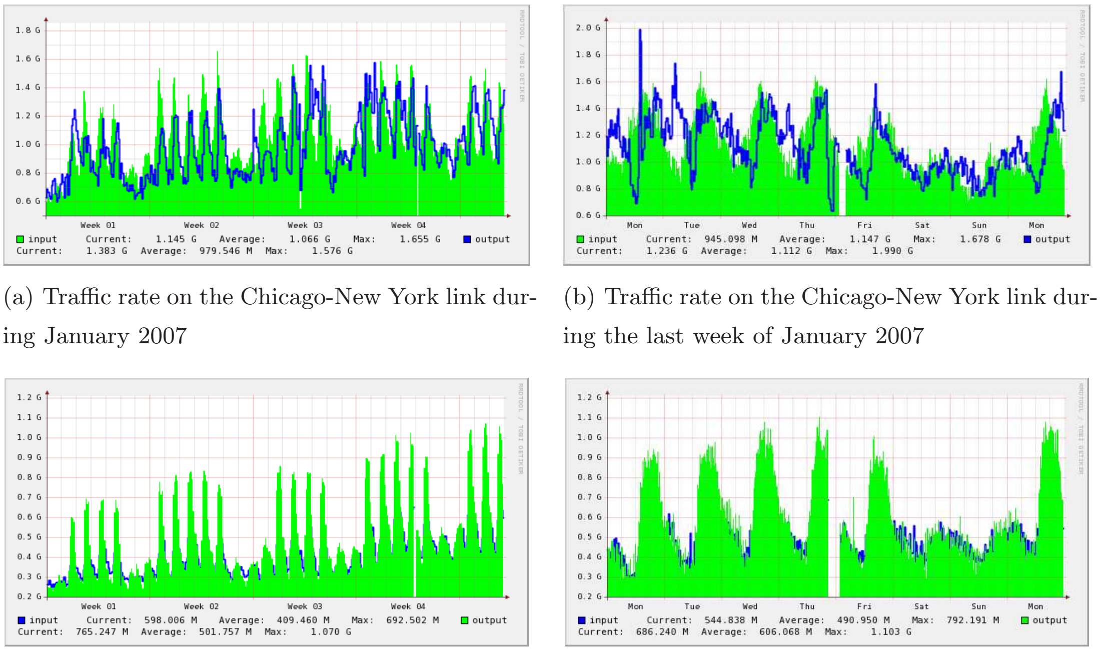
Up to now, optical WDM networks were considered as circuit oriented. In this perspective, a novel approach has been introduced in 2002 by Kuri *et al.* and is referred to as Scheduled Lightpath Demand (SLD) [C21, C22]. The proposed SLD traffic model is both dynamic and deterministic since it deterministically captures the time and space distribution of the traffic demands in the network. The SLD and the MHTM models are equivalent in that a set of SLDs may be represented by the MHTM model. However, the SLD model allows a more flexible modeling of individual lightpath properties (*e.g.,* route, wavelength, etc.) than the MHTM model. The SLD approach can be viewed as an intermediate approach between network planning and traffic engineering. Under network planning, the global knowledge of the traffic demands enables the use of global optimization tools for Routing and Wavelength Assignment (RWA). Under traffic engineering, the dynamics and the randomness of the demands impose an online RWA. The RWA must be carried out on the fly at the arrival of a new connection request. Only local optimization tools may be adopted for RWA in this context. Thanks to the predictability of the SLDs, it is possible to deal with dynamic traffic by means of global optimization tools for RWA problems [D156].

3.3 Data Measurement and Analysis

In this section, we present measurement results carried out on the Abilene backbone network. Abilene is the Internet2 backbone network [W165]. It has 11 nodes and spans the continental USA. The traffic on Abilene is non-commercial, arising mainly from the major universities in the USA. For this network, we have focused on sequence of data traffic collected during approximately one month interval (January 2007). We assume that such a time interval is sufficient to develop an accurate and current view of backbone traffic. We present results from two different links: the Chicago-New York link and the Seattle-Denver link. These links were chosen to cover nodes with different geographical areas and different time-zone shifts. Figure 3.1(a) shows a one month sample of the data stream (from January 1st to February 1st 2007) measured on the Chicago-New York link, while Figure 3.1(c) shows a sample of the data stream measured on the Seattle-Denver link during the same period of time. In Figure 3.1(b) and Figure 3.1(d), we focus on the last week of the data measurements (from January 22nd to 30th 2007) shown in Figure 3.1(a) and Figure 3.1(c), respectively. The profiles plotted on each of the sub-figures correspond to the measurements of the traffic flow on both directions of the link.

IP traffic is very dynamic, but in a backbone network, due to traffic aggregation, this dynamism becomes more predictable following a well-shaped daily profile. Figure 3.1 illustrates the daily and weekly variations in the traffic. One may observe the similarity between consecutive weeks of data. Daily peaks are visible between 9 am and 5 pm. On the weekend, the traffic decreases significantly. The same behavior is observed on all links with variations on peak height, duration, and hours depending on the geographic location and the nature of the traffic. The following observations are of interest:

- We observe distinct weekly and diurnal patterns. From Monday to Friday, the traffic surges during the busy hours, and the load comes down significantly at night. The day-to-night traffic ratio is about 5:1 to 7:1. On the weekend, the traffic load is significantly reduced compared to weekdays and does not exhibit clear patterns. For example, at a university, a larger traffic is generated during business days than during days off. An additional dimension of dynamism is the traffic shifts based on time-zones. As a result, the traffic load on the various links does not increase/decrease during the same time intervals. This occurs since the work hours of the different time-zones do not coincide.



(a) Traffic rate on the Chicago-New York link during January 2007

(b) Traffic rate on the Chicago-New York link during the last week of January 2007

(c) Traffic rate on the Seattle-Denver link during January 2007

(d) Traffic rate on the Seattle-Denver link during the last week of January 2007

Fig. 3.1. Traffic rate in the Abilene backbone network links for different time intervals

- In Figure 3.1, we see occasional peaks in traffic load. There can be many causes behind such peaks: Denial-of-Service (DoS) attacks, routing loops, and bursty traffic.
- Traffic on a bidirectional link is often asymmetric. This traffic asymmetry results from the nature of the applications. Many applications, such as web and ftp, are inherently asymmetric. The data sent from the end-users to the server farms corresponds to small messages, whereas large web files are transferred in the opposite direction.

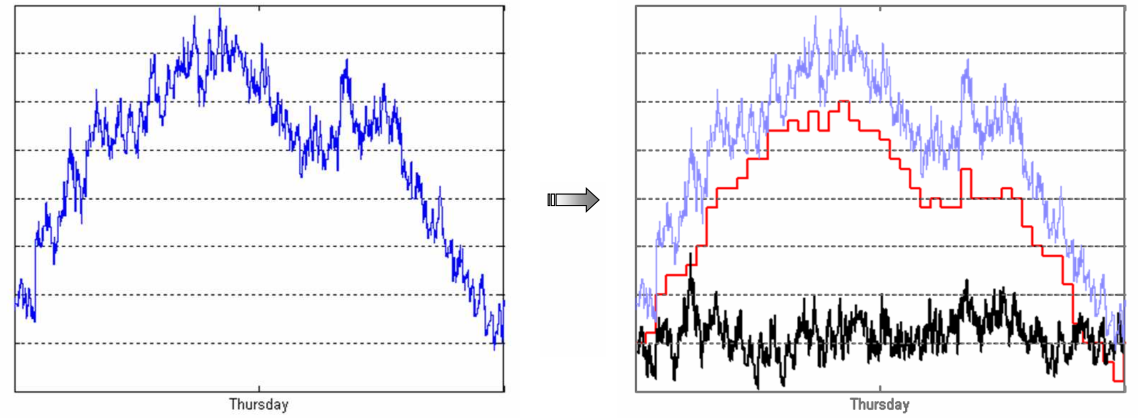
3.4 Traffic Modeling

Much of the works in network traffic analysis so far [C23, C24, C25, J87, J93] has focused on studying traffic on a single link in isolation. However, a wide range of important problems faced today by network researchers require modeling and analyzing the traffic on all the links simultaneously. These research problems include traffic engineering [C26], traffic matrix estimation [C25, C27, C28], anomaly detection [C29], attack detection [C30], traffic forecasting, and capacity planning [C31]. Unfortunately, whole-network traffic analysis (*i.e.,* modeling the traffic on all the links simultaneously) is a difficult objective amplified by the fact that modeling the traffic on a single link is itself a complex task. Whole-network traffic analysis therefore remains an important and unmet challenge. One way to address the problem of whole-network traffic analysis is to recognize that the traffic observed on different links of a network is not independent, but is in fact determined by a common set of underlying source-destination flows and a routing matrix [C32, J94]. A source-destination flow is the collection of all traffic that enters the network from a common ingress point and departs from a common egress point. The superposition of these point-to-point flows, as determined by routing, gives rise to the traffic transported on all the links of the network. Thus, instead of studying the traffic on all the links, a more direct and fundamental focus for whole-network traffic study is the analysis of the network's set of source-destination flows.

Through analyzing traffic measurement data, the dynamics of the network can be characterized by both daily and weekly periodic components, as well as longer-term variations. Shorter time-scale stochastic variations are superimposed on top of these components. As a result, each source-destination flow can be expressed as the sum of different traffic requests. These traffic requests fall into three natural classes:

1. deterministic traffic requests, which capture the predictable periodic trends in the source-destination flow
2. peak traffic requests, which capture the occasional short-lived bursts in the source-destination flow
3. noise traffic requests, which account for traffic fluctuations appearing to have relatively time-invariant properties across all the source-destination flows

This taxonomy can be viewed as being parallel to the characteristics observed in various analysis of network traffic in the literature: periodic trends [C24, C31], stochastic bursts [C33], and fractional Gaussian (or other) noise [J86, J90]. This broad characterization of the nature of traffic requests provides a useful basis for organizing and interpreting studies of the whole-network traffic. In the following, we have segmented the source-destination traffic flows into a regular predictable component referred to as *Scheduled Demands* (SxDs) and a stochastic component referred to as *Random Demands* (RxDs). The *semi-Random Demands* (sRxDs) are another class of stochastic component that requires higher priority services than the RxDs. Figure 3.2(a) illustrates an instance of a source-destination traffic flow between two nodes in the network during a one day period. Figure 3.2(b) shows how this traffic pattern can be decomposed into a deterministic component (step function) and a stochastic component (noise-like function).



(a) Source-Destination traffic flow

(b) Decomposition into predictable and stochastic components

Fig. 3.2. Segmenting of a given source-destination traffic flow into predictable and stochastic components

All three classes of traffic requests are dynamic requests that hold the network for a given period of time. As a result, the i^{th} connection of any class of traffic requests is represented by the tuple $(s_i, d_i, \alpha_i, \beta_i, n_i)$ where s_i and d_i are the source and the destination nodes of the demand, α_i and β_i are its set-up and its tear-down dates, and n_i is the traffic rate requested by the connection.

3.4.1 Scheduled Demands (SxDs)

The periodic source-destination traffic flows are represented by a set of dynamic but deterministic traffic requests referred to as *Scheduled Demands* (SxDs). Being periodic/deterministic, the SxDs are known in advance and are routed mainly by means of the management plane using an off-line global optimization tool. The centralized off-line tool simultaneously examines the global network topology and the set of traffic requests in order to optimize the path and the wavelength assigned to each demand. By taking advantage of the space-time correlation between the SxDs, the off-line tool dimensions a network able to handle these traffic requests at the lowest cost. The output of the off-line calculation is the minimal number of electrical ports and optical ports required to transport the set of SxDs. As a result, such connection requests will benefit from the reservation of a given route before their instant of arrival. Hence, the SxDs may correspond to traffic requests that require guaranteed Quality of Service (QoS). Examples of such guaranteed services are:

- setting the right optical links for the provision of grid computing services
- providing an Optical Virtual Private Network (OVPN)
- providing High-Definition TeleVision (HDTV) multicast for a remote medical operation
- providing periodic network backup capabilities without the need for local tapes (Storage Area Network (SAN)).

Based on our observations on the Abilene backbone network [W165], we have noticed the diurnal pattern of the traffic flows. In addition, the traffic load reaches some local maxima during working hours. Because the working hours at the source node area and those at the destination node area may not coincide, the peaks of the traffic flows do not occur in the same time intervals. As a result, the time-zone difference between the geographical areas of the source node and the destination node has a significant impact on the shape of the traffic flow. These time-zone shifts have been taken into account in our traffic characterization. Depending on the possible values of the time shift, we propose four different traffic shapes represented in Figure 3.3. It is to be noted that, due to the traffic diversity between different source-destination traffic flows, the real traffic shape falls within a range of ±10% of this hypothetical shape. All the plots correspond to a unitary average traffic flow. In other words, if $r_i(t)$ stands for the instantaneous data rate in bit-per-second on the plot i, ($i \in \{a, b, c, d\}$ of Figure 3.3), we have:

$$\frac{1}{24 \times 3600} \int_0^{24 \times 3600} r_i(t) dt = 1, \qquad \forall i \in \{a, b, c, d\} \tag{3.1}$$

In order to obtain the amount of the traffic flow circulating between two nodes, these plots must be weighted by the probability that the source node sends a flow to the destination node. This probability assumes a proportionality relation between the traffic entering the network at node i and destined to node j, the total amount of traffic entering at node i, and the total amount of traffic leaving the network at node j. As the source-destination traffic flow is the result of the activity of distinct user populations, the traffic entering/leaving the network at a node is proportional to the number of citizens residing in the area surrounding the physical location of this node. Let p_i be a weight proportional to the number of citizens in the area covered by node i. These weights p_i are integer numbers ranging from 1 to 10, and they represent the capacity of a node to send/receive data. Let us assume that the network can globally carry an average traffic flow of R_T (in *Tbps*). This traffic is spread over all the network nodes according to the weights assigned to each of them. Each fraction of this traffic represents the inbound traffic at a particular node. For instance, the average traffic flow φ_i generated

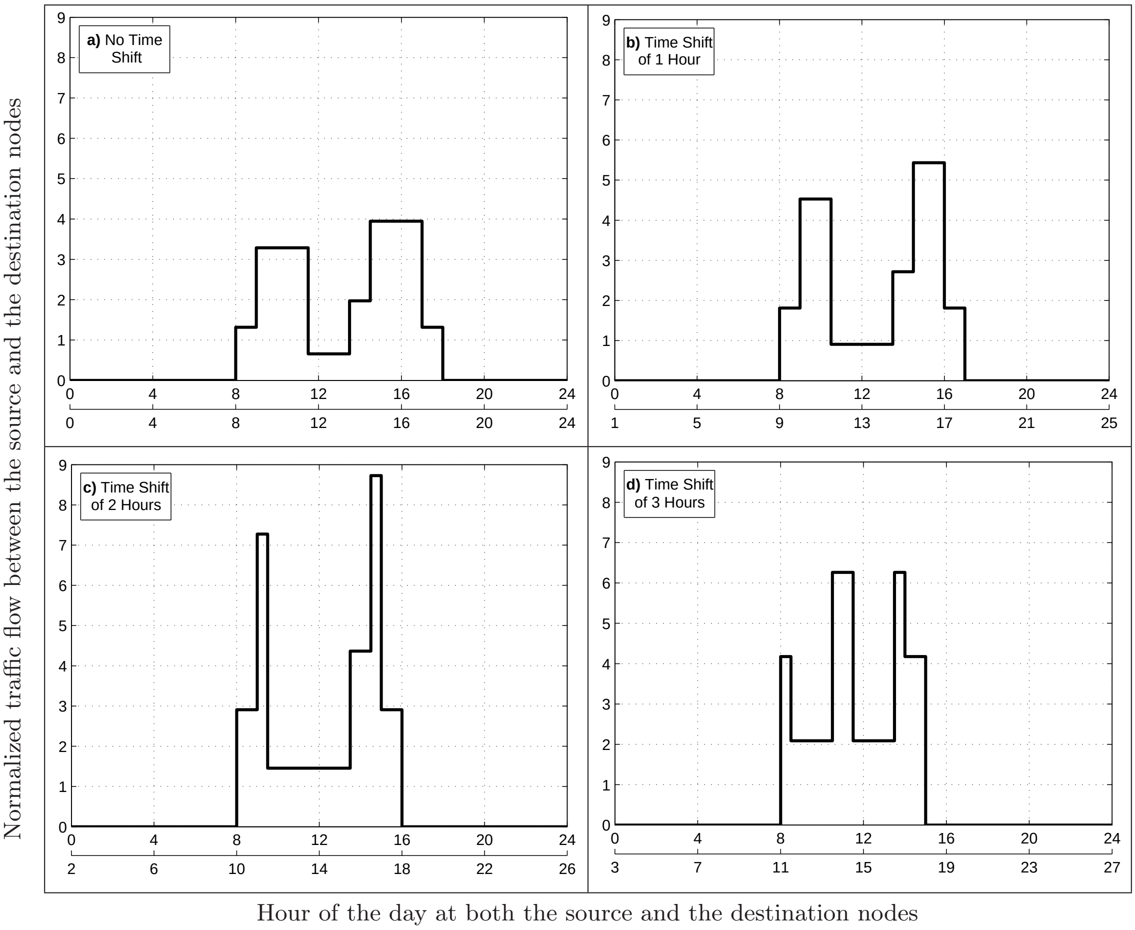


Fig. 3.3. Different shapes of the traffic load between source-destination nodes for 0, 1, 2 and 3 hours of time-zone shift respectively

at node i is equal to:

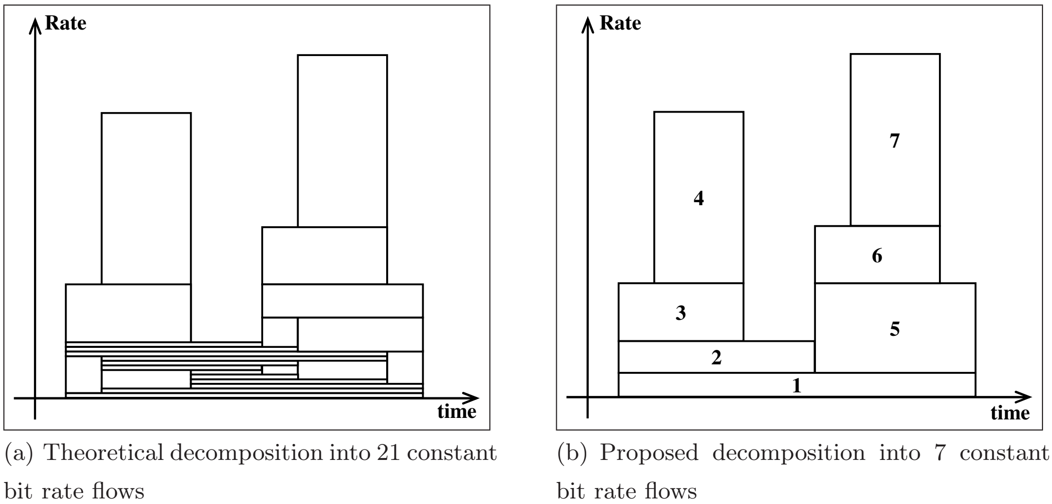
$$\varphi_i = \frac{p_i}{\sum\limits_k p_k} \times R_T \tag{3.2}$$

This inbound traffic is forwarded to the other nodes according to the weight of the receiving node. As a result, the traffic flow $\varphi_{i,j}$ forwarded from node i to node j ($j \neq i$) is equal to:

$$\begin{aligned}\varphi_{i,j} &= \frac{p_j}{\sum\limits_{k \neq i} p_k} \times \varphi_i \\ &= \frac{p_i \times p_j}{\sum\limits_{l,k \neq i} p_l \times p_k} \times R_T \end{aligned} \tag{3.3}$$

It is clear that the traffic flow between any couple of nodes is asymmetric ($\varphi_{i,j} \neq \varphi_{j,i}$).

At this level, we have already defined the shape of the traffic flow between any two nodes based on the time-zone shift between the areas surrounding these nodes. We have defined as well the amount of the traffic volume that must be exchanged between these two nodes based on the number of citizens in their surrounding areas. It still remains to decompose this traffic flow into a set of SxD requests. By definition, a traffic request has a constant rate during its service time. Hence, the traffic shape must be decomposed into fixed data rate demands. Theoretically, each shape can be decomposed into a set of 21 constant bit rate flows when considering all the time instances of traffic variations (up/down edges of the traffic shape). This theoretical decomposition is shown in Figure 3.4(a). To keep the problem at an acceptable complexity, we propose to limit our traffic flow decomposition to 7 constant bit rate flows. Figure 3.4(b) illustrates an example of such decomposition. Each constant bit rate flow is, in its turn, divided into multiple Scheduled Demands SxDs. Depending on the value of the required rate of an SxD, we can distinguish between two sub-classes of demands: Scheduled Lightpath Demands (SLDs) and Scheduled Electrical Demands (SEDs). The former are supposed to carry a traffic rate C_ω (*bps*) equal to that of a WDM channel, while the latter are supposed to carry a traffic rate equal to a fraction of C_ω.



(a) Theoretical decomposition into 21 constant bit rate flows

(b) Proposed decomposition into 7 constant bit rate flows

Fig. 3.4. Decomposition of source-destination traffic flow into constant bit rate flows

Example 3.1.

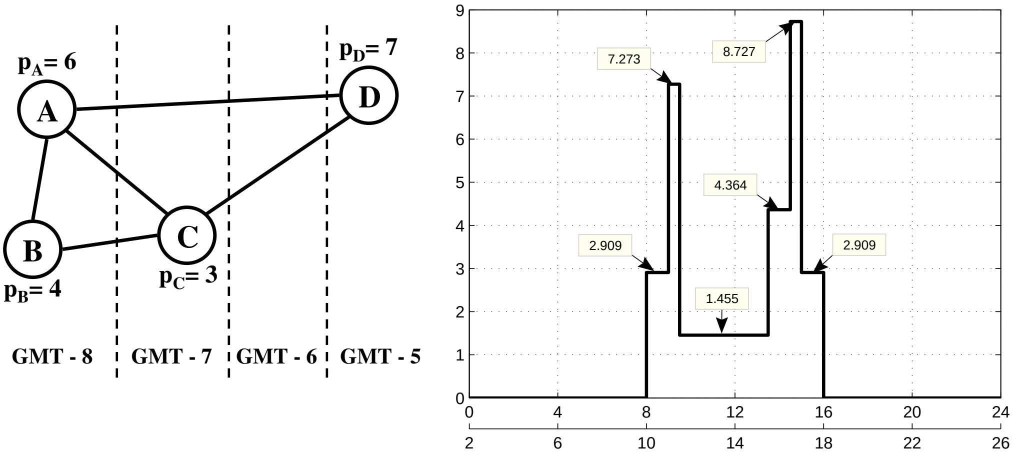
Let us consider a small network of 4 nodes and 5 links as represented in Figure 3.5(a). These nodes are located in different time-zones and they have various weights. We assume that the traffic flow averaged over the whole network is evaluated to $R_T = 1\ Tbps$. The average source-destination traffic flow between each pair of nodes is given by Table 3.1. For instance, let us detail the traffic flow from node C to node D. The inbound traffic at node C has a value of:

$$\varphi_C = \frac{3}{20} \times 1\ Tbps = 0.15\ Tbps$$

The average traffic flow from node C to node D is given by:

$$\varphi_{C,D} = \frac{7}{17} \times \varphi_C = 0.0618\ Tbps$$

The time-zone shift between the node C and the node D is of two hours. Hence, the shape of the traffic flow between these two nodes is the one described by Figure 3.5(b). As we mentioned earlier, this hypothetical plot was designed to match the working hours in the regions surrounding the two nodes. The value of the steps are computed in such a way to insure a unitary average traffic flow between the two nodes (*c.f.* Equation (3.1)). Noting that the average value of the data flow from node C to node D is of 0.0618 *Tbps*, this shape must



(a) 4-node and 5-link network sample (b) Shape of the traffic flow from node C to node D

Fig. 3.5. An example of SxD traffic generation for a 4-node and 5-link sample network

Table 3.1. Average source-destination traffic flow between any two nodes

Src *Dst*	**A**	**B**	**C**	**D**
A	0	0.075	0.0529	0.1615
B	0.0857	0	0.0353	0.1077
C	0.0643	0.0375	0	0.0808
D	0.150	0.0875	0.0618	0
Inbound Traffic	0.3	0.2	0.15	0.35

be scaled to match this average. Finally, the real shape of the source-destination traffic flow falls within ±10% of this hypothetical shape. Table 3.2 summarizes the step values of the hypothetical source-destination shape as well as those of the real source-destination shape of the traffic flow from node C to node D.

Table 3.2. Hypothetical and real shapes of the traffic flow from node C to node D

Time Interval	**Hypothetical bit rate** (Tbps)	**Real bit rate** (Tbps)
8h00 → 9h00	0.18	0.192
9h00 → 9h30	0.449	0.438
9h30 → 13h30	0.09	0.093
13h30 → 14h30	0.27	0.2748
14h30 → 15h00	0.539	0.5894
15h00 → 16h00	0.18	0.1933

It is to be noted that all the time instances cited in Table 3.2 are measured with respect to the time-zone of node C. If we choose the time-zone of node A as a time reference for the whole network, these time instances must be shifted by one hour (the relative time-zone shift of node C with respect to node A). Finally, we decompose this source-destination traffic flow into 7 constant bit rate traffic flows. Assuming that the nominal capacity of a wavelength channel is equal to 10 Gbps, Table 3.3 shows the decomposition of these constant bit rate traffic flows into subsets of SLDs and SEDs. For this decomposition we have chosen the node A as the time reference for the whole network.

Table 3.3. Decomposition of the traffic flow from node C to node D into a set of SLDs and SEDs with node A as time reference

	Set-up time	**Tear-down time**	**Rate**	**SLD**	**SED**
	hh:mm	hh:mm	(Gbps)	(WDM channel)	(%)
1	7:00	15:00	40.9	4	9
2	7:00	12:30	52.1	5	21
3	7:00	8:30	99.0	9	90
4	8:00	8:30	246.0	24	60
5	12:30	15:00	152.4	15	24
6	12:30	14:00	81.5	8	15
7	13:30	14:00	314.6	31	46

Since the precise details of the traffic are considered proprietary, no further details on the exact traffic characterization will be disclosed at this time. However, it is to be noted that this traffic characterization has been evaluated and validated by means of statistical measures provided by Alcatel on real networks.

3.4.2 Random and semi-Random Demands (RxDs/sRxDs)

The stochastic source-destination traffic flows are represented by a set of dynamic and completely random traffic requests referred to as *Random Demands* (RxDs). The RxDs arrive one at a time at the network and need to be routed on the fly. The path computing or path selection of the RxDs is carried out by the control plane. The control plane operates in real time and computes a route for a given RxD considering the requirement of this demand as well as the constraints imposed by the current network state. Such routing is mainly based on a distributed online sequential optimization tool that proceeds automatically to the grooming of low data rate traffic requests into grooming lightpaths. This online sequential algorithm must be able to also handle the routing information of the scheduled demands. Thus, an RxD can use some SxDs resources that remain inactive during the life time of the RxD. One assumes that GMPLS signaling, still under specification, is used to share information between the network nodes as well as between the control and the management planes. Best-effort Internet traffic requests are an example of RxDs.

While an SxD is a preplanned demand with pre-determined date of arrival, life duration, and capacity, an RxD corresponds to a connection request that arrives randomly and, if accepted, holds the network for an arbitrary period of time. In order to introduce a compromise between the determinism of the SxDs and the randomness of the RxDs, a new class of traffic requests has been introduced. This new class is referred to as *semi-Random Demands* (sRxDs) and can be viewed as a class of stochastic source-destination traffic flows that require higher priority services than the RxDs. An sRxD corresponds to a connection request that arrives randomly in the network but that requires network resources for a given/known period of time. Like the RxDs, the sRxDs are also routed by the control plane using an online sequential optimization tool. Depending on the value of their required rate, both RxDs and sRxDs can be divided into two sub-classes of traffic requests. The RxDs are classified into Random Lightpath Demands (RLDs) and Random Electrical Demands (REDs), while the sRxDs are classified into semi-Random Lightpath Demands (sRLDs) and semi-Random Electrical Demands (sREDs). The RLDs/sRLDs are supposed to carry a traffic rate C_ω (bps) equal to that of a WDM channel, while the REDs/sREDs are supposed to carry a traffic rate equal to a fraction of C_ω.

The arrival time of both sRLDs and sREDs follows a Poisson process of parameter λ and $\lambda'(sec^{-1})$, respectively, while the holding-time of these requests follows a negative Exponential distribution of parameter δ and $\delta'(sec^{-1})$, respectively. The source and the destination of these traffic requests are chosen uniformly among the network's nodes. The sRLDs are supposed to carry a traffic rate C_ω (in bps) equal to that of a WDM channel. In a one hour period, an average of $3600 \times \lambda$ new sRLD requests are generated. Each request requires a rate of C_ω during a time period of $1/\delta$. Thus, the average amount of data that is carried by such request is equal to C_ω/δ. A resulting data stream of $(3600 \times \lambda \times C_\omega)/\delta$ is carried by the network during one hour. Let ϕ_{sRLD} (in bps) be the global average data rate carried by the set of sRLDs.

$$\phi_{sRLD} = \frac{\lambda \times C_\omega}{\delta} \tag{3.4}$$

On the other hand, the sREDs are supposed to carry a traffic rate chosen uniformly in the interval $]0, C_{max}]$ ($C_{max} \leq C_\omega$). In order to optimize the network resources utilization, the electrical flow inherent to multiple sREDs can be multiplexed onto a so-called Grooming Lightpath (GL). A GL is a direct connection between two nodes (which are not necessarily adjacent) acting as a logical one-hop link. In a one hour period, an average of $3600 \times \lambda'$ new sRED requests are generated. One assumes that each sRED requires an average rate of $C_{max}/2$ during a time period of $1/\delta'$. Thus, the average amount of data that is carried by such request is equal to $C_{max}/(2 \times \delta')$. A resulting data stream of $(1800 \times \lambda' \times C_{max})/\delta'$ is carried by the network in a one hour period. Let ϕ_{sRED} (in bps) be the global average data rate carried by the set of sREDs.

$$\phi_{sRED} = \frac{\lambda' \times C_{max}}{2 \times \delta'} \tag{3.5}$$

It is to be noted that RxDs are generated with the same arrival date, source, destination, and capacity as the sRxDs. However, their life duration is ignored at their instant of arrival and is then assumed to be infinite. Table 3.4 summarizes the characteristics of the different types of traffic requests.

Table 3.4. Characteristics of the different sets of traffic requests

Request Type	**Source Node** s_i	**Destination Node** d_i	**Set-Up Date** α_i	**Tear-Down Date** β_i	**Traffic Rate** n_i
SLD	Weighted distribution[1]	Weighted distribution[1]	According to traffic shape [2]	According to traffic shape [2]	C_ω[3]
SED	Weighted distribution[1]	Weighted distribution[1]	According to traffic shape [2]	According to traffic shape [2]	According to traffic shape [2]
sRLD / RLD	Uniformly distributed	Uniformly distributed	$\alpha_i \rightsquigarrow Poisson(\lambda)$	$\beta_i = \alpha_i + \theta_i$ $\theta_i \rightsquigarrow Exponential(\delta)$	C_ω[3]
sRED / RED	Uniformly distributed	Uniformly distributed	$\alpha_i \rightsquigarrow Poisson(\lambda')$	$\beta_i = \alpha_i + \theta_i$ $\theta_i \rightsquigarrow Exponential(\delta')$	Uniformly in $]0, C_\omega$[3]$[$

[1] Each network node is assigned a weight p proportional to the number of citizens surrounding its geographical location

[2] Obtained by the decomposition of the traffic shapes into constant bit rate traffic flows (Figure 3.3)

[3] C_ω is the capacity of a WDM channel

3.5 Traffic Metrics

In order to evaluate the global load carried by a set of traffic requests, we have introduced 4 different traffic metrics. Let Y be a set of traffic requests $\big\{\delta_i(s_i, d_i, \alpha_i, \beta_i, n_i)\big\}$ generated during the simulation period. Y can be either SLD, SED, sRLD, sRED, RLD, or RED. We define:

1. the number N_Y of traffic requests in this set.
2. the average traffic flow ϕ_Y (in bps) of the requests in this set. Let $\phi_{i,j}$ denotes the average traffic flow between source node i and destination node j evaluated during the simulation period. Therefore, ϕ_Y is the sum of the average flow $\phi_{i,j}$ for all possible node pairs (i, j).
3. the overall traffic peak π_Y (in bps) of the requests in this set. Let $\phi_{i,j}(t_0)$ denotes the traffic flow between source node i and destination node j at instant t_0. We define $\pi_Y(t_0)$ as the sum of the flow $\phi_{i,j}(t_0)$ for all possible node pairs (i, j). Hence, π_Y is the maximum value of $\pi_Y(t_0)$ for any instant t_0 during the simulation period.
4. the time correlation $\mathcal{C}_Y$ between the requests in this set. It represents the time disjointness between the requests of the traffic set. Thus, $\mathcal{C}_Y$ has a significant impact on the reachable degree of channel sharing. Indeed, the smaller the time correlation, the greater the time disjointness, and the greater the possibility of channel sharing.

In [J95], the authors have already introduced the notion of time correlation between the SLDs. It is computed as follows:
Let $\mathcal{E}$ be the ordered set grouping the arrival times and the ending times of all the requests in the set Y. Because some requests may have the same starting/ending time, the number $\mathbb{E}$ of time instant in the set $\mathcal{E}$ ($\mathbb{E} = |\mathcal{E}|$) is less than $2 \times N_Y$.

$$\begin{aligned}\mathcal{E} &= \left(\bigcup_{i=1}^{N_Y} \alpha_i\right) \bigcup \left(\bigcup_{i=1}^{N_Y} \beta_i\right)\\ &= \{\epsilon_1, \epsilon_2, \cdots, \epsilon_{\mathbb{E}}\} \text{ such as } \epsilon_1 < \epsilon_2 < \cdots < \epsilon_{\mathbb{E}}\end{aligned} \tag{3.6}$$

Let $\mathcal{B}_q$ be the set of SLD index j such that the SLD δ_j is active at least over time period $[\epsilon_q, \epsilon_{q+1}]$

$$\mathcal{B}_q = \Big\{j \in \{1, \cdots, N_Y\} \backslash [\epsilon_q, \epsilon_{q+1}] \subseteq [\alpha_j, \beta_j]\Big\} \qquad q = 1 \cdots \mathbb{E} - 1 \tag{3.7}$$

The time correlation of this set of SLD requests is defined as:

$$\mathcal{C}_Y = \frac{\displaystyle\sum_{q=1}^{\mathbb{E}-1} \sum_{j \in \mathcal{B}_q \backslash |\mathcal{B}_q| > 1} n_j \times (\epsilon_{q+1} - \epsilon_q)}{\displaystyle\sum_{i=1}^{N_Y} n_i \times (\beta_i - \alpha_i)} \tag{3.8}$$

Note that only index sets with cardinality $|\mathcal{B}_q| > 1$ must be considered in the numerator's sum since they correspond to the overlapping time intervals of at least 2 SLDs. A value $\mathcal{C}_Y$ $(0 \leqslant \mathcal{C}_Y \leqslant 1)$ close to 0 means weak time correlation and a value close to 1 means strong time correlation.

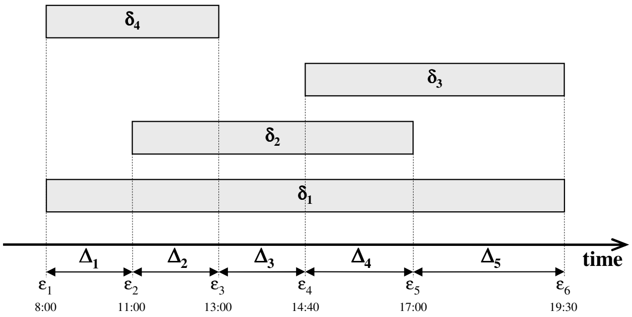
Example 3.2.

Let us compute the time correlation between the requests of a set Y composed of $N_Y = 4$ SLDs $\big(Y = \{\delta_1, \delta_2, \delta_3, \delta_4\}\big)$. The characteristics of the 4 SLDs are given by Table 3.5. For this set of traffic requests, the set of arriving/ending times $\mathcal{E}$ is defined as $\mathcal{E} =$

$\{480, 660, 780, 880, 1020, 1170\}$ (time instant being expressed as minutes since midnight). Let Δ_i be the length of the time interval $[\epsilon_i, \epsilon_{i+1}]$ ($\Delta_i = \epsilon_{i+1} - \epsilon_i,\ \forall i = 1 \cdots \mathbb{E} - 1 = 1 \cdots 5$). The time diagram of the 4 SLDs, the time intervals Δ_i, and the sets $\mathcal{B}_i$ of all active SLDs during the time interval $[\epsilon_i, \epsilon_{i+1}]$ are given in Figure 3.6.

Table 3.5. Characteristics of the 4 SLDs

δ_i	s_i	d_i	α_i	β_i	n_i
δ_1	2	8	8:00	19:30	$1 \times C_\omega$
δ_2	3	7	11:00	17:00	$1 \times C_\omega$
δ_3	1	6	14:40	19:30	$1 \times C_\omega$
δ_4	3	5	8:00	13:00	$1 \times C_\omega$



i	1	2	3	4	5
Δ_i	180	120	100	140	150
$\mathcal{B}_i$	{1,4}	{1,2,4}	{1,2}	{1,2,3}	{1,3}

Fig. 3.6. Associated time diagram of the 4 SLDs

For this set of SLDs, the time correlation $\mathcal{C}_Y$ is equal to:

$$\begin{aligned}
\mathcal{C}_Y &= \frac{\displaystyle\sum_{q=1}^{\mathbb{E}-1} \sum_{i \in \mathcal{B}_q \backslash |\mathcal{B}_q| > 1} n_i \times (\epsilon_{q+1} - \epsilon_q)}{\displaystyle\sum_{i=1}^{N_Y} n_i \times (\beta_i - \alpha_i)} \\
&= \frac{2 \times 180 + 3 \times 120 + 2 \times 100 + 3 \times 140 + 2 \times 150}{690 + 360 + 290 + 300} \\
&= \frac{1640}{1640} = 1.0
\end{aligned}$$

Being equal to 1.0, this time correlation states that the requests are strongly correlated between them. This result is quiet surprising because one does not expect to have such a unitary value for the time correlation unless the requests are permanent lightpath demands or all the requests have the same starting and the same ending times. This example motivates us to propose an enhanced method to compute the time correlation of a set of requests. This enhanced time correlation $\mathfrak{C}_Y$ is based on the correlation between the requests considered two by two. If $\Delta_{i,j}$ is the length of the time interval where both the requests i and j are active, the time correlation $\mathfrak{C}_{i,j}$ among the requests i and j is defined as:

$$\mathfrak{C}_{i,j} = \mathfrak{C}_{j,i} = \frac{2 \times \Delta_{i,j}}{(\beta_i - \alpha_i) + (\beta_j - \alpha_j)} \tag{3.9}$$

Thus, the overall enhanced time correlation is defined as:

$$\begin{aligned}
\mathfrak{C}_Y &= \frac{1}{N_Y \times (N_Y - 1)} \times \sum_{i=1}^{N_Y} \sum_{j=1 \backslash j \neq i}^{N_Y} \mathfrak{C}_{i,j} \\
&= \frac{4}{N_Y \times (N_Y - 1)} \times \sum_{i=1}^{N_Y} \sum_{j=i+1}^{N_Y} \frac{\Delta_{i,j}}{(\beta_i - \alpha_i) + (\beta_j - \alpha_j)}
\end{aligned} \tag{3.10}$$

The enhanced time correlation gives a more accurate measure of the disjointness between the requests. As it is for the time correlation $\mathcal{C}_Y$, a value of the enhanced time correlation $\mathfrak{C}_Y$ close to 0 means weak time correlation and a value close to 1 means strong time correlation.

Example 3.3.

Let us go back to the traffic set already introduced in the previous Example 3.2. For example, the requests δ_2 and δ_3 are active simultaneously between 14:40 and 17:00 ($\varDelta_{2,3} = 1020 - 880 = 140$ (time instant being expressed as minutes since midnight)), thus the time correlation between these two requests is equal to:

$$\begin{aligned}\mathfrak{C}_{2,3} = \mathfrak{C}_{3,2} &= \frac{2 \times \varDelta_{i,j}}{(\beta_i - \alpha_i) + (\beta_j - \alpha_j)} \\ &= \frac{2 \times 140}{360 + 290} = 0.4307\end{aligned}$$

The set of all the time correlation between the requests considered two by two is given by Table 3.6.

Table 3.6. Time correlation of the requests considered two by two

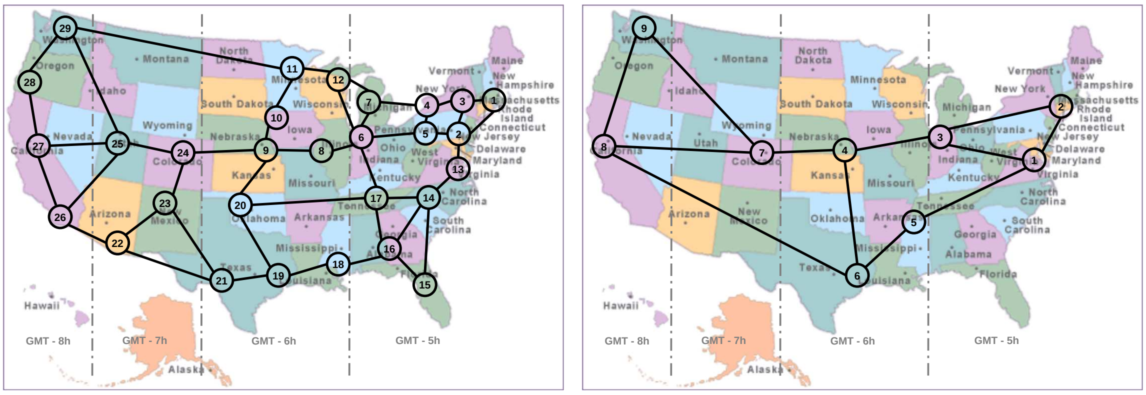
$\mathfrak{C}_{i,j}$	1	2	3	4
1	-	0.6857	0.5918	0.6060
2	0.6857	-	0.4307	0.3636
3	0.5918	0.4307	-	0
4	0.6060	0.3636	0	-

Finally, the enhanced time correlation between the requests of the set Y is equal to:

$$\begin{aligned}\mathfrak{C}_Y &= \frac{1}{N_Y \times (N_Y - 1)} \times \sum_{i=1}^{N_Y} \sum_{j=1 \backslash j \neq i}^{N_Y} \mathfrak{C}_{i,j} \\ &= \frac{2}{4 \times 3} \times \Big(0.6857 + 0.5918 + 0.6060 + 0.4307 + 0.3636 + 0\Big) \\ &= 0.4463\end{aligned}$$

3.6 Traffic Sets

For our simulations, we have mainly considered the National Science Foundation NETwork (NSFNET) topology [W162, W163] which has 29 nodes and 44 bidirectional links as shown in Figure 3.7(a). In some scenarios, we were obliged to use smaller network in order to keep the computation time within acceptable limits. For those scenarios, we have considered the 9-node and 12-link NSF network shown in Figure 3.7(b). A weight representing the physical length is associated to each link of both networks. This weight is used to calculate the shortest paths between each pair of nodes. In addition, as mentioned earlier in Section 3.4.1, a weight p_i is assigned to each node i according to the number of citizens of the corresponding region [W166]. These weights are integer numbers ranging from 1 to 10 and represent the capacity of a node to send/receive data. Information on the population surrounding a node, the geographical time-zone shift, as well as the weight assigned to each node are given in Table 3.7.



(a) 29-node and 44-link NSF network

(b) 9-node and 12-link NSF network

Fig. 3.7. The considered network topology

SLD/SED Traffic Sets

In our evaluations, we have assumed an average traffic flow ϕ_{SxD} evaluated over the whole network during a one day period. Based on our traffic model for scheduled traffic requests already introduced in Section 3.4.1, we have generated the corresponding set of deterministic source-destination traffic flows. Each of these traffic flows between a source and a destination may be decomposed into multiple SLDs and a single SED. Let us recall that an SLD corresponds to the full capacity of an optical channel whereas an SED corresponds to a fraction of this capacity. As a result, the global set of SxDs is composed of N_{SLD} SLDs and N_{SED} SEDs. The set of SLDs represents an average rate of $\phi_{SLD}(Tbps)$ with a peak value of $\pi_{SLD}(Tbps)$, while the set of SEDs represents an average rate of $\phi_{SED}(Tbps)$ with a peak value of $\pi_{SED}(Tbps)$. The overall traffic peak is equal to $\pi_{SxD}(Tbps)$. We have generated 7 sets of scheduled traffic requests. Table 3.8 summarizes the characteristics of the generated traffic sets.

Only the SxD_1 and the SxD_2 traffic sets are generated for the 9-node, 12-link NSF network. The remaining sets are generated for the 29-node, 44-link NSF network.

RLD/sRLD/RED/sRED Traffic Sets

In our evaluations, we have generated various sets of sRXDs according to the traffic model described in Section 3.4.2. Each set is divided into two subsets (sRLDs and sREDs) according to the requested capacity of the demands. Let us recall that the arrival time of both sRLDs and sREDs follows a Poisson process of parameter λ and $\lambda'(sec^{-1})$, respectively, while the holding-time of these requests follows a negative Exponential distribution of parameter δ and $\delta'(sec^{-1})$, respectively. Hence, on average, a new sRLD arrives every $1/\lambda$ seconds and holds the network for about $1/\delta$ seconds, while a new sRED arrives every $1/\lambda'$ and holds the network for about $1/\delta'$. For a given sRxDs traffic set, the sRLD subset contains N_{sRLD} sRLD requests with an average traffic flow of ϕ_{sRLD} and an overall traffic peak of π_{sRLD}, while the sRED subset contains N_{sRED} sRED requests with an average traffic flow of ϕ_{sRED} and an overall traffic peak of π_{sRED}. As a result, the global set of sRxDs contains N_{sRxD} requests ($N_{sRxD} = N_{sRLD} + N_{sRED}$) with an average traffic flow of ϕ_{sRxD} and an overall traffic peak of π_{sRxD}. It is to be noted that RxDs traffic sets can be obtained from the sRxDs traffic sets by neglecting their life duration. Hence, the life duration of the RxDs is assumed to be infinite. We have generated 5 different categories of sRxDs/RxDs traffic requests. For a given arrival rate and a given

Table 3.7. Physical location, time-zone shifts, and weight of each node in the network

29-node and 44-link network	9-node and 12-link network	State	Number of citizens (in 2002)	Time-zone shift (GMT)	Weight
1	2	Massachusetts	6.5 millions	-5 hour	3
2	-	New Jercy	8.5 millions	-5 hour	4
3	-	New York (1)	10.5 millions	-5 hour	5
4	-	New York (2)	10.5 millions	-5 hour	5
5	-	Pennsylvania	12.5 millions	-5 hour	6
6	3	Indiana	6 millions	-5 hour	3
7	-	Michigan	10 millions	-5 hour	5
8	-	Illinois	12.5 millions	-6 hour	6
9	4	Nebraska	1.75 millions	-6 hour	1
10	-	Iowa	3 millions	-6 hour	2
11	-	Minnesota	5 millions	-6 hour	3
12	-	Wisconsin	5.5 millions	-6 hour	3
13	1	Virginia	7.5 millions	-5 hour	4
14	-	North California	8.5 millions	-5 hour	4
15	-	Florida	16.75 millions	-5 hour	8
16	-	Georgia	8.5 millions	-5 hour	4
17	5	Tennessee	5.75 millions	-5 hour	3
18	-	Mississippi	4.5 millions	-6 hour	2
19	6	Texas (1)	12 millions	-6 hour	6
20	-	Oklahoma	3.5 millions	-6 hour	2
21	-	Texas (2)	9 millions	-6 hour	5
22	-	Arizona	5.5 millions	-7 hour	3
23	-	New Mexico	2 millions	-7 hour	1
24	7	Colorado	4.5 millions	-7 hour	2
25	-	Utah	2.5 millions	-7 hour	1
26	-	California (1)	20 millions	-8 hour	10
27	8	California (2)	15 millions	-8 hour	8
28	-	Oregon	3.5 millions	-8 hour	2
29	9	Washington	6 millions	-8 hour	3
			227,25 millions		114

Table 3.8. Statistic measurement of the SxD sets

	SxD				SLD				SED			
	N_{SxD}	ϕ_{SxD} (Tbps)	π_{SxD} (Tbps)	$\mathfrak{C}_{SxD}$ (%)	N_{SLD}	ϕ_{SLD} (Tbps)	π_{SLD} (Tbps)	$\mathfrak{C}_{SLD}$ (%)	N_{SED}	ϕ_{SED} (Tbps)	π_{SED} (Tbps)	$\mathfrak{C}_{SED}$ (%)
SxD_1	250	0.091	0.132	25.59	-	-	-	-	250	0.091	0.132	2.559
SxD_2	1000	0.658	1.010	23.12	1000	0.658	1.010	23.12	-	-	-	-
SxD_3	2000	1.275	1.797	20.62	2000	1.275	1.797	20.62	-	-	-	-
SxD_4	5000	1.645	2.401	21.93	-	-	-	-	5000	1.645	2.401	21.93
SxD_5	7000	2.920	4.198	21.55	2000	1.275	1.797	20.62	5000	1.645	2.401	21.93
SxD_6	5000	1.518	2.185	22.17	-	-	-	-	5000	1.518	2.185	22.17
SxD_7	7874	1.023	5.226	33.08	2190	0.376	2.182	18.99	5684	0.647	3.043	33.07

holding time (characterizing a category), we have generated 50 different sets of traffic requests. Table 3.9 summarizes the average characteristics of the generated traffic sets.

Table 3.9. Statistic measurement of the sRxD/RxD sets

	X = sRxD/RxD				Y = sRLD/RLD				Z = sRED/RED			
	N_X	ϕ_X (Gbps)	π_X (Gbps)	$\mathfrak{C}_X$ (‰)	N_Y	ϕ_Y (Gbps)	π_Y (Gbps)	$\mathfrak{C}_Y$ (‰)	N_Z	ϕ_Z (Gbps)	π_Z (Gbps)	$\mathfrak{C}_Z$ (‰)
$sRxD_1/RxD_1$	2659	66.47	112.5	6.67	2659	66.47	112.5	6.67	-	-	-	-
$sRxD_2/RxD_2$	8158	71.84	110.6	4.59	-	-	-	-	8158	71.84	110.6	4.59
$sRxD_3/RxD_3$	10817	138.3	195.9	5.04	2659	66.47	112.5	6.67	8158	71.84	110.6	4.59
$sRxD_4/RxD_4$	2818	75.08	118.9	13.7	-	-	-	-	2818	75.08	118.9	13.7
$sRxD_5/RxD_5$	4268	74.51	102.9	9.33	-	-	-	-	4268	74.51	102.9	9.33

The categories $sRxD_2/RxD_2$, $sRxD_4/RxD_4$, and $sRxD_5/RxD_5$ correspond to the same average traffic flow $\phi_{sRxD/RxD}$ but different burstiness level.

4

Nodal Architecture

Abstract: *Multiple node architectures are possible for WDM networks. These architectures strongly condition the cost of optical network planning and traffic engineering. This chapter aims first to introduce the four optical switching systems currently considered; namely Optical Cross-Connect (OXC), Optical Circuit Switching (OCS), Optical Packet Switching (OPS), and Optical Burst Switching (OBS). Then a detailed description of a multi-layer EXC/OXC node architecture is presented. Such an architecture can handle full-wavelength circuits as well as sub-wavelength electrical connections. The major functionalities of this EXC/OXC node (add, drop, switching, and grooming) are modeled by means of an auxiliary graph. The node architecture and its auxiliary graph model are used in the following chapters for network planning and traffic engineering, respectively.*

4.1 Overview

Traditional multi-layer networks have been deployed using different technologies such as Synchronous Optical NETwork (SONET) / Synchronous Digital Hierarchy (SDH) and Asynchronous Transfer Mode (ATM) as a transport infrastructure for Internet Protocol (IP)-based services. The first generation (1985) of such optical networks were based on ring topologies in the metro area and on meshed topologies in the long haul. A few years later (1990), Wavelength Division Multiplexing (WDM) has been deployed progressively first in the core and then in the metro area. A key element of such network topology is the nodal architecture and the ability of this architecture to insert traffic into the network and to drop locally oriented traffic to the client layer. In addition, a node might perform other functions such as traffic pass-through and traffic grooming.

In a traditional SONET/SDH ring network, in order to transmit or receive traffic on a wavelength, the wavelength must be added or dropped at a particular node. This is achieved by means of an electronic Add-Drop Multiplexer ADM that is able to separate a high-rate signal into lower rate components [J96]. Generally, each ADM is equipped with a fixed transceiver and operates only on one wavelength. As a result, one ADM is needed for each wavelength at every node to add/drop traffic signals (Figure 4.1). With the progress of WDM, over a hundred wavelengths are now supported simultaneously by a single fiber. It is therefore too costly to put the same amount of ADMs at every network node since a lot of traffic is only bypassed at intermediate nodes.

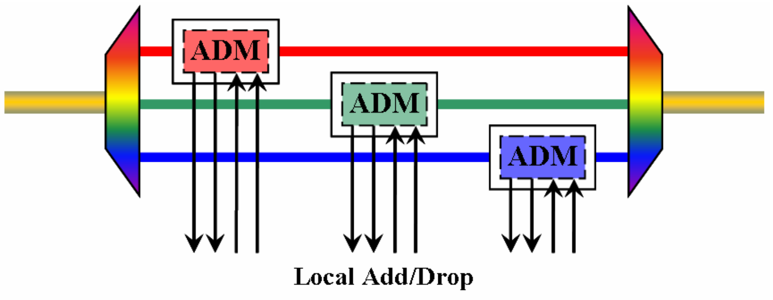


Fig. 4.1. Add/Drop Multiplexer (ADM)

With the emerging optical components such as Optical Add-Drop Multiplexers (OADM), it is possible for a node to optically bypass most of the wavelength channels and to drop only the wavelengths carrying the traffic destined to the node (Figure 4.2) [J97]. By carefully arranging these optical bypasses taking advantage of the OADM functionalities, one can decrease the number of ADMs used in the network. Therefore, it is clear that OADMs can reduce a large amount of the network cost.

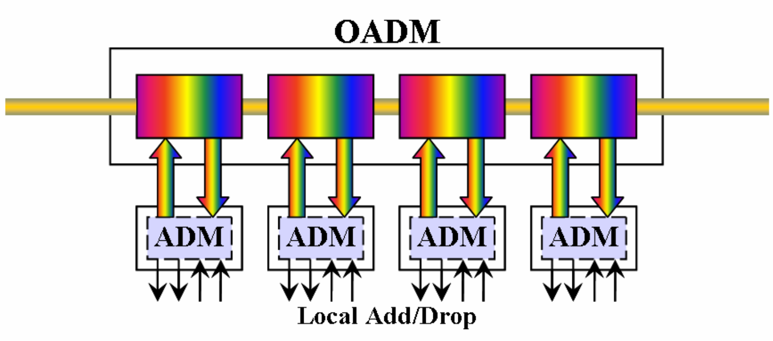


Fig. 4.2. Optical Add/Drop Multiplexer (OADM)

Although the SONET/SDH ring network has been used as the first generation of the optical network infrastructure, it has some limitations which make it hard to accommodate the increasing Internet traffic. Indeed, such equipment is not able to manage fluctuating traffic matrices. In addition, it lacks of multiplexing flexibility to cope with the dynamics of Internet traffic. The next-generation optical network is expected to be an intelligent wavelength-routed WDM mesh network [J98, J99]. This network will provide fast and convenient automatic bandwidth provisioning and efficient protection mechanisms, and it will be based on an irregular mesh topology which will make it much easier to scale. As a mesh network, the degree of connectivity of a particular node is greater than 2. This means that a node can be connected to two or more different nodes through the fiber lines. This high degree of connectivity needs a cross-connect like architecture to achieve wavelength routing in WDM mesh networks (Figure 4.3). A key feature of a cross-connect is its ability to switch any lightpath at any incoming port to any outgoing port as long as the outgoing port is not used. In the following, we only focus on the cross-connect architecture used in mesh networks.

4.2 Node Architecture

To satisfy a connection request in a WDM network, a lightpath is ideally established between the source node and the destination node. In practice, due to the lack of available optical channels, a connection

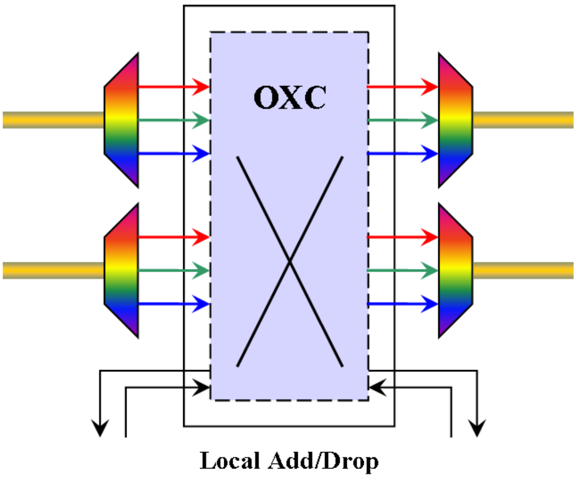


Fig. 4.3. Simplified architecture of a cross-connect

request may traverse through one or more lightpaths before it reaches its destination. Two important functionalities have to be supported by WDM network nodes, namely *wavelength routing* and *optical multiplexing/demultiplexing*. By means of optical multiplexing/demultiplexing, several wavelengths can be multiplexed to or demultiplexed from the same fiber-link. Besides, in order to groom low-speed connections onto a high-speed wavelength channel, a node has to be able to multiplex/demultiplex and switch low-speed connections using various multiplexing techniques.

A WDM mesh network may consist of systems from multiple vendors, and different vendors may employ different node architectures with different grooming capabilities. Some architectures may have full grooming capabilities, while others may impose some constraints on the grooming capability such as limiting the number of transceivers used for originating and terminating groomable wavelength channels. Some nodes may have no grooming capability at all. In terms of wavelength conversion capability, some nodes in the network may have full, partial, or no wavelength conversion capability. Despite this heterogeneity in the network node functionalities, each node is composed of an Electrical Circuit Switch (ECS) coupled to an Optical Circuit Switch (OCS). Such a two layered architecture allows dynamic configuration of the add, drop, and bypass ports as well as it facilitates multi-granular switching and grooming functionalities.

4.2.1 Optical Nodes

According to the state of the technology, one can distinguish four types of optical nodes: *Optical Cross-Connect* (OXC), *Optical Circuit Switch* (OCS), *Optical Packet Switch* (OPS), and *Optical Burst Switch* (OBS). An OXC is the most mature technology and the less complex to implement since it does not require any control plane nor signaling. It is circuit-switching oriented, circuits being established manually by the carrier for the long term. OCS and OPS are also circuit-switching oriented but require a control plane and a signaling channel in order to be reconfigured dynamically and automatically. With OCS, the whole capacity of an optical channel is dedicated to a logical connection during its life duration. At the opposite, an OPS shares this capacity into timeslots. One or multiple timeslots within successive frames are assigned to a logical connection. At last, an OBS is connectionless oriented since it does not require a call establishment procedure with resource reservation from source to destination. In terms of technology, it is widely admitted that OXC and OCS are totally mature and are already installed in existing operational networks. At the opposite, OPS and OBS rely on advanced optical devices and systems and are considered as mid/long term solutions.

Optical Cross-Connect (OXC)

Traditionally, OXCs can switch any optical channel from one port to any other port. A circuit or a lightpath (in the optical domain) is an end-to-end connection over which electrical data are transported. The philosophy behind circuit switching is that a lightpath between a particular source and destination pair can be established for a sufficient long period of time. Current technology facilitates circuit switching which makes it more feasible at the short term.

A number of OXC solutions based on different technologies have been proposed to date. Depending on the switching technology and the architecture used, they are commonly divided into two main categories: opaque and transparent [C34, C35]. Opaque OXCs are either based on electrical switching technology or on optical switching fabrics surrounded by expensive Optical-Electrical-Optical O/E/O conversion devices. Opaque OXCs also offer inherent regeneration, wavelength conversion, and bit-level monitoring. In transparent OXCs, the incoming signals are routed through an all-optical O/O/O switching fabric without undergoing opto-electrical conversions, thereby offering transparency to a variety of bit rates and protocols.

One of the viable all-optical switching technologies is based on Micro-Electro-Mechanical Systems (MEMS) [J100, J101]. MEMS optical switching fabrics have demonstrated their superiority over competing technologies such as bubble-jet switches [J102, J103], liquid-crystal switches [J104, J105], thermo-optical switches [C36, J106], and acousto-optical switches [J107, J108] in terms of their scalability, insertion loss, Polarization-Dependent Loss (PDL), wavelength dependence, and crosstalk properties. MEMS switches achieve switching using two approaches: two-dimensional (2D) MEMS and three-dimensional (3D) MEMS. The 2D MEMS architecture uses mirrors that can switch "ON" and "OFF" (binary behavior). The positioning of the two mirrors achieve the optical switching. Currently, a 2D MEMS optical switch is based on the conventional crossbar matrix architecture. One of the major drawbacks of the conventional crossbar matrix architecture is that the free-space optical path difference between the most distant and the least distant paths is significant. These path differences induce unfairness and unpredictability in terms of power budget. In practice, 2D MEMS switches are restricted to a size of 32×32. The 3D MEMS switches enable larger switching fabrics than 2D MEMS switches but are more sensitive to physical layer impairments such as crosstalk.

Optical Circuit Switch (OCS)

The OCS has the same switching fabric as the OXC and is based on the same technology. The only difference with an OXC network is that an OCS network has a control plane which makes it more dynamic and able to be reconfigured automatically. Optical circuit switching is performed at the wavelength granularity and is sometimes referred to as Optical Wavelength Switching (OWS). One of the drawbacks of OCS networks is their lack of multiplexing flexibility, especially when the network is dominated by packet-based Internet Protocol (IP) traffic [J109, J110]. This limitation can be addressed by using nodes that are capable of switching smaller granularities in the optical layer. Two approaches are possible in this matter. A first solution consists in coupling an Electrical Circuit Switch (ECS) or an Electrical Packet Switch (EPS) to an OCS (as mentioned above). A second solution is to replace the OCS by an Optical Packet Switch (OPS) or an Optical Burst Switch (OBS). OPS and OBS switch optical data at the granularities of packets and bursts, respectively. This second solution is at evidence more prospective. As OCSs and OXCs are based on the same technology, we refer by OXCs in the following to both OCSs and OXCs.

Optical Packet Switch (OPS)

Among all the optical sub-wavelength switching techniques, Optical Packet Switching (OPS) is the most challenging candidate for building the next-generation Internet backbone [J111, J112]. It has a fine switching granularity (at the packet level) and provides seamless integration between the WDM layer and the IP layer. Whereas OXC and OCS operate in continuous time, OPS networks operate in discrete time. Indeed, at the input of an OPS, all the time-slots arriving in parallel on different optical channels must be resynchronized. The main technological challenge of OPS networks is to proceed to this resynchronization all optically. When the packets are being switched, contention may occur whenever two or more packets are trying to leave the switch on the same output port [J113]. In EPS, such contention is resolved using electrical buffering (Random Access Memory (RAM)) at the output of the switch. The fact that there is no available optical RAMs represents the major difference between the design of an optical packet switched network and an electronic one.

Instead of using RAMs, the first OPS test-beds used optical delay lines in order to resolve contentions [J114, C37]. Such an approach clearly lacks of scalability. Later, other alternatives have been investigated in this matter; namely wavelength conversion [J115, J116] and path (space) deflection [C38, C39]. Wavelength conversion is very efficient; it is able to resolve contention without introducing extra delay to the packet if carried out in the optical domain. However, all-optical wavelength conversion is expensive to implement, and no full-range wavelength converters are available today. Path deflection does not require expensive devices since it shifts the burden of resolving contention to the next adjacent node. However, its main drawback is that it does not use efficiently the network resources and may require packet reordering at the destination. An OPS network being totally desynchronized, OPS technology remains a long-term solution for the next-generation Internet backbone.

Optical Burst Switch (OBS)

The OBS paradigm [J117, J118] emerged as a compromise between circuit switching and packet switching. The OBS is an approach that attempts to shift the computation and control complexity from the optical domain to the electrical domain at the edge nodes. Burst switching was introduced in optical networks in the mid 90's [C40, C41]. OBS has a switching granularity between a circuit and a packet. In an IP-over-WDM network, a burst formed at the network edge may contain a number of IP packets, and it can be a few tens of kilobytes to a few megabytes long. In OBS, a control packet, which is separated from the data burst that carries the payload, is first sent to configure each switch along the path of the data burst. By assembling packets into larger data bursts, the burden on electronic control is reduced. In addition, by transmitting data and control signals in two separated optical bursts, the signaling burst is processed in the electrical domain avoiding the need for optical header processing. Therefore, OBS is considered as one of mid-term solutions for the next-generation Internet backbone technology.

4.2.2 Electrical Nodes

A wavelength-routed network can only provide a whole wavelength as the smallest bandwidth granularity. A sub-wavelength circuit can only be obtained via traffic aggregation/de-aggregation performed in the electrical domain. The higher the bit rate, the more expensive the aggregation equipment. Traffic aggregation/de-aggregation, referred to as *"traffic grooming"*, is important for the efficient utilization of the large bandwidth provided by wavelength channel (*e.g.,* 10 Gbps (OC-192) and 40 Gbps (OC-768)). Electrical traffic grooming can use different multiplexing techniques such as Frequency Division Multiplexing (FDM) or Time Division Multiplexing (TDM) [J119]. In FDM, the whole bandwidth of the wavelength channel is divided into a number of frequency sub-channels, each sub-channel can accommodate an individual connection. As a result, each logical connection uses continuously a fraction of the bandwidth capacity. On the other hand, TDM divides the whole transmission time into repeated time-slots of fixed length. As a result, each logical connection uses the whole bandwidth capacity during discrete time-slots. As an example, traffic aggregation in SONET/SDH networks is performed using TDM scheme. However, this TDM-based synchronous aggregation scheme might not be the most cost-effective solution in mesh networks under bursty traffic. In the following, the electrical nodes refer to a circuit-oriented switches such as: SONET/SDH switches, Frame Relay swithes, ATM switches, MPLS switches/routers, and MPLS/Ethernet switches.

4.2.3 Wavelength Converters

A wavelength converter can transfer a signal on one wavelength at its input port to another wavelength at its output port. A group of wavelength converters is called wavelength converter bank or wavelength converter pool. A switch with wavelength conversion capability can route a signal from any input port to any output port on different wavelengths provided it does not conflict with the signals on that port. Wavelength conversion methods can be grouped into two types: Opto-electronic conversion and all-optical conversion [C42].

- Under opto-electronic conversion, wavelength conversion is achieved by transforming the optical signal to the electrical domain and by converting it back to the optical domain on a different wavelength. Opto-electronic conversion introduces an end-to-end delay penalty. The main drawback of opto-electronic conversion is its lack of flexibility since electrical circuits are designed to operate at a given data rate [J120].
- Under all-optical conversion, the signal remains in the optical domain. The all-optical conversion techniques can be further grouped into two categories [J121]: those which employ coherent effects such as Four Wave Mixing (FWM) [J122, J123] and Difference Frequency Generation (DFG) [J124, C43], and those which use optical gating effects such as absorption saturation of the semiconductor laser diode [J125], cross-modulation of optical amplifier in Semiconductor Optical Amplifiers (SOAs) (Cross-Phase Modulation (XPM) [J126, J127], and Cross-Gain Modulation (XGM) [J126, J128]). Wavelength conversion methods using coherent effects are typically based on wave-mixing properties. Wave-mixing arises from a nonlinear optical response of a medium when more than one wave is present. It results in the generation of another wave whose intensity is proportional to the product of the interacting wave intensities. Wave-mixing preserves both phase and amplitude information, offering strict transparency. Wavelength converters based on optical wavelength gating employs an optical device which changes its characteristics depending on the intensity of the input signal. It is to be noted that the quality of the signal deteriorates

when it passes through cascaded conversions, and therefore, the number of cascaded conversions should be kept to a minimum.

Wavelength conversion as such is quite an expensive technology. Current technology validates only O/E/O wavelength converters which are protocol and bit rate dependent. Conversely, all-optical conversion is protocol independent and bit rate independent, but their implementation is restricted at this time. However, protocol independent wavelength converters are expected to be more popular in the near future.

Considering a network with no wavelength conversion capability, the establishment of a new lightpath could be impossible even though there is a free wavelength on each of the fiber-links from its source to its destination. This is because the available wavelengths on the various fiber-links along its path are different. By deploying wavelength conversion or translation in optical networks, the wavelength continuity constraint is eliminated. Thus, full wavelength conversion may improve the efficiency/blocking probability in the network by resolving the wavelength conflicts of the lightpaths at the price of a larger network deployment cost. It may not be cost-effective to equip all the output ports of a switch with wavelength converters since all the wavelength converters may not be required simultaneously. Meanwhile, recent work has shown that a limited wavelength conversion yields similar results to full wavelength conversion under certain generic assumptions without inducing a large hardware cost [J129].

Sparse or limited wavelength conversion can be of three types:

1. a limited number of nodes are provided with full wavelength convertibility [J130]
2. a limited number of wavelength converters are placed at each node [J131]
3. only limited-range wavelength conversion are possible at the nodes [C44]

A wavelength converter can be dedicated to each channel, or a group of channels can share a wavelength converter to reduce the cost. Different node architectures with various sparse or limited wavelength conversion strategies have been analyzed. It has been shown that OXCs with a bank/pool of completely shared wavelength converters require less wavelength converters compared to OXC architectures that use a dedicated wavelength converter per channel [J131]. This reduction depends on the location of the OXC, the network diameter, the topology of the network, and the traffic load in the network. However, in general terms, about 50% to 90% of wavelength converters can be saved by architectures with shared wavelength conversion resources.

4.2.4 Signaling Protocol

Future Internet network will be built on various cores, edges, and individual networks composed of optical and wireless technologies. The skeleton of the Internet (*i.e.,* the backbone) would, in all possibility, be composed of a high bandwidth multi-wavelength optical core. An all-optical Internet core would facilitate fast and efficient switching as well as dynamic lightpaths adding and dropping to ensure seamless communication between end users. In addition, traffic grooming is a practical and important problem for WDM network design and implementation. The unified control plane of such a network, known as Generalized Multi-Protocol Label Switching (GMPLS) [J132, J133], is being standardized by the Internet Engineering Task Force (IETF)[1] [2]. The purpose of this network control plane is to provide an intelligent automatic end-to-end circuit (virtual circuit) provisionning/signaling scheme throughout the different network domains. GMPLS to a great extent aims at coordinating the operation of the various layers of a protocol stack. Setting-up and tearing-down lightpaths dynamically is its primary function. This can be extended to packet switching. Although burst switching is a promising paradigm, GMPLS has not been extended to OBS functionalities.

There are three components in the WDM control plane that need to be carefully designed in order to support traffic grooming, namely *resource discovery protocol*, *signaling protocol*, and *path computation algorithms*.

- Resource discovery protocols determine how the network resources are discovered, represented, and maintained in the nodes' link-state databases (for distributed control) or by the network control and management system (for centralized control). Several resource discovery protocols based on traffic engineering extensions of link-state protocols (Open Shortest Path First (OSPF)[3], Intermediate System to Intermediate System (IS-IS)[4]) and link management protocols[5] have been proposed by IETF as the resource discovery component in the control plane.
  Thanks to the neighbor discovery process, each node in the network learns about all neighbors. Moreover, a flooding procedure allows each node to periodically flood this information through the network. Each node processes all incoming link-state packets and stores the received status information of each link in its link-state database. By having such a link-state database in each network node, each node shares a common view of the network topology and resources, and thus can compute the constraint shortest path along which a Label Switched Path (LSP) has to be set-up.
- Path computation algorithms determine how the route of a low-speed connection request is computed and selected according to the network carrier's grooming policy. The grooming policy reflects the strategy of the network operator on how to allocate network resources to a given request if multiple routes are available. It is a traffic engineering decision and the design of an efficient path computation algorithm is still an open issue.
  Path computation has previously been performed either in a management system or at the head-end of each lightpath. Such path computation includes the generation of primary, protection, and recovery paths, as well as computations for (local/global) reoptimization and load balancing. Thus, path computation in large, multi-domain, multi-region, or multi-layer networks is complex and may require special computational components and cooperation between the different network

[1] [RFC 3945]: Generalized Multi-Protocol Label Switching (GMPLS) Architecture (October 2004). Available at: http://www.ietf.org/rfc/rfc3945.txt

[2] [RFC 3471]: Generalized Multi-Protocol Label Switching (GMPLS): Signaling Functional Description (January 2003). Available at: http://www.ietf.org/rfc/rfc3471.txt

[3] [RFC 4203]: OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS) (October 2005). Available at: http://tools.ietf.org/rfc/rfc4203.txt

[4] [RFC 4205]: Intermediate System to Intermediate System (IS-IS) Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS) (October 2005). Available at: http://tools.ietf.org/rfc/rfc4205.txt

[5] [RFC 4204]: Link Management Protocol (LMP) (October 2005). Available at: http://tools.ietf.org/rfc/rfc4204.txt

domains. Recently[6], the IETF specified the architecture for a Path Computation Element (PCE)-based model to address this problem space. A PCE is defined as an entity capable of computing complex paths for a single or set of services. It is aware of the network resources and has the ability to consider multiple constraints for sophisticated path computation. A PCE might be a network node, a network management station, or a dedicated computational platform. PCE applications include computing LSPs for MPLS and GMPLS Traffic Engineering. Future applications might allow for dynamic control of network resources. For instance, in the event of a catastrophic failure, a PCE entity would have the capability to reroute services in real-time and around network failures in order to minimize network downtime and service interruption. The various components of the PCE architecture are in the process of being standardized by the IETF's PCE Working Group [W167].

- Signaling protocols determine how the connections are configured and how a network node allocates its local network resources to these connections (port mapping, label assignment, ...). Two protocols, Resource Reservation Protocol (RSVP) with traffic engineering extensions[7] and Constraint based Routing Label Distribution Protocol[8] (CR-LDP), have been proposed as the standard signaling protocols in the GMPLS control plane. The main idea of GMPLS is to extend the original MPLS scheme.

  MPLS extended the suite of IP protocols to expedite the forwarding scheme used by IP routers. Routers, to date, have used complex and time-consuming route lookups and address matching schemes to determine the next hop for a received packet, primarily by examining the destination address in the header of the packet. MPLS has greatly simplified this operation by basing the forwarding decision on a simple label represented as an integer number and attached to the IP packet as an additional shim header. Another major feature of MPLS is its ability to place IP traffic on a defined path through the network. This capability was not previously possible with IP traffic. In this way, MPLS provides bandwidth guarantees and other differentiated service features for a specific user application (or flow).

  GMPLS extends MPLS to provide the control plane (signaling and routing) for devices that switch in any of these domains: packet, time, wavelength, and fiber. Therefore, a label can now be represented as an integer, a timeslot in a TDM-frame, a wavelength or waveband on a fibre, a fibre in a cable, and so on. The idea is to reuse the same protocol suite, adopted in regular MPLS to set-up and tear-down LSPs, to be able to control switched (instead of permanent) connections through optical transport networks. However, conversely to MPLS which needs to recognize the packet boundaries and extract the label information before switching the packet, GMPLS performs switching without depending on recognizing the packet boundaries or the header information. Such a scheme must depend on some other optical properties (label mapping spaces) to find out the forwarding class for a packet before switching it to its destination. To conclude, GMPLS promises to simplify network operation and management by automating end-to-end provisioning

[6] [RFC 4655]: A Path Computation Element (PCE)-Based Architecture (August 2006). Available at: http://www.ietf.org/rfc/rfc4655.txt

[7] [RFC 3473]: Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions (January 2003). Available at: http://www.ietf.org/rfc/rfc3473.txt

[8] [RFC 3472]: Generalized Multi-Protocol Label Switching (GMPLS) Signaling Constraint-based Routed Label Distribution Protocol (CR-LDP) Extensions (January 2003). Available at: http://www.ietf.org/rfc/rfc3472.txt

of connections, managing network resources, and providing the level of QoS that is expected in the new sophisticated applications.

4.3 Adopted Node Architecture

For our study, we have adopted a rather simple and flexible node architecture where a non-blocking Electrical Cross-Connect (EXC) is coupled to a non-blocking Optical Cross-Connect (OXC). This double stage node architecture allows dynamic configuration of the add, drop, and bypass ports as well as hierarchical switching and grooming functionalities. The optical layer, at the OXC level, is used to add, drop, and switch full wavelength data requests, while the electrical layer, at the EXC level, is used to add, drop, switch, and groom lower rate traffic requests and can perform arbitrary wavelength conversion. Figure 4.4 illustrates the proposed node architecture.

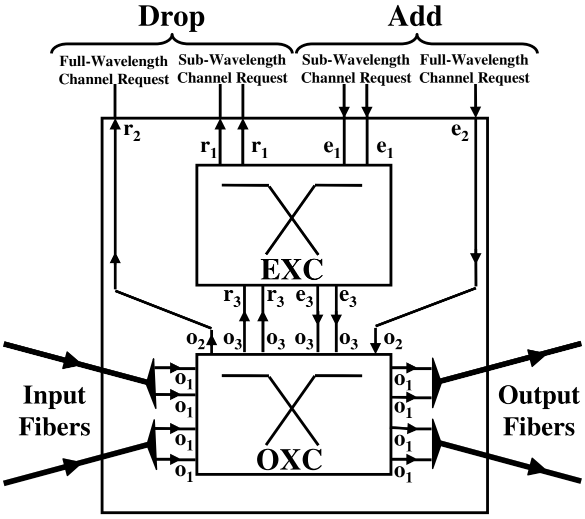

Drop
Add
Full-Wavelength Channel Request
Sub-Wavelength Channel Request
Sub-Wavelength Channel Request
Full-Wavelength Channel Request
r2
r1
r1
e1
e1
e2
EXC
r3
r3
e3
e3
o2
o3
o3
o3
o3
o2
Input Fibers
Output Fibers
o1
o1
o1
o1
o1
o1
o1
o1
OXC

Fig. 4.4. Node architecture

Due to the hierarchical and hybrid nature of this node architecture, the network topology can be considered on two different levels: the physical topology and the logical/virtual topology. The physical topology is the one seen by the optical layer and stays relatively static once configured. It consists of a set of OXCs interconnected by fiber-links. A wavelength comb received at each input fiber is first demultiplexed in order to isolate each optical channel. These optical channels are then inserted into the OXC via receiving o_1 optical ports. If a particular wavelength needs to be relayed to an output fiber for retransmission, it can be switched all-optically to an emitting o_1 optical port at this level. This wavelength switching operates with optical channel granularity. At the other end of the OXC, wavelength multiplexers collect the wavelengths carrying data stream and merge them onto common outgoing fibers. The logical/virtual topology, seen by the electrical layer, consists of a set of EXCs interconnected by lightpaths. In this way, WDM networks provide a way to interconnect electronic switches with high bandwidth bit-pipes without requiring a full-mesh physical topology. The design of static network topologies has been studied extensively in the past [J70]. However, the configurable nature of the EXCs also allows the logical topology to be dynamically reconfigured in response to traffic fluctuations. This is achieved by changing the lightpath connectivity between the EXCs, thereby reconfiguring the virtual topology. In our case, these reconfigurable lightpaths may be shared by many sub-wavelength requests and are referred to as *Grooming Lightpaths* (GLs). A GL is a direct connection between two nodes acting as a logical one-hop link, where all intermediate nodes are passed through at the OXC level. In case a low-speed data-connection carried by a given GL needs to be switched to a different GL or needs to be dropped to the local client, the wavelength carrying the parent GL is switched from the OXC to the EXC. For this purpose, one uses an emitting o_3 optical port and a receiving r_3 electrical port. At the EXC level, some low-speed data connections may be extracted from the GL, while others may be inserted into it. Finally, the remaining sub-wavelength connections are groomed together onto a new GL that is switched back to the OXC using an emitting e_3 electrical port and a receiving o_3 optical port. The number of ports, which connect the EXC to the OXC, determines the multi-hop grooming capability of this node. Both the OXC and the EXC have "add" and "drop" ports for locally-generated and locally-terminated traffic requests. An OXC add port (an emitting e_2 electrical port and a receiving o_2 optical port) can add a full wavelength data connection, while an EXC add port (an emitting e_1 electrical port) can add a single sub-wavelength data connection. Similarly, an OXC drop port (an emitting o_2 optical port and a receiving r_2 electrical port) can drop a full wavelength data connection, while an EXC drop port (a receiving r_1 electrical port) can drop a single sub-wavelength data connection. Since an all-optical wavelength switch does not provide wavelength conversion capability, wavelength conversion can be carried out at the EXC level or by means of a bank of wavelength converters. Let us recall that, compared to full wavelength conversion, limited wavelength conversion achieves an acceptable connection rejection ratio [J131]. In addition, networks equipped with multiple fibers on each link have been suggested as a possible alternative to wavelength conversion [C45]. For all these reasons, we have chosen to provide each node in the network with a pool of wavelength converters. The size of these pools is out of the scope of this work.

It is to be noted that the input and the output ports of the EXC as well as the input and the output ports of the OXC operate at an optical channel data rate (2.5 Gbps or 10 Gbps in current networks). In addition, the emitting/receiving optical ports could be either tunable transceivers or fixed transceivers. A tunable transceiver can be tuned between different wavelengths so that it can send out (or receive) an optical signal on any free wavelength in its tuning range. A fixed transceiver can emit (or receive) an optical signal on only one wavelength. To explore all the wavelength channels of a fiber, a set of fixed transceivers, one per wavelength, can be grouped together to form a transceiver array. The size of a fixed transceiver array can be equal to or smaller than the number of wavelengths on a fiber, and the number of transceiver arrays can be equal to or smaller than the number of fibers joining/leaving a node. When designing our node, we have chosen to use tunable transceivers. The number of input/output ports of a single OXC is determined by:

- the number of input/output fibers
- the number of full-wavelength channel add/drop ports (o_2 optical ports)
- the number of ports that connect it to the EXC (o_3 optical ports)

Similarly, the size of the EXC is determined by:

- the number of ports that connect the EXC to the OXC (receiving r_3 electrical ports and emitting e_3 electrical ports)
- the number of local add/drop ports that are required (receiving r_1 electrical ports and emitting e_1 electrical ports)

In other words, let R be the number of optical ports at the entry side of the OXC and let E be the number of optical ports at the exit side of the OXC. R is equal to the sum of the number of emitting e_2 electrical ports, the number of emitting e_3 electrical ports, and the number of receiving o_1 optical ports. Similarly, E is equal to the sum of the number of receiving r_2 electrical ports, the number of receiving r_3 electrical ports and the number of emitting o_1 optical ports. Hence, the size of the OXC is equal to the maximum value between R and E. In the same context, the number r of electrical ports at the entry side of the EXC is equal to the sum of the number of emitting e_1 electrical ports and the number of receiving r_3 electrical ports. Similarly, the number e of electrical ports at the exit side of the EXC is equal to the sum of the number of receiving r_1 electrical ports and the number of emitting e_3 electrical ports. Hence, the size of the EXC is equal to the maximum value between r and e.

4.4 Auxiliary Graph Model

In a traffic-groomable WDM network, there may be multiple ways to carry a low-speed connection request, *i.e.,* there may exist multiple routes from a given source node to a given destination node. Each of these routes may use different amount of network resources, *e.g.,* wavelength channels, grooming ports, etc. The decision on how to choose a proper route among multiple candidate routes is known as the grooming policy [C46]. Different grooming policies exist and reflect the way to engineer the traffic requests using the available network resources. For example, a low-speed connection can be carried through the existing lightpaths, by setting-up a new lightpath between the given source-destination node pair, or by a combination of both existing lightpaths and newly established lightpaths. According to the chosen policy, the network operator can achieve different objectives, such as minimizing the number of optical channels, minimizing the number of lightpaths, minimizing the number of hops on the virtual topology, etc. As the network state changes, the optimization objective may also vary. Grooming may be performed under various policies according to the network state. Such dynamic policy is also a challenge for traffic grooming.

In [C11, J62, B150], a generic graph model for traffic grooming in heterogeneous WDM mesh network has been proposed. The representation of a node in the graph model is determined by the node architecture. Different node architectures have different graph representations. Based on the auxiliary graph of each node, the authors construct an auxiliary graph for the entire network, compute a route in the auxiliary graph for a given traffic request, set-up the connection according to the route, and then update the auxiliary graph to reflect the changes in the network state. In this model, various factors of network heterogeneity are represented by different edges. The network heterogeneity may include the number of transceivers at each node, the number of wavelengths on each fiber-link, as well as the wavelength conversion capability and the grooming capability of each node. Besides, this model can achieve various objectives using different grooming policies. Moreover, instead of designing a route computation algorithm for each grooming policy, simple shortest-path route computation algorithms can be used in this model and can achieve various grooming policies by judiciously choosing the weight functions for the edges in the auxiliary graph.

In general, the physical topology of the network can be represented by a graph $G(V, E)$, where V is its set of nodes and E is its set of links. Based on this graph representation, we can construct an auxiliary graph $G'(V', E')$ for the whole network. In this new representation, V' denotes the set of network vertices and E' denotes the set of network edges. A key component of this representation is the nodal auxiliary graph. Thus, for each node i in the network ($i \in V$), we construct an auxiliary graph $G'_i(V'_i, E'_i)$ where V'_i is its set of nodal vertices and E'_i is its set of nodal edges. Subsequently, the network vertex set V' is the union of the vertex sets V'_i of the auxiliary graph of each node i in the network $G(V, E)$. Moreover, the network edge set E' is the union of the edge sets E'_i of the auxiliary graph of each node i in the network $G(V, E)$ and some additional edges. These additional edges represent the physical topology and the logical topology of the network, *i.e.,* the fiber-links and the already established lightpaths. From now on and for clarity reasons, we will use the terms *node* and *link* to represent a vertex and an edge in the original network graph $G(V, E)$, respectively, and the terms *vertex* and *edge* to represent a vertex and an edge in the network auxiliary graph $G'(V', E')$, respectively.

The network auxiliary graph $G'(V', E')$ can be divided into three layers, namely the "Access Layer", the "Lightpath Layer", and the "Wavelength Layer". The access layer (upper layer) represents the access points of a connection request, *i.e.,* the points where a customer's connection starts and terminates. The lightpath layer (middle layer) represents the grooming component of the network node (*e.g.,* EXC) where low-speed traffic streams are multiplexed (demultiplexed) onto (from) grooming lightpaths. The wavelength layer (lower layer) represents the wavelength switching capability (*e.g.,* OXC). When no wavelength conversion is allowed, the wavelength layer is divided into W sublayers; W being the number of wavelengths per fiber. A network node is divided into two vertices at each layer: a vertex marked with "I" and a vertex marked with "O". These two vertices represent the input port and the output port of the network node at this layer. As a result, a network node is represented by means of $(2 \times W + 4)$ vertices connected by a set of inter-node and intra-node edges. We associate to each edge in the auxiliary graph G' a property tuple $P(C_t, y_t)$, where C_t denotes the capacity limiting the use of this edge and y_t denotes the current load of the edge. Based on these information, a weight is computed and assigned to each edge. This weight reflects the current network state and a given grooming policy. The characteristics and purposes of these edges are as follows:

Type *'a'* edges: An ***'a'*** edge is an intra-node edge connecting the output vertex of the access layer to the output vertex of the lightpath layer at node i. This edge is used to add a sub-wavelength channel request at the node. Its capacity is limited by the number of e_1-ports present at this node.

Type *'b'* edges: A ***'b'*** edge is an intra-node edge connecting the output vertex of the lightpath layer to the output vertex of the wavelength layer/sublayer at node i. This edge represents the starting point of a GL and its capacity is limited by the number of e_3-ports/receiving o_3-ports present at this node.

Type *'c'* edges: A ***'c'*** edge is an intra-node edge connecting the output vertex of the access layer to the output vertex of the wavelength layer/sublayer at node i. This edge is used to add a full-wavelength channel request at the node. Its capacity is limited by the number of e_2-ports/receiving o_2-ports present at this node.

Type *'d'* edges: A ***'d'*** edge is an intra-node edge connecting the input vertex of the lightpath layer to the input vertex of the access layer at node i. This edge is used to drop a sub-wavelength channel request at the node. Its capacity is limited by the number of r_1-ports present at this node.

Type *'e'* edges: An ***'e'*** edge is an intra-node edge connecting the input vertex of the wavelength layer/sublayer to the input vertex of the lightpath layer at node i. This edge represents the ending point of a GL and its capacity is limited by the number of r_3-ports/emitting o_3-ports ports present at this node.

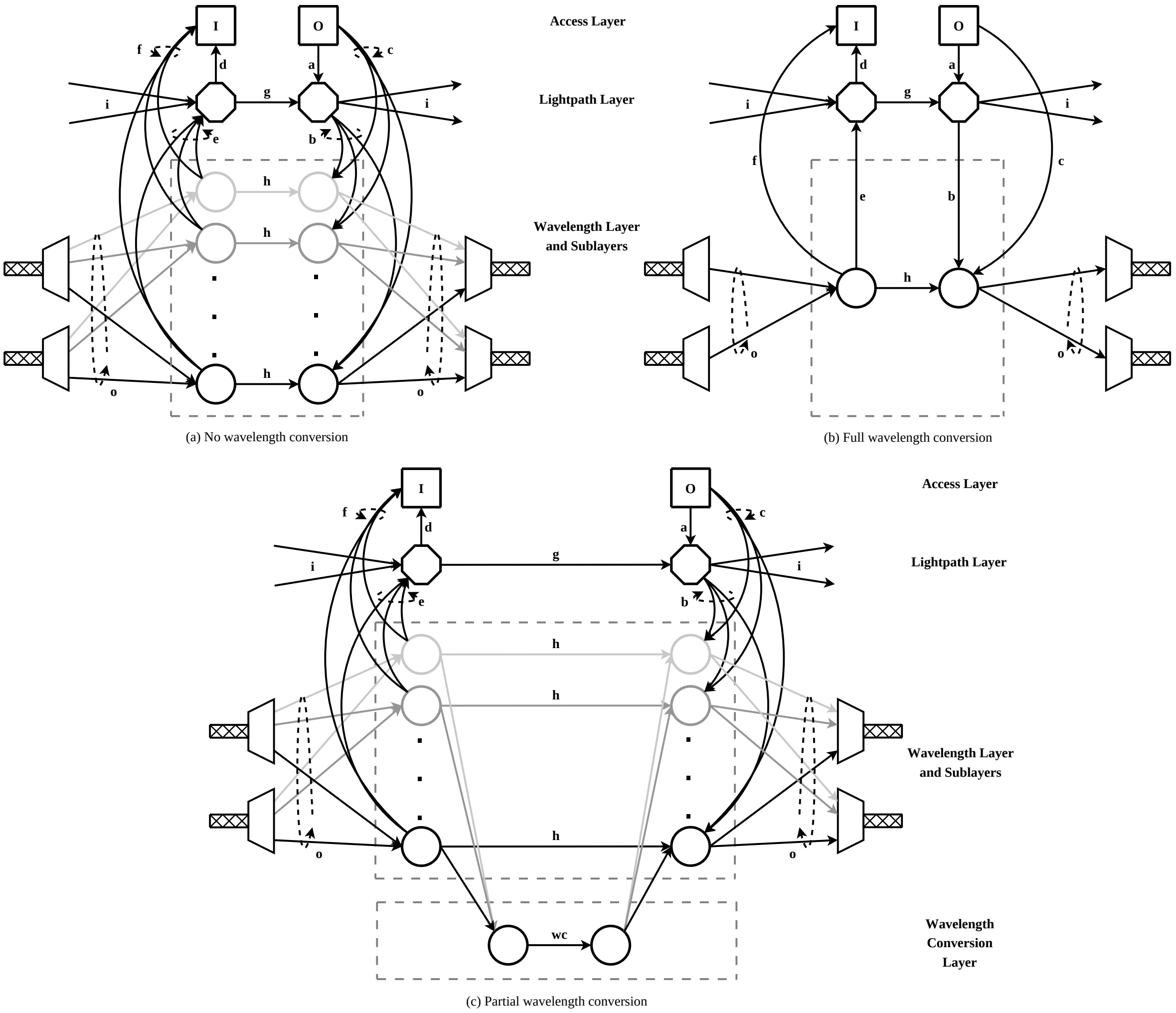
Type *'f'* edges: An ***'f'*** edge is an intra-node edge connecting the input vertex of the wavelength layer/sublayer to the input vertex of the access layer at node i. This edge is used to drop a full-wavelength channel request at the node. Its capacity is limited by the number of r_2-ports/emitting o_2-ports ports present at this node.

Type *'g'* edges: A ***'g'*** edge is an intra-node edge connecting the input vertex of the lightpath layer to the output vertex of the same lightpath layer at node i. This edge represents the grooming capability of the node.

Type *'h'* edges: An ***'h'*** edge is an intra-node edge connecting the input vertex of the wavelength layer/sublayer to the output vertex of the same wavelength layer/sublayer at node i. This edge represents the wavelength switching capability of the node.

Type *'i'* edges: An ***'i'*** edge is an inter-node edge connecting the output vertex of the lightpath layer at node i to the input vertex of the lightpath layer at node j. This edge represents a pre-established GL between these two nodes which are not necessarily adjacent. The capacity of this edge is equal to the rate that can be carried by the corresponding GL.

Type *'o'* edges: An ***'o'*** edge is an inter-node edge connecting the output vertex of the wavelength layer/sublayer at node i to the input vertex of the same wavelength layer/sublayer at an adjacent node j. This edge represents the presence of a physical fiber-link between these two nodes. Its capacity is limited by the number of o_1-ports present at each end of the physical link. The number of o_1-ports at the extremity of a fiber corresponds to the number of wavelengths in this fiber.



(a) No wavelength conversion

(b) Full wavelength conversion

(c) Partial wavelength conversion

Fig. 4.5. Auxiliary graph representation of a network node

Figure 4.5(a) represents the resulting auxiliary graph model corresponding to a network node with no wavelength conversion capability. This model can be easily extended to the case of full wavelength conversion and the case of partial wavelength conversion. When the network node has full wavelength conversion capability, the W wavelength sublayers of the auxiliary graph can be merged into a single wavelength layer. As a result, the graph representation of a network node is reduced to a set of 6 vertices connected by a set of 8 intra-node edges. The equivalent graphs of the network nodes are connected between them by a set of inter-node edges which yields the graph representation of the entire network. The auxiliary graph model of a network node with full wavelength conversion capability is given by Figure 4.5(b). Finally, when the network node has a limited-size pool of wavelength converters, the equivalent graph model of the node is almost identical to the graph representation of a node without any wavelength conversion capability. The major difference is the presence of a new layer representing the partial wavelength conversion capability. This layer is referred to as the "Wavelength Conversion Layer". Like the other layers, this layer is also composed of two vertices connected by a ***'wc'*** edge. This edge represents the wavelength conversion pool and its capacity is limited by the number of wavelength converters in the pool. The weight assigned to this edge can reflect the cost of using a wavelength converter. The input vertices of the W wavelength sublayers are connected to the input vertex of the wavelength conversion layer by edges of null cost. Similarly, the output vertex of the wavelength conversion layer is connected to the output vertex of each of the W wavelength sublayers by an edge of null cost. A node implementing partial wavelength conversion capability has an equivalent graph representation given by Figure 4.5(c).

Through all this thesis, we have considered nodes with partial wavelength conversion capability. Assuming that the size of the wavelength conversion pool has been well-dimensioned, we can achieve network performance comparable to that obtained using nodes with full wavelength conversion capability. As a result, we will use the auxiliary graph representation of a node with full wavelength conversion capability.

5

Routing and Grooming of Scheduled Demands (SxD)

Abstract: *On the basis of the multi-layer node architecture described in the previous chapter, we investigate the routing and grooming problem under scheduled traffic requests (SxDs). Both full-wavelength demands (SLDs) and sub-wavelength demands (SEDs) are considered. As a first step, we illustrate by means of simple examples the principle of resource reutilization inherent to SxDs routing as well as the principle of SEDs traffic grooming. We then provide, as a second step, the mathematical formulation of the network cost applied to the routing and grooming problem. This cost is expressed in terms of the number of required electrical ports and optical ports. As a third step, we propose a mathematical ILP formulation of the SLDs and SEDs routing and grooming problem. Due to the complexity of this approach, we describe an original heuristic approach for this same problem. Finally, we compare the performance of the heuristic approach with the exact approach by means of numerical results.*

5.1 Introduction

As WDM technology advances, the capacity of a wavelength channel continues to increase. However, the bandwidth requirement of a typical connection is versatile and corresponds usually to a small fraction of the bandwidth of a WDM channel. To efficiently use the bandwidth offered by an optical channel, grooming switches have been developed. These grooming switches must be able to unpack a wavelength channel to its finer granularity, switch these fin granularities independently, and finally pack them back onto wavelength channels. Grooming is carried out by means of electrical switches which need to be equipped with transponders to perform electro-optical conversion. On the other hand, if only all-optical wavelength switches are deployed instead of electrical grooming switches, we do not need transponders at the input and the output ports of these switches. Based on the current technology, such all-optical wavelength switches induce significant savings in optical transport networks at the price of no grooming capability and wasted channel capacity. In order to satisfy various traffic requests of different bandwidth granularities, each network node has two switching fabrics. The first is the Optical Cross-Connect (OXC) which is capable of switching wavelengths in the optical domain. The second is the Electrical Cross-Connect (EXC) which is capable of aggregating/de-aggregating traffic streams in the electrical domain. The detailed architecture of such a network node is given in

Chapter 4. The main challenge is to dimension a network able to handle a given set of traffic requests at the lowest cost. The overall cost of the network ζ is evaluated as a function of the number ϱ of OXC optical ports and the number ε of EXC electrical ports as follows:

$$\zeta = \varrho + \kappa.\varepsilon \tag{5.1}$$

The coefficient κ (in front of ε) expresses the fact that an electrical port is κ times more expensive than an optical port. The network design problem is thus reduced to the problem of determining the size of the OXC and the size of the EXC at each node of the network in order to satisfy all the traffic requests. To date, topology-design algorithms have considered static traffic demands based on peak traffic between network nodes. Such static design methods over-allocate the bandwidths in the case of dynamic traffic. Our objective is to exploit the dynamics of the traffic to reconfigure connection bandwidth when necessary. For this task, we have mainly considered Scheduled Demands (SxDs = SLDs + SEDs). The SxDs are pre-planned demands with pre-determined date of arrival, life duration, and capacity. This type of traffic request is already introduced in Section 3.4.1. Due to the fact that SxDs are known in advance, they are said to be deterministic and are routed mainly by means of the management plane using a global optimization tool performed off-line. Taking into account their time and space correlation, this off-line tool determines the best routing for the SxDs. The time correlation of the SxDs has a significant impact on the cost of an instance of the routing solution. In fact, a set of SxDs with significant time disjointness should ease the reuse of WDM channels and, as a consequence, should lead to a small number of required electrical ports and optical ports. Conversely, a set of SEDs with significant time correlation should promote the use of grooming. To sum up, when trying to route a set of SLDs, the saving in network cost is obtained by means of channel reuse thanks to the knowledge of time and space correlation between the requests. However, when trying to route a set of SEDs, the cost saving is obtained by means of channel reuse as well as by means of traffic grooming. The cost benefit obtained by means of resource reutilization as well as by means of grooming schemes are best highlighted using examples.

Example 5.1.

In this first example, we want to highlight the impact of resource sharing on the network cost. For this task, we have considered a set of 3 SLDs. The characteristics of these requests are given in Table 5.1, and their time diagram is shown in Figure 5.1.

Table 5.1. Characteristics of the 3 SLDs

δ_i	s_i	d_i	α_i	β_i	n_i
δ_1	2	8	8:00	14:40	$1 \times C_\omega$
δ_2	3	7	11:00	13:00	$1 \times C_\omega$
δ_3	2	6	17:00	19:30	$1 \times C_\omega$

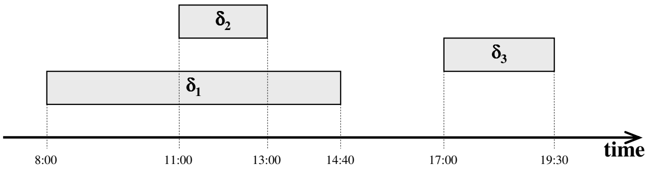
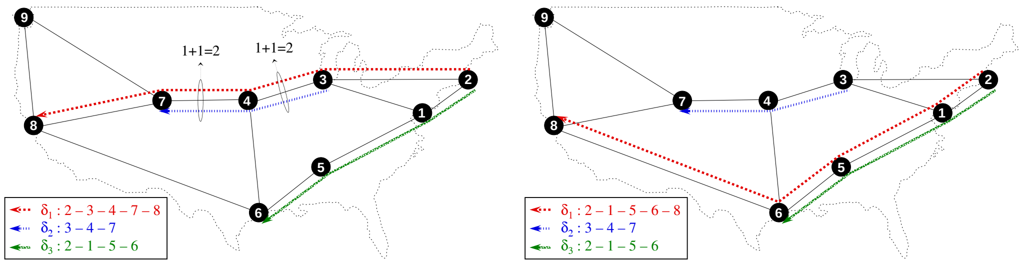


Fig. 5.1. Associated time diagram of the 3 SLDs

When routing SLDs, emitting e_2 electrical ports, receiving r_2 electrical ports, and o_2 optical ports are only needed to add and to drop the traffic requests. Thus, the number of such ports is independent of the routing solution, and subsequently it remains invariant. Consequently, the network cost is a function of only the number of required o_1 optical ports which is proportional to the number of WDM optical channels used in the network (the number of required o_1 optical ports is twice the number of WDM optical channels used). When routing the SLDs along the shortest path between their source and their destination nodes (Figure 5.2(a)), we need two WDM optical channels on each of the 3-4 and the 4-7 fiber-links. This is due to the fact that the SLDs δ_1 and δ_2 are both active from 11:00 to 13:00. As a result, this routing solution requires a total amount of 9 WDM optical channels. However, when changing the path assigned to the SLD δ_1 from its shortest path to the path 2-1-5-6-8 (Figure 5.2(b)), the WDM channels on the links 2-1, 1-5, and 5-6 used from 08:00 to 14:40 by the SLD δ_1 can be reused by the SLD δ_3 from 17:00 to 19:30. Hence, only 6 WDM optical channels are needed for this new routing solution.



(a) Routing along the shortest path

(b) Another routing solution

Fig. 5.2. Two possible routing solutions for the set of 3 SLDs

---

Example 5.2.

In this second example, we want to highlight the gain brought by the grooming process on the network cost. For this task, we have considered a set of 3 SEDs. The characteristics of these requests are given in Table 5.2, and their time diagram is shown in Figure 5.3.

Table 5.2. Characteristics of the 3 SEDs

δ_i	s_i	d_i	α_i	β_i	n_i
δ_1	2	8	8:00	14:40	$0.25 \times C_\omega$
δ_2	3	7	11:00	13:00	$0.25 \times C_\omega$
δ_3	2	6	17:00	19:30	$0.25 \times C_\omega$

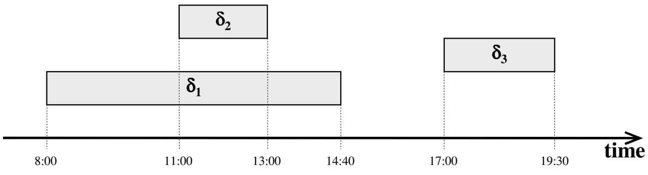
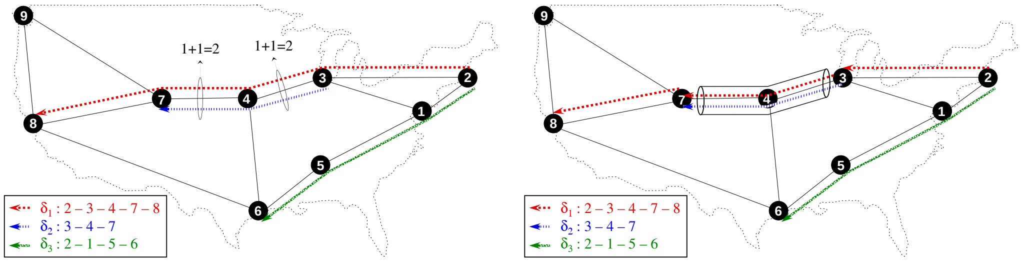


Fig. 5.3. Associated time diagram of the 3 SEDs

In order to emphasize the benefit of grooming, we have considered a fixed routing solution where each SED request is routed along its shortest path. Emitting e_1 electrical ports and receiving r_1 electrical ports are only needed to add and to drop the traffic requests. Consequently, the number of such ports is independent of the routing and grooming solution, and subsequently it remains invariant. When the network does not have any grooming capability (Figure 5.4(a)), the SEDs δ_1 and δ_2 are routed over distinct lightpaths. Thus, we need two WDM optical channels on each of the 3-4 and the 4-7 fiber-links. As a result, this routing and grooming solution requires a total amount of 9 WDM optical channels. Let us recall that the number of required o_1 optical ports is twice the number of WDM optical channels used. On the other hand, when grooming is enabled in the network (Figure 5.4(b)), the SEDs δ_1 and δ_2 can be groomed together since both requests are active from 11:00 to 13:00 and they share a common path between node 3 and node 7. Therefore, a grooming lightpath is created between node 3 and node 7 along the path 3-4-7 using only one WDM optical channel on each of the 3-4 and the 4-7 fiber-links. This grooming lightpath is used to carry simultaneously the two SEDs δ_1 and δ_2 between node 3 and node 7. In our example, the SED δ_1 still needs to be routed using two additional lightpaths; the first lightpath is between node 2 and node 3 along the path 2-3, while the second lightpath is between node 7 and node 8 along the path 7-8. As a result, only 7 WDM optical channels are needed for this new routing and grooming solution. However, it should be noted that in order to proceed to the aggregation of the SEDs δ_1 and δ_2 at node 3, one additional emitting o_3 optical port and one additional receiving r_3 electrical port are needed at this node. Similarly, in order to proceed to the de-aggregation of these two SEDs at node 7, one additional emitting e_3 electrical port and one additional receiving o_3 optical port are needed at this node. Depending on the relative costs of the different o_1, o_3, e_3, and r_3 ports, the new routing and grooming solution can be more or less economical than the first solution.



(a) Shortest path without any grooming functionality

(b) Shortest path with grooming functionality

Fig. 5.4. Two possible routing and grooming solutions for the set of 3 SEDs

---

In this chapter, we investigate the problem of efficiently designing a WDM mesh network with grooming capability for a given set of traffic requests via theoretical formulation as well as simulation experiments. The results of our research indicate that, through careful network design, we can dimension a network able to satisfy the totality of a traffic set while significantly reducing the network cost. In the remaining of this thesis, the term “routing” will refer to the routing operation in the case of Lightpath Demands (xLDs = SLDs + RLDs + sRLDs) as well as to the routing and the grooming operations in the case of Electrical Demands (xEDs = SEDs + REDs + sREDs).

5.2 Grooming

The characteristic of the xEDs lies in the fact that their flow is smaller than the capacity of an optical channel. This particularity is taken into account by use of electrical aggregation. The grooming of multiple xEDs consists in multiplexing their electrical flows onto the same GL within an EXC.

Grooming requires all the considered xEDs to share at least one common fiber-link and all of them to be active during a common period of time. In this section, we consider an example to illustrate the xEDs grooming process and the number of lightpaths involved in such an operation. As a first step, we consider the grooming of two individual xEDs. We then generalize this process to the case of multiple xEDs.

Grooming of Two Demands

Let us consider two xED traffic requests δ_1 and δ_2 with the following characteristics:

- Source node: s_1 & s_2, resp.
- Destination node: d_1 & d_2, resp.
- Set-up time: α_1 & α_2, resp.
- Tear-down time: β_1 & β_2, resp.
- Rate: n_1 & n_2, resp.

Let us assume that the xED δ_1 is routed along the physical path $s_1 - \cdots - g_1 - \cdots - g_2 - \cdots - d_1$, while the xED δ_2 is routed along the physical path $s_2 - \cdots - g_1 - \cdots - g_2 - \cdots - d_2$. The time diagram of these two requests as well as their routing solution are shown in Figure 5.5. From these characteristics, it can be noticed that both requests are active from α_2 to β_1, and they have a common path between node g_1 and node g_2. Hence, they verify the grooming constraints and can be groomed together if the network allows traffic aggregation.

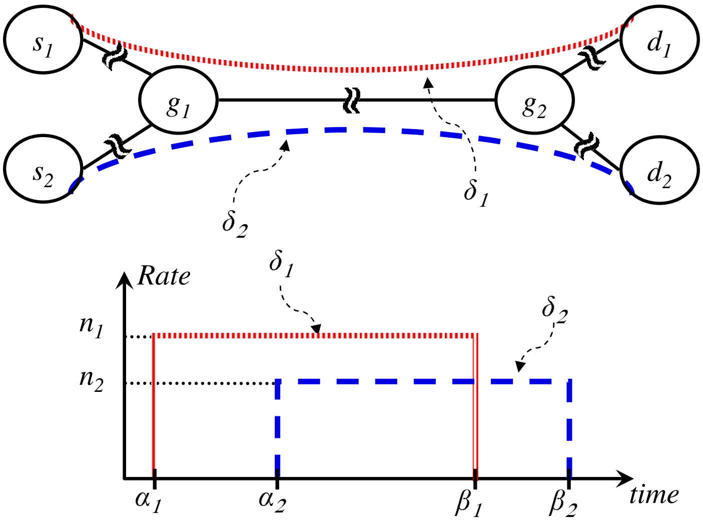


Fig. 5.5. Time diagram of two xED traffic requests

When the network does not provide any grooming functionality, each xED is routed using a separate lightpath between its source node and its destination node (Figure 5.6). In other words, the xED δ_1 is routed using the lightpath $\wp_1$ between node s_1 and node d_1, while the xED δ_2 is routed using the lightpath $\wp_2$ between node s_2 and node d_2. As a result, two optical channels are needed on each fiber-link between node g_1 and node g_2.

On the other hand, when grooming is enabled in the network, the xEDs δ_1 and δ_2 can be groomed together between node g_1 and node g_2 if the accumulated capacity is less than the capacity of an optical channel ($n_1 + n_2 \leqslant C_\omega$) (Figure 5.7). Theoretically, this aggregation must yield the aggregated

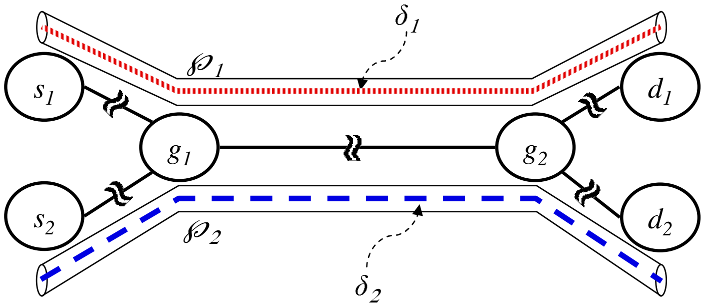


Fig. 5.6. Network without any grooming functionality

demand δ_a characterized by $\delta_a\ (g_1, g_2, \alpha_2, \beta_1, n_1 + n_2)$. This aggregated demand δ_a is carried by the network using the GL $\wp_a$. However, as the xED δ_1 is also active during the period $[\alpha_1, \alpha_2]$ and the xED δ_2 is also active during the period $[\beta_1, \beta_2]$, two additional requests must be considered. The first additional request δ_f given by $\delta_f\ (g_1, g_2, \alpha_1, \alpha_2, n_1)$ takes into account the remaining active period of the xED δ_1. Similarly, the remaining active period of the xED δ_2 is represented by the additional request δ_g characterized by $\delta_g\ (g_1, g_2, \beta_1, \beta_2, n_2)$. Both δ_f and δ_g are carried by the GL $\wp_a$. In addition, we still need to take into account the non common segments of the paths of the xEDs δ_1 and δ_2. This can be done by introducing four additional traffic requests δ_b, δ_c, δ_d, and δ_e. These additional traffic requests are defined as follows:

- The request δ_b represents a segment of the path of δ_1 totally disjoint from the path of δ_2. It connects the source node s_1 of the xED δ_1 to the grooming node g_1. This request is defined by $\delta_b\ (s_1, g_1, \alpha_1, \beta_1, n_1)$ and is carried by the GL $\wp_b$.
- The request δ_c represents another segment of the path of δ_1 totally disjoint from the path of δ_2. It connects the grooming node g_2 to the destination node d_1 of the xED δ_1. This request is defined by $\delta_c\ (g_2, d_1, \alpha_1, \beta_1, n_1)$ and is carried by the GL $\wp_c$.
- The request δ_d represents a segment of the path of δ_2 totally disjoint from the path of δ_1. It connects the source node s_2 of the xED δ_2 to the grooming node g_1. This request is defined by $\delta_d\ (s_2, g_1, \alpha_2, \beta_2, n_2)$ and is carried by the GL $\wp_d$.
- The request δ_e represents another segment of the path of δ_2 totally disjoint from the path of δ_1. It connects the grooming node g_2 to the destination node d_2 of the xED δ_2. This request is defined by $\delta_e\ (g_2, d_2, \alpha_2, \beta_2, n_2)$ and is carried by the GL $\wp_e$.

To sum up, grooming together the requests δ_1 and δ_2 yields the aggregated request δ_a mainly, and a set of six additional requests. Some of these requests (δ_b, δ_c, δ_d, and δ_e) take into account the non common segments of the paths of δ_1 and δ_2. The remaining requests (δ_f and δ_g) take into account the non common periods of time. According to the characteristics of the initial requests δ_1 and δ_2, some of these additional requests may not exist. However, those which do exist form the set of marginal demands. As a result of the grooming process, when grooming two SED requests, the set of all the SEDs must be modified by adding the aggregated request δ_a and the set of marginal demands and by removing the original requests δ_1 and δ_2.

At node g_1 and in order to be groomed together, the two demands δ_1 and δ_2 must be switched from the OXC to the EXC. Hence, two receiving r_3 electrical ports and two emitting o_3 optical ports are needed at this node. Once groomed together, the resulting GL $\wp_a$ is first switched to the OXC using an emitting e_3 electrical port and a receiving o_3 optical port, and then it is redirected towards node g_2. Arriving at node g_2, the GL $\wp_a$ is switched back from the OXC to the EXC using an emitting

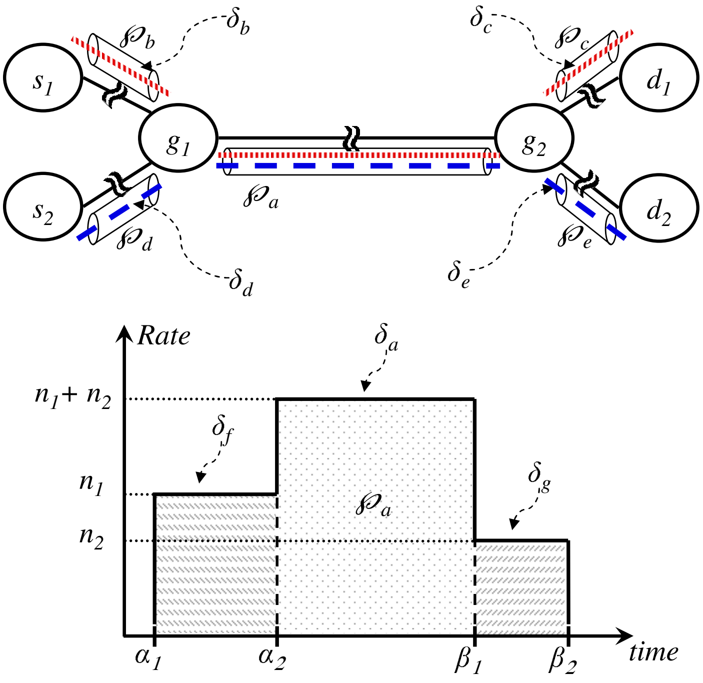


Fig. 5.7. Network with grooming functionality

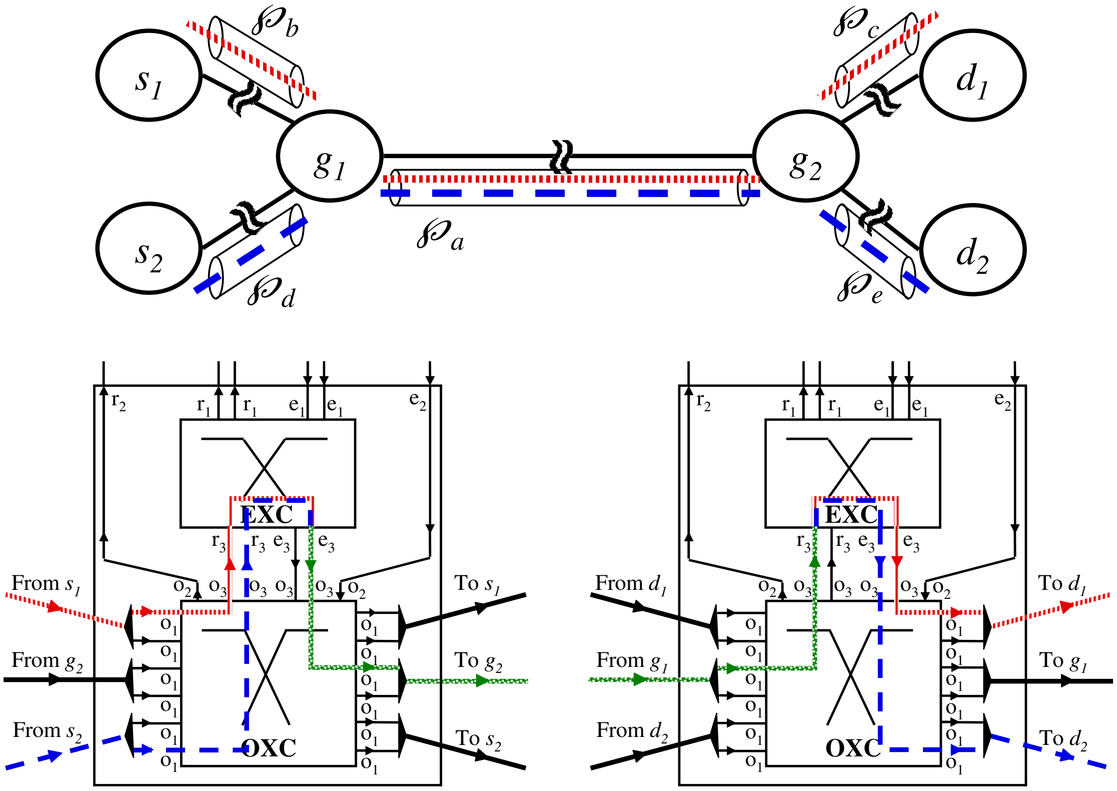
o_3 optical port and a receiving r_3 electrical port. At the EXC of node g_2, the initial requests δ_1 and δ_2 are rebuilt. These two requests are first switched to the OXC using two emitting e_3 electrical ports and two receiving o_3 optical ports, and then they are redirected to their respective destination nodes. In brief, in order to groom these two requests, six additional electrical ports and six additional optical ports are to be provided. Figure 5.8 shows the detailed description of the grooming operation and the ports involved in such a process.

Grooming of Multiple Demands

Now let us consider the case where we try to groom multiple demands. For example and without loss of generality, we consider three traffic requests that share a same path between node g_1 and node g_2 during a period of time θ. We are going to show that simultaneously grooming together these three requests is equivalent to successively groom the first two requests alone, then the resulting aggregated demand with the third request.

Theoretically, in order to be groomed together, the three traffic requests must be switched from the OXC to the EXC at node g_1, hence three receiving r_3 electrical ports and three emitting o_3 optical ports are needed at this node. Similarly, in order to restitute these three traffic requests, three emitting e_3 electrical ports and three receiving o_3 optical ports are needed at node g_2. Also, the resulting aggregated demand requires electrical (e_3/r_3) and optical (o_3) ports at both ends. In brief, in order to groom these three demands, eight additional electrical ports (e_3/r_3) and eight additional optical o_3 ports are to be provided. The gain obtained thanks to the grooming is the reduction of the number of o_1 optical ports to be provided on the common path between node g_1 and node g_2.

Another approach is to consider the requests two by two. The iterative nature of the proposed approach insures that multiple requests are groomed together. In our case study and as a first step, we groom together two traffic requests chosen randomly among the three. As a result, these two requests are switched from the OXC to the EXC at the node g_1 and the node g_2, and a grooming



(a) Grooming process at node g_1

(b) Grooming process at node g_2

Fig. 5.8. Detailed description of the grooming operation and the ports involved in such a process

lightpath is created between these two nodes. Thus, six additional electrical ports and six additional optical ports are to be provided (*c.f.* Section 5.2). As a second step, the third remaining traffic request is groomed with the two previous traffic requests. In this case, the first two requests are already present at the EXC level of node g_1 and node g_2, and the grooming lightpath is already created. Thus, the only need is to switch the third demand from the OXC to the EXC at the node g_1 and the node g_2 in order to be groomed with the other two requests. Thus, two additional electrical ports and two additional optical ports are to be provided. This result shows that grooming multiple traffic requests onto the same GL can be achieved by a series of consecutive grooming operations where the requests are considered two by two.

In this example, we have assumed that the three xEDs can be groomed onto a single GL since the accumulated capacity of the three traffic requests is less than the capacity of an optical channel ($n_1 + n_2 + n_3 \leqslant C_\omega$). However, when $C_\omega < n_1 + n_2 + n_3 \leqslant 2 \times C_\omega$, one needs to consider the combinatorics of the 3 xEDs considered two by two, and then to consider for each combination the resulting groomed demand with the third remaining xED. In this case, 3 grooming combinations are possible. The combination with the lowest cost is the one to be accepted. In the more general case with $\aleph$ xED requests to be groomed two by two, the number $\digamma$ of possible combinations increases drastically according to the following equation:

$$\digamma = \frac{\aleph!(\aleph-1)!}{2^{\aleph-1}} \tag{5.2}$$

5.3 Description of the Problem

Traffic grooming refers to various problems including network planning, topology design, and dynamic circuit provisioning. The traffic grooming problem based on scheduled (predictable) dynamic traffic requests is essentially an optimization problem. The problem description is as follows: for a given set of scheduled traffic demands and a given physical topology, compute the size of the OXCs and the EXCs in order to satisfy all the demands as well as minimize the network cost. Typically, the network cost is measured by the cost of the optical ports and the electrical ports used in the network. Another criterion can also be used; it consists in minimizing the congestion observed in the network. The congestion is defined as the number of wavelengths or active optical channels used on the most loaded link. In this way, we intend to spread out the traffic load homogenously over all the links of the network. The solution of this problem needs also to determine how to route, groom, and assign wavelengths to the traffic demands.

In this chapter, we consider the network design problem to satisfy a given set of SxDs (SLDs and SEDs as they have been defined in Chapter 3). In this context, cost optimization consists in considering the impact of time and space correlation as well as the impact of traffic grooming on the number of required electrical ports and optical ports. Another significant parameter that could impact the network cost is the signaling procedure. In general, under dynamic traffic, a signaling channel is required to configure in real time the network nodes. Since this chapter only deals with dynamic preplanned traffic, we can neglect the impact of signaling on connection establishment duration. Let us recall that under dynamic but scheduled traffic, connection establishment is carried out by the management plane. However, under dynamic random traffic demands, connection establishment must be carried out by the control plane.

Previous investigations in our laboratory have presented solutions to the network design problem for SLD traffic requests [C21, C47, D156]. In this chapter, we extend this work to consider a mix of SLDs and SEDs, several SEDs being groomed onto dedicated GLs. Such an aggregation implies the use of additional electrical ports within the network. Our optimization problem is then based on a trade-off between the cost penalty due to additional electrical ports and the cost benefit due to the reduction in the number of optical ports. In our study, we make the following assumptions:

- The network is a single-fiber irregular mesh network, *i.e.,* there is at most one pair of fiber-links (⇒ bidirectional) between each node pair.
- We have considered nodes with partial wavelength conversion capability. Assuming that the size of the wavelength conversion pool has been well-dimensioned, we can achieve network performance comparable to that obtained using nodes with full wavelength conversion capability. Hence, as a first approach, we consider a network with full wavelength conversion capability. Afterwards, we determine the size of the wavelength converters pool and we assign wavelengths to the lightpaths in order to satisfy all the requests at the lowest cost.
- The transceivers in the network are tunable to any wavelength in the fiber.
- A connection request (either SLD or SED) cannot be divided into several lower-speed connections that are routed separately using different physical paths from their source node to their destination node. The data flow of a connection request may only be split onto multiple optical channels following always the same route.
- A sub-wavelength connection request (SED) can traverse multiple grooming lightpaths before it reaches its destination node. So, an SED may be groomed with different SEDs on different GLs.

5.4 Mathematical Formulation

The design of a grooming-capable network is formally stated below. The inputs to the problem are:

- A physical topology represented by an equivalent graph $G = (V, E, w)$. G is an arc-weighted symmetrical directed graph where $V = \{v_1, v_2, \cdots, v_N\}$ is the set of network nodes and $E = \{e_1, e_2, \cdots, e_L\}$ is the set of physical links interconnecting these nodes. The nodes correspond to the EXC/OXC multi-layer switches and the links correspond to the fibers of the network. As the links are unidirectional, we assume that there is a single fiber linking two nodes in each direction. The links are assigned weights $w : E \mapsto \mathbb{R}^+$ which may correspond to the physical length or the cost of the links. $N = |V|$ and $L = |E|$ are, respectively, the number of nodes and the number of links in the network.
- A set $\mathcal{D} = \{\delta_1, \delta_2, \cdots, \delta_M\}$ of M SxD requests which require to be routed on the network. As previously introduced in Chapter 3, these requests can be divided into a set $\mathcal{D}^L = \{\delta_1^L, \delta_2^L, \cdots, \delta_{M^L}^L\}$ of M^L SLD requests and another set $\mathcal{D}^E = \{\delta_1^E, \delta_2^E, \cdots, \delta_{M^E}^E\}$ of M^E SED requests. Each request, whether it is an SLD (δ_i^L) or an SED (δ_i^E), is represented by a tuple $\big(s_i, d_i, \alpha_i, \beta_i, n_i\big)$ where $s_i, d_i \in V$ are the source and the destination nodes of the demand, α_i and β_i are its set-up and its tear-down dates, and n_i is the traffic rate requested by the connection. It is to be noted that for SLDs, n_i is a unit value, while for SEDs, n_i is a real number less than 1.
- A set of available routes $\mathcal{P}_i = \{P_{i,1}, P_{i,2}, \cdots, P_{i,K}\}$ connecting the source node s_i to the destination node d_i of the request δ_i^{LorE}. For each request, we compute beforehand K-alternate shortest paths in terms of effective length connecting its source node to its destination node according to the algorithm described in [C48] (if as many paths exist, otherwise we consider the available ones).

The set-up/tear-down dates of all these requests are to be taken into account because they represent time instants when a change in the traffic flow is observed. Let $\mathcal{E}$ be the ordered set grouping the arrival/extinction time of all the requests in the set $\mathcal{D}$. Because some requests may have the same starting/ending time, the number $\mathbb{E}$ of time instants in the set $\mathcal{E}$ ($\mathbb{E} = |\mathcal{E}|$) is less than or equal to $2 \times M$.

$$\begin{aligned}\mathcal{E} &= \left(\bigcup_{i=1}^{M} \alpha_i\right) \bigcup \left(\bigcup_{i=1}^{M} \beta_i\right)\\ &= \{\epsilon_1, \epsilon_2, \cdots, \epsilon_{\mathbb{E}}\} \qquad \text{such as } \epsilon_1 < \epsilon_2 < \cdots < \epsilon_{\mathbb{E}} \end{aligned} \tag{5.3}$$

When the set $\mathcal{D}$ is composed of only SLD requests, the tear-down dates can be omitted from the set $\mathcal{E}$ since no grooming is carried out in the network. In this case, the number $\mathbb{E}$ of time instants in the set $\mathcal{E}$ ($\mathbb{E} = |\mathcal{E}|$) is less than or equal to M. Equation (5.3) is resolved into:

$$\mathcal{E} = \bigcup_{i=1}^{M} \alpha_i = \{\epsilon_1, \epsilon_2, \cdots, \epsilon_{\mathbb{E}}\} \qquad \text{such as } \epsilon_1 < \epsilon_2 < \cdots < \epsilon_{\mathbb{E}} \tag{5.4}$$

In the case of SLDs, the solution of the optimization problem consists in assigning a path $P_{i,\rho_i} \in \mathcal{P}_i$ to the request δ_i^L ($1 \leqslant i \leqslant M^L$). P_{i,ρ_i} is the ρ_i^{th} shortest path between s_i and d_i ($1 \leqslant \rho_i \leqslant K$). This solution is represented by $\varPi_{\mathcal{D}^L,\rho} = \{P_{1,\rho_1}, P_{2,\rho_2}, \cdots, P_{M^L,\rho_{M^L}}\}$. In the case of SEDs, we must first assign a physical path $P_{i,\rho_i} \in \mathcal{P}_i$ to the request δ_i^E ($1 \leqslant i \leqslant M^E$). Once the physical path P_{i,ρ_i} is chosen, we need, as a second step, to determine at which intermediate nodes of P_{i,ρ_i}, the SED δ_i^E must be switched to the electrical level for grooming. By this way, we obtain the set of grooming lightpaths that the request traverses through in the logical topology. A physical path P_{i,ρ_i} that is composed of ℓ_{i,ρ_i} hops has $(\ell_{i,\rho_i} - 1)$ intermediate nodes and can be decomposed into $\mathbb{L}_{i,\rho_i} = 2^{\ell_{i,\rho_i}-1}$ different sets of grooming lightpaths. Let $\mathcal{L}_{i,\rho_i} = \big\{L_{i,\rho_i,1}, L_{i,\rho_i,2}, \cdots, L_{i,\rho_i,\mathbb{L}_{i,\rho_i}}\big\}$ be the set of all possible decompositions of the path P_{i,ρ_i} into its inherent sets of grooming lightpaths. The solution of the optimization problem consists in assigning to the request δ_i^E a physical path $P_{i,\rho_i} \in \mathcal{P}_i$ and a set of grooming lightpaths $L_{i,\rho_i,\lambda_{i,\rho_i}} \in \mathcal{L}_{i,\rho_i}$. As in the case of SLDs, P_{i,ρ_i} is the ρ_i^{th} shortest path between s_i and d_i $(1 \leqslant \rho_i \leqslant K)$. However, $L_{i,\rho_i,\lambda_{i,\rho_i}}$ is the λ_{i,ρ_i}^{th} possible decomposition of the physical path P_{i,ρ_i} into a set of grooming lightpaths $(1 \leqslant \lambda_{i,\rho_i} \leqslant \mathbb{L}_{i,\rho_i})$. For example, let us consider the physical path $P_{i,\rho_i} = 1-2-3-4$ between nodes 1 and 4. This path is 3 hops long and can be decomposed into $2^{3-1} = 4$ different sets of grooming lightpaths as follows:

- $L_{i,\rho_i,1} = \big\{[1-2-3-4]\big\}$
- $L_{i,\rho_i,2} = \big\{[1-2], [2-3-4]\big\}$
- $L_{i,\rho_i,3} = \big\{[1-2-3], [3-4]\big\}$
- $L_{i,\rho_i,4} = \big\{[1-2], [2-3], [3-4]\big\}$

The solution to the optimization problem is represented by $\mathit{\Pi}_{\mathcal{D}^E,\rho,\lambda} = \big\{(P_{1,\rho_1}, L_{1,\rho_1,\lambda_{1,\rho_1}}), (P_{2,\rho_2}, L_{2,\rho_2,\lambda_{2,\rho_2}}), \cdots, (P_{M^E,\rho_{M^E}}, L_{M^E,\rho_{M^E},\lambda_{M^E,\rho_{M^E}}})\big\}$.

Our objective is to satisfy all the requests at the lowest cost. The network cost is expressed in terms of the number of electrical ports and optical ports used and can be obtained by means of mathematical operations based on matrix algebra. In the following, we detail the network cost computation for both SLDs and SEDs.

5.4.1 Network Cost Computation in the Case of SLDs

For an instance $\mathit{\Pi}_{\mathcal{D}^L,\rho} = \{P_{1,\rho_1}, P_{2,\rho_2}, \cdots, P_{M^L,\rho_{M^L}}\}$ of the network design problem under SLD traffic requests, we define the following matrices:

Request Matrix: The request matrix, denoted by $\underline{\mathit{\Delta}}^L = \{\delta_{i,t}^L\}$, represents the SLD traffic requests over time. $\underline{\mathit{\Delta}}^L$ is a $[M^L \times \mathbb{E}]$ matrix. An element $\delta_{i,t}^L$ of such a matrix is a binary value $(\delta_{i,t}^L \in \{0,1\})$ specifying the presence $(\delta_{i,t}^L = 1)$ or the absence $(\delta_{i,t}^L = 0)$ of the SLD request $\delta_i^L(s_i, d_i, \alpha_i, \beta_i, 1.0)$ at time instant ϵ_t.

$$\delta_{i,t}^L = \begin{cases} 1 & \text{if } \alpha_i \leqslant \epsilon_t < \beta_i, \\ 0 & \text{otherwise.} \end{cases} \tag{5.5}$$

Routing Matrix: The routing matrix, denoted by $\underline{R}^L = \{r_{i,l}^L\}$, represents the use of the physical links by the SLD requests. $\underline{R}^L$ is a $[M^L \times L]$ matrix. An element $r_{i,l}^L$ of such a matrix is a binary value $(r_{i,l}^L \in \{0,1\})$ specifying if the physical path P_{i,ρ_i} assigned to the SLD request $\delta_i^L(s_i, d_i, \alpha_i, \beta_i, 1.0)$ passes through the physical link e_l $(r_{i,l}^L = 1)$ or not $(r_{i,l}^L = 0)$.

$$r_{i,l}^L = \begin{cases} 1 & \text{if } e_l \in P_{i,\rho_i}, \\ 0 & \text{otherwise.} \end{cases} \tag{5.6}$$

Source Matrix: The source matrix, denoted by $\underline{S}^L = \{s_{i,n}^L\}$, represents the source nodes of the SLD requests. $\underline{S}^L$ is a $[M^L \times N]$ matrix. An element $s_{i,n}^L$ of such a matrix is a binary value $(s_{i,n}^L \in \{0,1\})$ specifying if the SLD request $\delta_i^L(s_i, d_i, \alpha_i, \beta_i, 1.0)$ has node v_n as a source node $(s_{i,n}^L = 1)$ or not $(s_{i,n}^L = 0)$.

$$s_{i,n}^L = \begin{cases} 1 & \text{if } s_i = v_n, \\ 0 & \text{otherwise.} \end{cases} \tag{5.7}$$

Destination Matrix: The destination matrix, denoted by $\underline{D}^L = \{d^L_{i,n}\}$, represents the destination nodes of the SLD requests. $\underline{D}^L$ is a $[M^L \times N]$ matrix. An element $d^L_{i,n}$ of such a matrix is a binary value ($d^L_{i,n} \in \{0,1\}$) specifying if the SLD request $\delta^L_i(s_i, d_i, \alpha_i, \beta_i, 1.0)$ has node v_n as a destination node ($d^L_{i,n} = 1$) or not ($d^L_{i,n} = 0$).

$$d^L_{i,n} = \begin{cases} 1 & \text{if } d_i = v_n, \\ 0 & \text{otherwise.} \end{cases} \tag{5.8}$$

By multiplying the transposed request matrix $\underline{\Delta}^{L^T}$ by the routing matrix $\underline{R}^L$, a new matrix $\underline{\Phi}^L = \{\phi^L_{t,l}\}$ is obtained. $\underline{\Phi}^L$, of dimension $[\mathbb{E} \times L]$, represents the traffic load over time carried by the links. An element $\phi^L_{t,l}$ of such a matrix is an integer value representing the number of WDM optical channels carried by link e_l at instant ϵ_t. For a given link e_l, let φ^L_l be the maximum value of this traffic $\phi^L_{t,l}$ evaluated over the whole observation period. φ^L_l gives the number of WDM optical channels used on this link. As the o_1 optical ports go by pair, one port at each end of a WDM channel, φ^L_l represents also the number of o_1 optical port pairs installed at both ends of the link e_l. The number $\gamma^L_{o_1}$ of o_1 optical ports in the network is twice the sum of φ^L_l over all the network links. The maximum value of φ^L_l ($\forall l \backslash 1 \leqslant l \leqslant L$) represents the congestion in the network, *i.e.,* the number of active channels on the most loaded link.

$$\underline{\Phi}^L = \underline{\Delta}^{L^T} \times \underline{R}^L \tag{5.9a}$$

$$\varphi^L_l = \max_{1 \leq t \leq \mathbb{E}} \phi^L_{t,l} \tag{5.9b}$$

$$\gamma^L_{o_1} = 2 \times \sum_{l=1}^{L} \varphi^L_l \tag{5.9c}$$

By multiplying the transposed request matrix $\underline{\Delta}^{L^T}$ by the source matrix $\underline{S}^L$, a new matrix $\underline{\Psi}^{L,Em} = \{\psi^{L,Em}_{t,n}\}$ is obtained. $\underline{\Psi}^{L,Em}$, of dimension $[\mathbb{E} \times N]$, represents the use of full-wavelength 'add' ports at the nodes. An element $\psi^{L,Em}_{t,n}$ of such a matrix is an integer value representing the number of emitting e_2 electrical ports in use at node v_n at instant ϵ_t. For a given node v_n, let $\upsilon^{L,Em}_n$ be the maximum value of the node's add ports $\psi^{L,Em}_{t,n}$ evaluated over the whole observation period. $\upsilon^{L,Em}_n$ gives the number of emitting e_2 electrical ports installed at the node v_n. The number $\gamma^L_{e_2}$ of emitting e_2 electrical ports in the network is the sum of $\upsilon^{L,Em}_n$ over all the network nodes.

$$\underline{\Psi}^{L,Em} = \underline{\Delta}^{L^T} \times \underline{S}^L \tag{5.10a}$$

$$\upsilon^{L,Em}_n = \max_{1 \leq t \leq \mathbb{E}} \psi^{L,Em}_{t,n} \tag{5.10b}$$

$$\gamma^L_{e_2} = \sum_{n=1}^{N} \upsilon^{L,Em}_n \tag{5.10c}$$

Similarly, by multiplying the transposed request matrix $\underline{\Delta}^{L^T}$ by the destination matrix $\underline{D}^L$, a new matrix $\underline{\Psi}^{L,Re} = \{\psi^{L,Re}_{t,n}\}$ is obtained. $\underline{\Psi}^{L,Re}$, of dimension $[\mathbb{E} \times N]$, represents the use of full-wavelength 'drop' ports at the nodes. An element $\psi^{L,Re}_{t,n}$ of such a matrix is an integer value representing the number of receiving r_2 electrical ports in use at node v_n at instant ϵ_t. For a given node v_n, let $\upsilon^{L,Re}_n$ be the maximum value of the node's drop ports $\psi^{L,Re}_{t,n}$ evaluated over the whole observation period. $\upsilon^{L,Re}_n$ gives the number of receiving r_2 electrical ports installed at the node v_n. The number $\gamma^L_{r_2}$ of receiving r_2 electrical ports in the network is the sum of $\upsilon^{L,Re}_n$ over all the network nodes.

$$\underline{\mathit{\Psi}}^{L,Re} = \underline{\mathit{\Delta}}^{L^T} \times \underline{D}^{L} \tag{5.11a}$$

$$\upsilon_n^{L,Re} = \max_{1\le t\le \mathbb{E}} \psi_{t,n}^{L,Re} \tag{5.11b}$$

$$\gamma_{r_2}^{L} = \sum_{n=1}^{N} \upsilon_n^{L,Re} \tag{5.11c}$$

Consequently, the number $\gamma_{o_2}^{L}$ of o_2 optical ports in the network is equal to:

$$\gamma_{o_2}^{L} = \gamma_{e_2}^{L} + \gamma_{r_2}^{L} \tag{5.12}$$

As a set of SLDs does not have any sub-wavelength component and hence does not need any grooming, the number of the other types of ports is null; namely,

$$\gamma_{e_1}^{L} = \gamma_{r_1}^{L} = \gamma_{e_3}^{L} = \gamma_{r_3}^{L} = \gamma_{o_3}^{L} = 0 \tag{5.13}$$

Finally, the total number ϱ^L of optical ports and the total number ε^L of electrical ports are given by:

$$\begin{aligned} \varrho^L &= \gamma_{o_1}^{L} + \gamma_{o_2}^{L} + \gamma_{o_3}^{L} \\ &= \gamma_{o_1}^{L} + \gamma_{o_2}^{L} \end{aligned} \tag{5.14a}$$

$$\begin{aligned} \varepsilon^L &= \gamma_{e_1}^{L} + \gamma_{e_2}^{L} + \gamma_{e_3}^{L} + \gamma_{r_1}^{L} + \gamma_{r_2}^{L} + \gamma_{r_3}^{L} \\ &= \gamma_{e_2}^{L} + \gamma_{r_2}^{L} \end{aligned} \tag{5.14b}$$

Electrical ports use opto-electronic devices such as laser diodes and photo-detectors. For this reason, electrical ports are more expensive than optical ones. Let κ ($\kappa > 1$) be the multiplicative factor representing the ratio of the cost of an electrical port to the cost of an optical port. The cost ζ^L of routing the SLD requests is expressed as:

$$\zeta^L = \varrho^L + \kappa \times \varepsilon^L \tag{5.15}$$

Example 5.3.

Let us go back and reconsider the set $\mathcal{D} = \{\delta_1^L, \delta_2^L, \delta_3^L\}$ of $M = 3$ SLD traffic requests already introduced in Example 5.1. The characteristics of these requests are given in Table 5.3, and their time diagram is shown in Figure 5.9. Given the characteristics of the 3 SLDs, the ordered set of the arrival dates is $\mathcal{E} = \{\epsilon_1 = 480\ (8{:}00), \epsilon_2 = 660\ (11{:}00), \epsilon_3 = 1020\ (17{:}00)\}$ with $\mathbb{E} = |\mathcal{E}| = 3$.

Table 5.3. Characteristics of the 3 SLDs

δ_i^L	s_i	d_i	α_i	β_i	n_i
δ_1^L	2	8	8:00	14:40	$1 \times C_\omega$
δ_2^L	3	7	11:00	13:00	$1 \times C_\omega$
δ_3^L	2	6	17:00	19:30	$1 \times C_\omega$

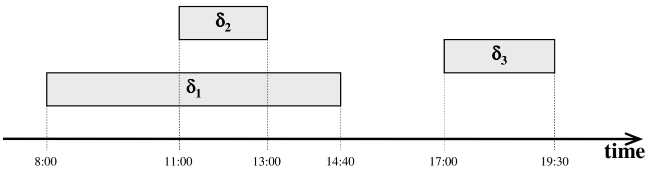


Fig. 5.9. Associated time diagram of the 3 SLDs

The physical topology used is the 9-node and 12-link NSF network (see Figure 5.10). Its equivalent graph is defined by the set of nodes $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\}$ and the set of physical links $E = \{e_{1-2}, e_{2-1}, e_{1-3}, e_{3-1}, e_{1-5}, e_{5-1}, e_{2-3}, e_{3-2}, e_{3-4}, e_{4-3}, e_{4-6}, e_{6-4}, e_{4-7}, e_{7-4}, e_{5-6}, e_{6-5}, e_{6-8}, e_{8-6}, e_{7-8}, e_{8-7}, e_{7-9}, e_{9-7}, e_{8-9}, e_{9-8}\}$. Thus $N = |V| = 9$ and $L = |E| = 24$.

For each SLD δ_i^L, let us compute 2-shortest paths connecting its source node to its destination node. The various sets of available paths are as follows:

- $\mathcal{P}_1 = \{P_{1,1} = [e_{2-3}, e_{3-4}, e_{4-7}, e_{7-8}], P_{1,2} = [e_{2-1}, e_{1-5}, e_{5-6}, e_{6-8}]\}$
- $\mathcal{P}_2 = \{P_{2,1} = [e_{3-4}, e_{4-7}], P_{2,2} = [e_{3-4}, e_{4-6}, e_{6-8}, e_{8-7}]\}$
- $\mathcal{P}_3 = \{P_{3,1} = [e_{2-1}, e_{1-5}, e_{5-6}], P_{3,2} = [e_{2-3}, e_{3-4}, e_{4-6}]\}$

Without loss of generality, the set of physical links can be shrunken to $E = \{e_{2-1}, e_{1-5}, e_{2-3}, e_{3-4}, e_{4-6}, e_{4-7}, e_{5-6}, e_{6-8}, e_{7-8}, e_{8-7}\}$. This is possible since no paths traverse through the other physical links. Hence, the number of links is reduced to $L = |E| = 10$.

An instance of the routing solution is the assignment of the shortest path to each SLD request. This routing solution is represented by $\mathit{\Pi}_{\mathcal{D},\rho} = \big\{P_{1,1} = [e_{2-3}, e_{3-4}, e_{4-7}, e_{7-8}], P_{2,1} = [e_{3-4}, e_{4-7}], P_{3,1} = [e_{2-1}, e_{1-5}, e_{5-6}]\big\}$ and is shown in Figure 5.10.

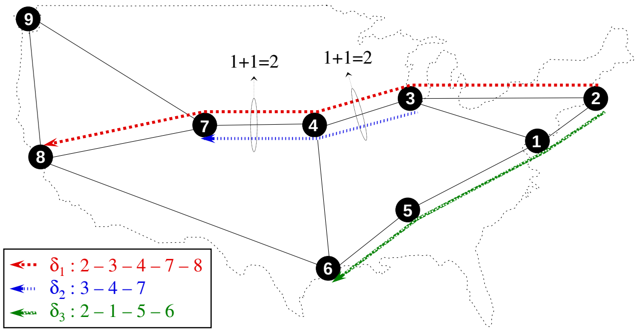


Fig. 5.10. A first routing solution for the set of 3 SLDs

For this instance of the SLD routing solution, the request matrix $\underline{\mathit{\Delta}}^L$, the routing matrix $\underline{R}^L$, the source matrix $\underline{S}^L$, and the destination matrix $\underline{D}^L$ are given by:

$$\underline{\mathit{\Delta}}^L = \begin{array}{c} \\ \delta_1^L \\ \delta_2^L \\ \delta_3^L \end{array} \begin{array}{c} \begin{array}{ccc} \epsilon_1 & \epsilon_2 & \epsilon_3 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

$$\underline{R}^L = \begin{array}{c|cccccccccc} & e_{2-1} & e_{1-5} & e_{2-3} & e_{3-4} & e_{4-6} & e_{4-7} & e_{5-6} & e_{6-8} & e_{7-8} & e_{8-7} \\ \delta_1^L & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ \delta_2^L & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ \delta_3^L & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}$$

$$\underline{S}^L = \begin{array}{c|ccccccccc} & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 \\ \delta_1^L & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \delta_2^L & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \delta_3^L & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

$$\underline{D}^L = \begin{array}{c|ccccccccc} & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 \\ \delta_1^L & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \delta_2^L & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \delta_3^L & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}$$

The matrix $\underline{\Phi}^{\,L}$ representing the traffic load over time carried by the links is given by:

$$\underline{\Phi}^{\,L} = \underline{\Delta}^{\,L^T} \times \underline{R}^{\,L} = \begin{array}{c} \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{array} \begin{array}{c} \begin{array}{cccccccccc} e_{2-1} & e_{1-5} & e_{2-3} & e_{3-4} & e_{4-6} & e_{4-7} & e_{5-6} & e_{6-8} & e_{7-8} & e_{8-7} \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 2 & 0 & 2 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

The number $\gamma_{o_1}^L$ of o_1 optical ports needed in the network is evaluated as:

$$\begin{aligned} \gamma_{o_1}^L &= 2 \times \sum_{l=1}^{L} \max_{1 \leq t \leq \mathbb{E}} \phi_{t,l}^L \\ &= 2 \times (1+1+1+2+0+2+1+0+1+0) = 18 \end{aligned}$$

The matrix $\underline{\Psi}^{\,L,Em}$ representing the use of full-wavelength 'add' ports is given by:

$$\underline{\Psi}^{\,L,Em} = \underline{\Delta}^{\,L^T} \times \underline{S}^{\,L} = \begin{array}{c} \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{array} \begin{array}{c} \begin{array}{ccccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

The number $\gamma_{e_2}^L$ of emitting e_2 electrical ports needed in the network is evaluated as:

$$\begin{aligned} \gamma_{e_2}^L &= \sum_{n=1}^{N} \max_{1 \leq t \leq \mathbb{E}} \psi_{t,n}^{L,Em} \\ &= 0+1+1+0+0+0+0+0+0 = 2 \end{aligned}$$

Similarly, the matrix $\underline{\Psi}^{\,L,Re}$ representing the use of full-wavelength 'drop' ports is given by:

$$\underline{\Psi}^{\,L,Re} = \underline{\Delta}^{\,L^T} \times \underline{D}^{\,L} = \begin{array}{c} \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{array} \begin{array}{c} \begin{array}{ccccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

The number $\gamma_{r_2}^L$ of receiving r_2 electrical ports needed in the network is evaluated as:

$$\begin{aligned} \gamma_{r_2}^L &= \sum_{n=1}^{N} \max_{1 \leq t \leq \mathbb{E}} \psi_{t,n}^{L,Re} \\ &= 0+0+0+0+0+1+1+1+0 = 3 \end{aligned}$$

The number $\gamma_{o_2}^L$ of o_2 optical ports needed in the network is:

$$\gamma_{o_2}^L = \gamma_{e_2}^L + \gamma_{r_2}^L = 2 + 3 = 5$$

Finally, the total number of optical ports and the total number of electrical ports are:

$$\begin{aligned} \varrho^L &= \gamma_{o_1}^L + \gamma_{o_2}^L = 18 + 5 = 23 \\ \varepsilon^L &= \gamma_{e_2}^L + \gamma_{r_2}^L = 2 + 3 = 5 \end{aligned}$$

With κ being equal to 5, the total cost ζ^L of routing the SLD requests is evaluated as:

$$\zeta^L = \varrho^L + \kappa \times \varepsilon^L = 23 + 5 \times 5 = 48$$

A new routing solution can be obtained by changing the path assigned to a given number of requests. For instance, let us change the path assigned to the SLD δ_1^L from its shortest path $P_{1,1}$ to its alternate path $P_{1,2}$. This new solution is represented by $\Pi'_{\mathcal{D},\rho} = \{P_{1,2} = [e_{2-1}, e_{1-5}, e_{5-6}, e_{6-8}], P_{2,1} = [e_{3-4}, e_{4-7}], P_{3,1} = [e_{2-1}, e_{1-5}, e_{5-6}]\}$ and is shown in Figure 5.11.

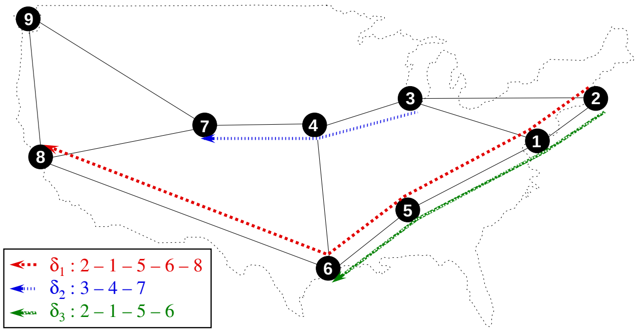


Fig. 5.11. A second routing solution for the set of 3 SLDs

For this new instance of the SLD routing solution, the request matrix $\underline{\Delta}^L$, the source matrix $\underline{S}^L$, and the destination matrix $\underline{D}^L$ remain the same. Only the routing matrix $\underline{R}^L$ changes into:

$$\underline{R}'^L = \begin{array}{c} \\ \delta_1^L \\ \delta_2^L \\ \delta_3^L \end{array} \begin{array}{cccccccccc} e_{2-1} & e_{1-5} & e_{2-3} & e_{3-4} & e_{4-6} & e_{4-7} & e_{5-6} & e_{6-8} & e_{7-8} & e_{8-7} \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}$$

Consequently, the number of emitting e_2 electrical ports, the number of receiving r_2 electrical ports, and the number of o_2 optical ports remain unchanged. Only the number of o_1 optical ports changes and is given by:

$$\underline{\Phi}'^L = \underline{\Delta}^{L^T} \times \underline{R}'^L = \begin{array}{c} \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{array} \begin{array}{cccccccccc} e_{2-1} & e_{1-5} & e_{2-3} & e_{3-4} & e_{4-6} & e_{4-7} & e_{5-6} & e_{6-8} & e_{7-8} & e_{8-7} \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}$$

$$\begin{aligned} \gamma'^L_{o_1} &= 2 \times \sum_{l=1}^{L} \max_{1 \le t \le \mathbb{E}} \phi'^L_{t,l} \\ &= 2 \times (1 + 1 + 0 + 1 + 0 + 1 + 1 + 1 + 0 + 0) = 12 \end{aligned}$$

The total cost ζ'^L of the new routing solution is evaluated as:

$$\begin{aligned} \varrho'^L &= \gamma'^L_{o_1} + \gamma^L_{o_2} = 12 + 5 = 17 \\ \varepsilon^L &= \gamma^L_{e_2} + \gamma^L_{r_2} = 2 + 3 = 5 \\ \zeta'^L &= \varrho'^L + \kappa \times \varepsilon^L = 17 + 5 \times 5 = 42 \end{aligned}$$

To sum up, by changing the path assigned to the SLD δ_1^L from its shortest path to its alternate path, the number of o_1 optical ports required in the network has decreased from 18 to 12. This result confirms the values estimated in the Example 5.1.

5.4.2 Network Cost Computation in the Case of SEDs

Similarly to the case of SLD traffic requests, for an instance $\Pi_{\mathcal{D}^E,\rho,\lambda} = \big\{(P_{1,\rho_1}, L_{1,\rho_1,\lambda_{1,\rho_1}}), (P_{2,\rho_2}, L_{2,\rho_2,\lambda_{2,\rho_2}}), \cdots, (P_{M^E,\rho_{M^E}}, L_{M^E,\rho_{M^E},\lambda_{M^E,\rho_{M^E}}})\big\}$ of the network design problem under SED traffic requests, we define the following matrices:

Request Matrix: The request matrix, denoted by $\underline{\Delta}^E = \{\delta^E_{i,t}\}$, represents the SED traffic requests over time. $\underline{\Delta}^E$ is a $[M^E \times \mathbb{E}]$ matrix. An element $\delta^E_{i,t}$ of such a matrix is a binary value ($\delta^E_{i,t} \in \{0,1\}$) specifying the presence ($\delta^E_{i,t} = 1$) or the absence ($\delta^E_{i,t} = 0$) of the SED request $\delta^E_i(s_i, d_i, \alpha_i, \beta_i, n_i)$ at time instant ϵ_t.

$$\delta^E_{i,t} = \begin{cases} 1 & \text{if } \alpha_i \leqslant \epsilon_t < \beta_i, \\ 0 & \text{otherwise.} \end{cases} \tag{5.16}$$

Source Matrix: The source matrix, denoted by $\underline{S}^E = \{s^E_{i,n}\}$, represents the source nodes of the SED requests. $\underline{S}^E$ is a $[M^E \times N]$ matrix. An element $s^E_{i,n}$ of such a matrix is a binary value ($s^E_{i,n} \in \{0,1\}$) specifying if the SED request $\delta^E_i(s_i, d_i, \alpha_i, \beta_i, n_i)$ has node v_n as a source node ($s^E_{i,n} = 1$) or not ($s^E_{i,n} = 0$).

$$s^E_{i,n} = \begin{cases} 1 & \text{if } s_i = v_n, \\ 0 & \text{otherwise.} \end{cases} \tag{5.17}$$

Destination Matrix: The destination matrix, denoted by $\underline{D}^E = \{d^E_{i,n}\}$, represents the destination nodes of the SED requests. $\underline{D}^E$ is a $[M^E \times N]$ matrix. An element $d^E_{i,n}$ of such a matrix is a binary value ($d^E_{i,n} \in \{0,1\}$) specifying if the SED request $\delta^E_i(s_i, d_i, \alpha_i, \beta_i, n_i)$ has node v_n as a destination node ($d^E_{i,n} = 1$) or not ($d^E_{i,n} = 0$).

$$d^E_{i,n} = \begin{cases} 1 & \text{if } d_i = v_n, \\ 0 & \text{otherwise.} \end{cases} \tag{5.18}$$

As shown in Section 5.2, when grooming two SED requests $\delta^E_{i_1}$ and $\delta^E_{i_2}$, the set $\mathcal{D}^E$ of all the SEDs must be modified by adding the aggregated request $\delta^E_{i_a}$ and the set of marginal demands $\{\delta^E_{i_b}, \delta^E_{i_c}, \delta^E_{i_d}, \delta^E_{i_e}, \delta^E_{i_f}, \delta^E_{i_g}\}$ and removing the original requests $\delta^E_{i_1}$ and $\delta^E_{i_2}$. Consequently, the number of SED requests as well as their characteristics are modified. Let $\widetilde{\mathcal{D}}^E = \{\widetilde{\delta}^E_1, \widetilde{\delta}^E_2, \cdots, \widetilde{\delta}^E_{\widetilde{M}^E}\}$ be the set of the $\widetilde{M}^E$ groomed SED requests obtained at the end of the grooming optimization process. Each groomed SED request $\widetilde{\delta}^E_i \in \widetilde{\mathcal{D}}^E$ is routed over a unique path $\widetilde{P}^E_i$. As each element $\widetilde{\delta}^E_i$ can be either a single low-speed connection ($\widetilde{\delta}^E_i = \delta^E_j$) or the aggregation of two or several low-speed connections ($\widetilde{\delta}^E_i = \uplus_j \delta^E_j$), its corresponding path is determined according to the grooming lightpath sets of its lower-speed connections ($\widetilde{P}^E_i \in L_{j,\rho_j,\lambda_{j,\rho_j}}$).

Based on the new SED set $\widetilde{\mathcal{D}}^E$, we define the following additional matrices:

Groomed Request Matrix: The groomed request matrix, denoted by $\widetilde{\underline{\Delta}}^E = \{\widetilde{\delta}^E_{i,t}\}$, represents the groomed SED traffic requests over time. $\widetilde{\underline{\Delta}}^E$ is a $[\widetilde{M}^E \times \mathbb{E}]$ matrix. An element $\widetilde{\delta}^E_{i,t}$ of such a matrix is an integer number due to the fact that the groomed SED demand $\widetilde{\delta}^E_i(\widetilde{s}_i, \widetilde{d}_i, \widetilde{\alpha}_i, \widetilde{\beta}_i, \widetilde{n}_i)$ can be the concatenation of several requests of smaller rates. Consequently, it could require more than one lightpath to carry its traffic load. $\widetilde{\delta}^E_{i,t}$ is equal to the smallest integer greater than or equal to the traffic required by the groomed SED request $\widetilde{\delta}^E_i$ and specifies the number of lightpaths used by this request at time instant ϵ_t.

$$\widetilde{\delta}^{E}_{i,t} = \begin{cases} \lceil \widetilde{n}_i \rceil & \text{if } \widetilde{\alpha}_i \leqslant \epsilon_t < \widetilde{\beta}_i, \\ 0 & \text{otherwise.} \end{cases} \tag{5.19}$$

Groomed Routing Matrix: The groomed routing matrix, denoted by $\underline{\widetilde{R}}^{\,E} = \{\widetilde{r}^{E}_{i,l}\}$, represents the use of the physical links by the groomed SED requests. $\underline{\widetilde{R}}^{\,E}$ is a $[\widetilde{M}^{E} \times L]$ matrix. An element $\widetilde{r}^{E}_{i,l}$ of such a matrix is a binary value ($\widetilde{r}^{E}_{i,l} \in \{0,1\}$) specifying if the physical path $\widetilde{P}^{E}_{i}$ assigned to the groomed SED request $\widetilde{\delta}^{E}_{i}(\widetilde{s}_i, \widetilde{d}_i, \widetilde{\alpha}_i, \widetilde{\beta}_i, \widetilde{n}_i)$ passes through the physical link e_l ($\widetilde{r}^{E}_{i,l} = 1$) or not ($\widetilde{r}^{E}_{i,l} = 0$).

$$\widetilde{r}^{E}_{i,l} = \begin{cases} 1 & \text{if } e_l \in \widetilde{P}^{E}_{i}, \\ 0 & \text{otherwise.} \end{cases} \tag{5.20}$$

Groomed Source Matrix: The groomed source matrix, denoted by $\underline{\widetilde{S}}^{\,E} = \{\widetilde{s}^{E}_{i,n}\}$, represents the source nodes of the groomed SED requests. $\underline{\widetilde{S}}^{\,E}$ is a $[\widetilde{M}^{E} \times N]$ matrix. An element $\widetilde{s}^{E}_{i,n}$ of such a matrix is a binary value ($\widetilde{s}^{E}_{i,n} \in \{0,1\}$) specifying if the groomed SED request $\widetilde{\delta}^{E}_{i}(\widetilde{s}_i, \widetilde{d}_i, \widetilde{\alpha}_i, \widetilde{\beta}_i, \widetilde{n}_i)$ has node v_n as a source node ($\widetilde{s}^{E}_{i,n} = 1$) or not ($\widetilde{s}^{E}_{i,n} = 0$).

$$\widetilde{s}^{E}_{i,n} = \begin{cases} 1 & \text{if } \widetilde{s}_i = v_n, \\ 0 & \text{otherwise.} \end{cases} \tag{5.21}$$

Groomed Destination Matrix: The groomed destination matrix, denoted by $\underline{\widetilde{D}}^{\,E} = \{\widetilde{d}^{E}_{i,n}\}$, represents the destination nodes of the groomed SED requests. $\underline{\widetilde{D}}^{\,E}$ is a $[\widetilde{M}^{E} \times N]$ matrix. An element $\widetilde{d}^{E}_{i,n}$ of such a matrix is a binary value ($\widetilde{d}^{E}_{i,n} \in \{0,1\}$) specifying if the groomed SED request $\widetilde{\delta}^{E}_{i}(\widetilde{s}_i, \widetilde{d}_i, \widetilde{\alpha}_i, \widetilde{\beta}_i, \widetilde{n}_i)$ has node v_n as a destination node ($\widetilde{d}^{E}_{i,n} = 1$) or not ($\widetilde{d}^{E}_{i,n} = 0$).

$$\widetilde{d}^{E}_{i,n} = \begin{cases} 1 & \text{if } \widetilde{d}_i = v_n \ , \\ 0 & \text{otherwise.} \end{cases} \tag{5.22}$$

By multiplying the transposed request matrix $\underline{\varDelta}^{\,E^{T}}$ by the source matrix $\underline{S}^{\,E}$, a new matrix $\underline{\varPsi}^{\,E,Em} = \{\psi^{E,Em}_{t,n}\}$ is obtained. $\underline{\varPsi}^{\,E,Em}$, of dimension $[\mathbb{E} \times N]$, represents the use of sub-wavelength 'add' ports at the nodes. An element $\psi^{E,Em}_{t,n}$ of such a matrix is an integer value representing the number of emitting e_1 electrical ports in use at node v_n at instant ϵ_t. For a given node v_n, let $\upsilon^{E,Em}_{n}$ be the maximum value of the node's add ports $\psi^{E,Em}_{t,n}$ evaluated over the whole observation period. $\upsilon^{E,Em}_{n}$ gives the number of emitting e_1 electrical ports installed at the node v_n. The number $\gamma^{E}_{e_1}$ of emitting e_1 electrical ports in the network is the sum of $\upsilon^{E,Em}_{n}$ over all the network nodes.

$$\underline{\varPsi}^{\,E,Em} = \underline{\varDelta}^{\,E^{T}} \times \underline{S}^{\,E} \tag{5.23a}$$

$$\upsilon^{E,Em}_{n} = \max_{1 \leq t \leq \mathbb{E}} \psi^{E,Em}_{t,n} \tag{5.23b}$$

$$\gamma^{E}_{e_1} = \sum_{n=1}^{N} \upsilon^{E,Em}_{n} \tag{5.23c}$$

By multiplying the transposed request matrix $\underline{\varDelta}^{\,E^{T}}$ by the destination matrix $\underline{D}^{\,E}$, a new matrix $\underline{\varPsi}^{\,E,Re} = \{\psi^{E,Re}_{t,n}\}$ is obtained. $\underline{\varPsi}^{\,E,Re}$, of dimension $[\mathbb{E} \times N]$, represents the use of sub-wavelength 'drop' ports at the nodes. An element $\psi^{E,Re}_{t,n}$ of such a matrix is an integer value representing the number of receiving r_1 electrical ports in use at node v_n at instant ϵ_t. For a given node v_n, let $\upsilon^{E,Re}_{n}$ be the maximum value of the node's drop ports $\psi^{E,Re}_{t,n}$ evaluated over the whole observation period.

$\upsilon_n^{E,Re}$ gives the number of receiving r_1 electrical ports installed at the node v_n. The number $\gamma_{r_1}^E$ of receiving r_1 electrical ports in the network is the sum of $\upsilon_n^{E,Re}$ over all the network nodes.

$$\underline{\Psi}^{E,Re} = \underline{\Delta}^{E^T} \times \underline{D}^{E} \tag{5.24a}$$

$$\upsilon_n^{E,Re} = \max_{1 \leq t \leq \mathbb{E}} \psi_{t,n}^{E,Re} \tag{5.24b}$$

$$\gamma_{r_1}^E = \sum_{n=1}^{N} \upsilon_n^{E,Re} \tag{5.24c}$$

By multiplying the transposed groomed request matrix $\underline{\widetilde{\Delta}}^{E^T}$ by the groomed routing matrix $\underline{\widetilde{R}}^{E}$, a new matrix $\underline{\widetilde{\Phi}}^{E} = \{\widetilde{\phi}_{t,l}^{E}\}$ is obtained. $\underline{\widetilde{\Phi}}^{E}$, of dimension $[\mathbb{E} \times L]$, represents the traffic load over time carried by the links. An element $\widetilde{\phi}_{t,l}^{E}$ of such a matrix is an integer value representing the number of WDM optical channels carried by link e_l at instant ϵ_t. For a given link e_l, let $\widetilde{\varphi}_l^E$ be the maximum value of this traffic $\widetilde{\phi}_{t,l}^{E}$ evaluated over the whole observation period. $\widetilde{\varphi}_l^E$ gives the number of WDM optical channels used on this link. As the o_1 optical ports go by pair, one port at each end of a WDM channel, $\widetilde{\varphi}_l^E$ represents also the number of o_1 optical port pairs installed at both ends of the link e_l. The number $\gamma_{o_1}^E$ of o_1 optical ports in the network is twice the sum of $\widetilde{\varphi}_l^E$ over all the network links. The maximum value of $\widetilde{\varphi}_l^E$ $(\forall l \backslash 1 \leqslant l \leqslant L)$ represents the congestion in the network, *i.e.,* the number of active channels on the most loaded link.

$$\underline{\widetilde{\Phi}}^{E} = \underline{\widetilde{\Delta}}^{E^T} \times \underline{\widetilde{R}}^{E} \tag{5.25a}$$

$$\widetilde{\varphi}_l^E = \max_{1 \leq t \leq \mathbb{E}} \widetilde{\phi}_{t,l}^{E} \tag{5.25b}$$

$$\gamma_{o_1}^E = 2 \times \sum_{l=1}^{L} \widetilde{\varphi}_l^E \tag{5.25c}$$

By multiplying the transposed groomed request matrix $\underline{\widetilde{\Delta}}^{E^T}$ by the groomed source matrix $\underline{\widetilde{S}}^{E}$, a new matrix $\underline{\widetilde{\Psi}}^{E,Em} = \{\widetilde{\psi}_{t,n}^{E,Em}\}$ is obtained. $\underline{\widetilde{\Psi}}^{E,Em}$, of dimension $[\mathbb{E} \times N]$, represents the use of switching ports from the EXC to the OXC at the nodes. An element $\widetilde{\psi}_{t,n}^{E,Em}$ of such a matrix is an integer value representing the number of emitting e_3 electrical ports in use at node v_n at instant ϵ_t. For a given node v_n, let $\widetilde{\upsilon}_n^{E,Em}$ be the maximum value of the node's switching ports $\widetilde{\psi}_{t,n}^{E,Em}$ evaluated over the whole observation period. $\widetilde{\upsilon}_n^{E,Em}$ gives the number of emitting e_3 electrical ports installed at the node v_n. The number $\gamma_{e_3}^E$ of emitting e_3 electrical ports in the network is the sum of $\widetilde{\upsilon}_n^{E,Em}$ over all the network nodes.

$$\underline{\widetilde{\Psi}}^{E,Em} = \underline{\widetilde{\Delta}}^{E^T} \times \underline{\widetilde{S}}^{E} \tag{5.26a}$$

$$\widetilde{\upsilon}_n^{E,Em} = \max_{1 \leq t \leq \mathbb{E}} \widetilde{\psi}_{t,n}^{E,Em} \tag{5.26b}$$

$$\gamma_{e_3}^E = \sum_{n=1}^{N} \widetilde{\upsilon}_n^{E,Em} \tag{5.26c}$$

Similarly, by multiplying the transposed groomed request matrix $\underline{\widetilde{\Delta}}^{E^T}$ by the groomed destination matrix $\underline{\widetilde{D}}^{E}$, a new matrix $\underline{\widetilde{\Psi}}^{E,Re} = \{\widetilde{\psi}_{t,n}^{E,Re}\}$ is obtained. $\underline{\widetilde{\Psi}}^{E,Re}$, of dimension $[\mathbb{E} \times N]$, represents the use of switching ports from the OXC to the EXC at the nodes. An element $\widetilde{\psi}_{t,n}^{E,Re}$ of such a matrix is an integer value representing the number of receiving r_3 electrical ports in use at node v_n at instant ϵ_t. For a given node v_n, let $\widetilde{\upsilon}_n^{E,Re}$ be the maximum value of the node's switching ports $\widetilde{\psi}_{t,n}^{E,Re}$ evaluated over the whole observation period. $\widetilde{\upsilon}_n^{E,Re}$ gives the number of receiving r_3 electrical ports installed at the node v_n. The number $\gamma_{r_3}^{E}$ of receiving r_3 electrical ports in the network is the sum of $\widetilde{\upsilon}_n^{E,Re}$ over all the network nodes.

$$\underline{\widetilde{\Psi}}^{E,Re} = \underline{\widetilde{\Delta}}^{E^T} \times \underline{\widetilde{D}}^{E} \tag{5.27a}$$

$$\widetilde{\upsilon}_n^{E,Re} = \max_{1 \le t \le \mathbb{E}} \widetilde{\psi}_{t,n}^{E,Re} \tag{5.27b}$$

$$\gamma_{r_3}^{E} = \sum_{n=1}^{N} \widetilde{\upsilon}_n^{E,Re} \tag{5.27c}$$

Consequently, the number $\gamma_{o_3}^{E}$ of o_3 optical ports in the network is equal to:

$$\gamma_{o_3}^{E} = \gamma_{e_3}^{E} + \gamma_{r_3}^{E} \tag{5.28}$$

As a set of SEDs does not have any full-wavelength component to be added or dropped at the nodes, the number of e_2/r_2 electrical ports and the number of o_2 optical ports are null.

$$\gamma_{e_2}^{E} = \gamma_{r_2}^{E} = \gamma_{o_2}^{E} = 0 \tag{5.29}$$

Finally, the total number ϱ^{E} of optical ports and the total number ε^{E} of electrical ports are given by:

$$\begin{aligned} \varrho^{E} &= \gamma_{o_1}^{E} + \gamma_{o_2}^{E} + \gamma_{o_3}^{E} \\ &= \gamma_{o_1}^{E} + \gamma_{o_3}^{E} && (5.30a) \\ \varepsilon^{E} &= \gamma_{e_1}^{E} + \gamma_{e_2}^{E} + \gamma_{e_3}^{E} + \gamma_{r_1}^{E} + \gamma_{r_2}^{E} + \gamma_{r_3}^{E} \\ &= \gamma_{e_1}^{E} + \gamma_{e_3}^{E} + \gamma_{r_1}^{E} + \gamma_{r_3}^{E} && (5.30b) \end{aligned}$$

With κ being the multiplicative factor representing the ratio of the cost of an electrical port to the cost of an optical port, the cost ζ^{E} of routing the SED requests is expressed as:

$$\zeta^{E} = \varrho^{E} + \kappa \times \varepsilon^{E} \tag{5.31}$$

Example 5.4.

Let us go back and reconsider the set $\mathcal{D} = \{\delta_1^E, \delta_2^E, \delta_3^E\}$ of $M = 3$ SED traffic requests already introduced in Example 5.2. The characteristics of these request are given by Table 5.4, and their time diagram is shown in Figure 5.12. Given the characteristics of the 3 SEDs, the ordered set of the set-up/tear-down dates is $\mathcal{E} = \{\epsilon_1 = 480\ (8{:}00), \epsilon_2 = 660\ (11{:}00), \epsilon_3 = 780\ (13{:}00), \epsilon_4 = 880\ (14{:}40), \epsilon_5 = 1020\ (17{:}00), \epsilon_6 = 1170\ (19{:}30)\}$ with $\mathbb{E} = |\mathcal{E}| = 6$.

Table 5.4. Characteristics of the 3 SEDs

δ_i^E	s_i	d_i	α_i	β_i	n_i
δ_1^E	2	8	8:00	14:40	$0.25 \times C_\omega$
δ_2^E	3	7	11:00	13:00	$0.25 \times C_\omega$
δ_3^E	2	6	17:00	19:30	$0.25 \times C_\omega$

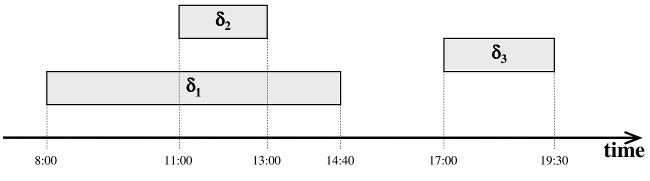


Fig. 5.12. Associated time diagram of the 3 SEDs

The physical topology used is the 9-node and 12-link NSF network. Its equivalent graph is defined by the set of nodes $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\}$ and the set of physical links

$E = \{e_{1-2}, e_{2-1}, e_{1-3}, e_{3-1}, e_{1-5}, e_{5-1}, e_{2-3}, e_{3-2}, e_{3-4}, e_{4-3}, e_{4-6}, e_{6-4}, e_{4-7}, e_{7-4}, e_{5-6}, e_{6-5}, e_{6-8}, e_{8-6}, e_{7-8}, e_{8-7}, e_{7-9}, e_{9-7}, e_{8-9}, e_{9-8}\}$. Thus $N = |V| = 9$ and $L = |E| = 24$.

As in Example 5.2, and in order to highlight the impact of the grooming operation on the network cost, each SED δ_i^E is assumed to be routed along the shortest path between its source node and its destination node. The shortest paths of the various SED requests are:

- $\mathcal{P}_1 = \{P_{1,1} = [e_{2-3}, e_{3-4}, e_{4-7}, e_{7-8}]\}$
- $\mathcal{P}_2 = \{P_{2,1} = [e_{3-4}, e_{4-7}]\}$
- $\mathcal{P}_3 = \{P_{3,1} = [e_{2-1}, e_{1-5}, e_{5-6}]\}$

By varying the choice of the intermediate node(s) where the SED request is switched from the OXC to the EXC, the physical path is decomposed into sub-paths. Each sub-path corresponds to a grooming lightpath in the logical topology. As a result, a given physical path can be the concatenation of one or several GLs in the logical topology. The set $\mathcal{L}_{i,\rho_i}$ of all possible decompositions of the path P_{i,ρ_i} into its inherent sets of grooming lightpaths is:

- $P_{1,1} = [e_{2-3}, e_{3-4}, e_{4-7}, e_{7-8}] \quad \longmapsto \quad \mathcal{L}_{1,1} = \Big\{L_{1,1,1} = \big\{[e_{2-3}, e_{3-4}, e_{4-7}, e_{7-8}]\big\},$
  $L_{1,1,2} = \big\{[e_{2-3}], [e_{3-4}, e_{4-7}, e_{7-8}]\big\},\ L_{1,1,3} = \big\{[e_{2-3}, e_{3-4}], [e_{4-7}, e_{7-8}]\big\},$
  $L_{1,1,4} = \big\{[e_{2-3}, e_{3-4}, e_{4-7}], [e_{7-8}]\big\},\ L_{1,1,5} = \big\{[e_{2-3}], [e_{3-4}], [e_{4-7}, e_{7-8}]\big\},$
  $L_{1,1,6} = \big\{[e_{2-3}], [e_{3-4}, e_{4-7}], [e_{7-8}]\big\},\ L_{1,1,7} = \big\{[e_{2-3}, e_{3-4}], [e_{4-7}], [e_{7-8}]\big\},$
  $L_{1,1,8} = \big\{[e_{2-3}], [e_{3-4}], [e_{4-7}], [e_{7-8}]\big\}\Big\}$
- $P_{2,1} = [e_{3-4}, e_{4-7}] \quad \longmapsto \quad \mathcal{L}_{2,1} = \Big\{L_{2,1,1} = \big\{[e_{3-4}, e_{4-7}]\big\},\ L_{2,1,2} = \big\{[e_{3-4}], [e_{4-7}]\big\}\Big\}$
- $P_{3,1} = [e_{2-1}, e_{1-5}, e_{5-6}] \quad \longmapsto \quad \mathcal{L}_{3,1} = \Big\{L_{3,1,1} = \big\{[e_{2-1}, e_{1-5}, e_{5-6}]\big\},$
  $L_{3,1,2} = \big\{[e_{2-1}], [e_{1-5}, e_{5-6}]\big\},\ L_{3,1,3} = \big\{[e_{2-1}, e_{1-5}], [e_{5-6}]\big\},$
  $L_{3,1,4} = \big\{[e_{2-1}], [e_{1-5}], [e_{5-6}]\big\}\Big\}$

Without loss of generality, the set of physical links can be shrunken to $E = \{e_{2-1}, e_{1-5}, e_{2-3}, e_{3-4}, e_{4-7}, e_{5-6}, e_{7-8}\}$. This is possible since no paths traverse through the other physical links. Hence, the number of links is reduced to $L = |E| = 7$.

For this set of SED requests, the request matrix $\underline{\Delta}^E$, the source matrix $\underline{S}^E$, and the destination matrix $\underline{D}^E$ are given by:

$$\underline{\Delta}^E = \begin{array}{c} \\ \delta_1^E \\ \delta_2^E \\ \delta_3^E \end{array} \begin{array}{c} \begin{array}{cccccc} \epsilon_1 & \epsilon_2 & \epsilon_3 & \epsilon_4 & \epsilon_5 & \epsilon_6 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

$$\underline{S}^E = \begin{array}{c} \\ \delta_1^E \\ \delta_2^E \\ \delta_3^E \end{array} \begin{array}{c} \begin{array}{ccccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$\underline{D}^E = \begin{array}{c} \\ \delta_1^E \\ \delta_2^E \\ \delta_3^E \end{array} \begin{array}{c} \begin{array}{ccccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

A first instance of the routing and grooming solution is the assignment of the shortest path to each SED request without any grooming process. This routing solution is represented by $\varPi_{\mathcal{D},\rho,\lambda} = \big\{(P_{1,1}, L_{1,1,1}), (P_{2,1}, L_{2,1,1}), (P_{3,1}, L_{3,1,1})\big\}$ and is shown in Figure 5.13.

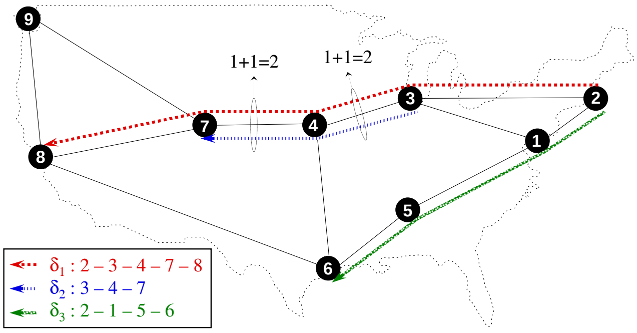


Fig. 5.13. A first routing and grooming solution for the set of 3 SEDs

As no grooming is performed, the set $\widetilde{\mathcal{D}}^E$ of the groomed requests is the same as the set $\mathcal{D}^E$ of the original requests. The characteristics of the $\widetilde{M}^E = M^E = 3$ groomed SED requests are given in Table 5.5.

Table 5.5. Characteristics of the groomed SEDs

$\widetilde{\delta}_i^E$	$\widetilde{s}_i$	$\widetilde{d}_i$	$\widetilde{\alpha}_i$	$\widetilde{\beta}_i$	$\widetilde{n}_i$	$\widetilde{P}_i^E$
$\widetilde{\delta}_1^E$	2	8	8:00	14:40	$0.25 \times C_\omega$	$e_{2-3}, e_{3-4}, e_{4-7}, e_{7-8}$
$\widetilde{\delta}_2^E$	3	7	11:00	13:00	$0.25 \times C_\omega$	e_{3-4}, e_{4-7}
$\widetilde{\delta}_3^E$	2	6	17:00	19:30	$0.25 \times C_\omega$	$e_{2-1}, e_{1-5}, e_{5-6}$

From the previous statement, one can derive the following:

- the groomed request matrix $\underline{\widetilde{\Delta}}^E$ is equal to the request matrix $\underline{\Delta}^E$
- the groomed source matrix $\underline{\widetilde{S}}^E$ is equal to the source matrix $\underline{S}^E$
- the groomed destination matrix $\underline{\widetilde{D}}^E$ is equal to the destination matrix $\underline{D}^E$

Finally, the groomed routing matrix $\underline{\widetilde{R}}^E$ associated with this set $\widetilde{\mathcal{D}}^E$ of groomed SED requests is given by:

$$\underline{\widetilde{R}}^E = \begin{array}{c} \\ \widetilde{\delta}_1^E \\ \widetilde{\delta}_2^E \\ \widetilde{\delta}_3^E \end{array} \begin{array}{c} \begin{array}{ccccccc} e_{2-1} & e_{1-5} & e_{2-3} & e_{3-4} & e_{4-7} & e_{5-6} & e_{7-8} \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

The matrix $\underline{\Psi}^{E,Em}$ representing the use of sub-wavelength 'add' ports is given by:

$$\underline{\Psi}^{E,Em} = \underline{\Delta}^{E^T} \times \underline{S}^E = \begin{array}{c} \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{array} \begin{array}{c} \begin{array}{ccccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

The number $\gamma_{e_1}^E$ of emitting e_1 electrical ports needed in the network is evaluated as:

$$\begin{aligned} \gamma_{e_1}^E &= \sum_{n=1}^{N} \max_{1 \leq t \leq \mathbb{E}} \psi_{t,n}^{E,Em} \\ &= 0 + 1 + 1 + 0 + 0 + 0 + 0 + 0 + 0 = 2 \end{aligned}$$

Similarly, the matrix $\underline{\varPsi}^{E,Re}$ representing the use of sub-wavelength 'drop' ports is given by:

$$\underline{\varPsi}^{E,Re} = \underline{\varDelta}^{E^T} \times \underline{D}^{E} = \begin{array}{c} \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{array} \begin{array}{c} \begin{array}{ccccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

The number $\gamma^{E}_{r_1}$ of receiving r_1 electrical ports needed in the network is evaluated as:

$$\begin{aligned} \gamma^{E}_{r_1} &= \sum_{n=1}^{N} \max_{1 \le t \le \mathbb{E}} \psi^{E,Re}_{t,n} \\ &= 0+0+0+0+0+1+1+1+0 = 3 \end{aligned}$$

The matrix $\widetilde{\underline{\varPhi}}^{E}$ representing the traffic load carried by the links is given by:

$$\widetilde{\underline{\varPhi}}^{E} = \widetilde{\underline{\varDelta}}^{E^T} \times \widetilde{\underline{R}}^{E} = \begin{array}{c} \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{array} \begin{array}{c} \begin{array}{ccccccc} e_{2-1} & e_{1-5} & e_{2-3} & e_{3-4} & e_{4-7} & e_{5-6} & e_{7-8} \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 2 & 2 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

The number $\gamma^{E}_{o_1}$ of o_1 optical ports needed in the network is evaluated as:

$$\begin{aligned} \gamma^{E}_{o_1} &= 2 \times \sum_{l=1}^{L} \max_{1 \le t \le \mathbb{E}} \widetilde{\phi}^{E}_{t,l} \\ &= 2 \times (1+1+1+2+2+1+1) = 18 \end{aligned}$$

The matrix $\widetilde{\underline{\varPsi}}^{E,Em}$ representing the use of switching ports from the EXC to the OXC is given by:

$$\begin{aligned} \widetilde{\underline{\varPsi}}^{E,Em} &= \widetilde{\underline{\varDelta}}^{E^T} \times \widetilde{\underline{S}}^{E} \\ &= \underline{\varDelta}^{E^T} \times \underline{S}^{E} \\ &= \underline{\varPsi}^{E,Em} \end{aligned}$$

Similarly, the matrix $\widetilde{\underline{\varPsi}}^{E,Re}$ representing the use of switching ports from the OXC to the EXC is given by:

$$\begin{aligned} \widetilde{\underline{\varPsi}}^{E,Re} &= \widetilde{\underline{\varDelta}}^{E^T} \times \widetilde{\underline{D}}^{E} \\ &= \underline{\varDelta}^{E^T} \times \underline{D}^{E} \\ &= \underline{\varPsi}^{E,Re} \end{aligned}$$

Consequently, the number $\gamma^{E}_{e_3}$ of emitting e_3 electrical ports, the number $\gamma^{E}_{r_3}$ of receiving r_3 electrical ports, and the number $\gamma^{E}_{o_3}$ of o_3 optical ports are evaluated as:

$$
\begin{aligned}
\gamma^E_{e_3} &= \gamma^E_{e_1} = 2\\
\gamma^E_{r_3} &= \gamma^E_{r_1} = 3\\
\gamma^E_{o_3} &= \gamma^E_{e_3} + \gamma^E_{r_3} = 2 + 3 = 5
\end{aligned}
$$

Finally, the total number of optical ports and the total number of electrical ports are:

$$
\begin{aligned}
\varrho^E &= \gamma^E_{o_1} + \gamma^E_{o_3} = 18 + 5 = 23\\
\varepsilon^E &= \gamma^E_{e_1} + \gamma^E_{e_3} + \gamma^E_{r_1} + \gamma^E_{r_3} = 2 + 2 + 3 + 3 = 10
\end{aligned}
$$

With κ being equal to 5, the total cost ζ^E of routing and grooming the SED requests is evaluated as:

$$\zeta^E = \varrho^E + \kappa \times \varepsilon^E = 23 + 5 \times 10 = 73$$

A new instance of the routing and grooming solution is the assignment of the shortest path to each SED request with the requests δ^E_1 and δ^E_2 being aggregated together between the nodes v_3 and v_7. This routing solution is represented by $\mathit{\Pi}'_{\mathcal{D},\rho,\lambda} = \big\{(P_{1,1}, L_{1,1,6}), (P_{2,1}, L_{2,1,1}), (P_{3,1}, L_{3,1,1})\big\}$ and is shown in Figure 5.14.

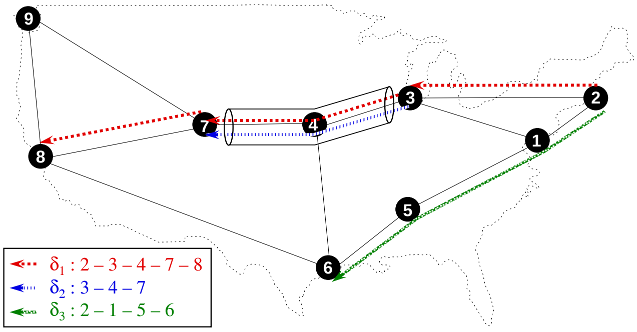


Fig. 5.14. A second routing and grooming solution for the set of 3 SEDs

As shown in Section 5.2, when grooming the two SED requests δ^E_1 and δ^E_2, the set $\mathcal{D}^E$ of all the SEDs must be modified by adding the aggregated request and the set of marginal demands and removing the original requests δ^E_1 and δ^E_2. Hence, the new set $\widetilde{\mathcal{D}}'^E$ of groomed SED requests is defined by $\widetilde{\mathcal{D}}'^E = \big\{\widetilde{\delta}'^E_i, i = 1 \cdots \widetilde{M}'^E = 1 \cdots 6\big\}$ where:

- $\widetilde{\delta}'^E_1$ is the aggregated request resulting from the aggregation of δ^E_1 and δ^E_2 ($\widetilde{\delta}'^E_1 = \delta^E_1 \uplus \delta^E_2$)
- $\widetilde{\delta}'^E_2$ and $\widetilde{\delta}'^E_3$ take into account the non common period of time between δ^E_1 and δ^E_2
- $\widetilde{\delta}'^E_4$ and $\widetilde{\delta}'^E_5$ take into account the non common segments of the paths of δ^E_1 and δ^E_2
- $\widetilde{\delta}'^E_6$ is the original SED request δ^E_3

The characteristics of the new set $\widetilde{\mathcal{D}}'^E$ of groomed SED requests are given in Table 5.6.

For this new instance of the SED routing and grooming solution, the request matrix $\underline{\mathit{\Delta}}^E$, the source matrix $\underline{S}^E$, and the destination matrix $\underline{D}^E$ remain the same. Consequently, the number of emitting e_1 electrical ports and the number of receiving r_1 electrical ports remain unchanged.

$$
\begin{aligned}
\gamma'^E_{e_1} &= \gamma^E_{e_1} = 2\\
\gamma'^E_{r_1} &= \gamma^E_{r_1} = 3
\end{aligned}
$$

Table 5.6. Characteristics of the new groomed SEDs

$\widetilde{\delta}'^E_i$	$\widetilde{s}_i$	$\widetilde{d}_i$	$\widetilde{\alpha}_i$	$\widetilde{\beta}_i$	$\widetilde{n}_i$	$\widetilde{P}^E_i$
$\widetilde{\delta}'^E_1$	3	7	11:00	13:00	$0.5 \times C_\omega$	e_{3-4}, e_{4-7}
$\widetilde{\delta}'^E_2$	3	7	8:00	11:00	$0.25 \times C_\omega$	e_{3-4}, e_{4-7}
$\widetilde{\delta}'^E_3$	3	7	13:00	14:40	$0.25 \times C_\omega$	e_{3-4}, e_{4-7}
$\widetilde{\delta}'^E_4$	2	3	8:00	14:40	$0.25 \times C_\omega$	e_{2-3}
$\widetilde{\delta}'^E_5$	7	8	8:00	14:40	$0.25 \times C_\omega$	e_{7-8}
$\widetilde{\delta}'^E_6$	2	6	17:00	19:30	$0.25 \times C_\omega$	$e_{2-1}, e_{1-5}, e_{5-6}$

The groomed request matrix $\underline{\widetilde{\Delta}}'^E$, the groomed routing matrix $\underline{\widetilde{R}}'^E$, the groomed source matrix $\underline{\widetilde{S}}'^E$, and the groomed destination matrix $\underline{\widetilde{D}}'^E$ have changed and are given by:

$$\underline{\widetilde{\Delta}}'^E = \begin{array}{c|cccccc|} & \epsilon_1 & \epsilon_2 & \epsilon_3 & \epsilon_4 & \epsilon_5 & \epsilon_6 \\ \widetilde{\delta}'^E_1 & 0 & 1 & 0 & 0 & 0 & 0 \\ \widetilde{\delta}'^E_2 & 1 & 0 & 0 & 0 & 0 & 0 \\ \widetilde{\delta}'^E_3 & 0 & 0 & 1 & 0 & 0 & 0 \\ \widetilde{\delta}'^E_4 & 1 & 1 & 1 & 0 & 0 & 0 \\ \widetilde{\delta}'^E_5 & 1 & 1 & 1 & 0 & 0 & 0 \\ \widetilde{\delta}'^E_6 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}$$

$$\underline{\widetilde{R}}'^E = \begin{array}{c|ccccccc|} & e_{2-1} & e_{1-5} & e_{2-3} & e_{3-4} & e_{4-7} & e_{5-6} & e_{7-8} \\ \widetilde{\delta}'^E_1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ \widetilde{\delta}'^E_2 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ \widetilde{\delta}'^E_3 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ \widetilde{\delta}'^E_4 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \widetilde{\delta}'^E_5 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \widetilde{\delta}'^E_6 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{array}$$

$$\underline{\widetilde{S}}'^E = \begin{array}{c|ccccccccc|} & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 \\ \widetilde{\delta}'^E_1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \widetilde{\delta}'^E_2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \widetilde{\delta}'^E_3 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \widetilde{\delta}'^E_4 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \widetilde{\delta}'^E_5 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \widetilde{\delta}'^E_6 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

$$\underline{\widetilde{D}}'^E = \begin{array}{c|ccccccccc|} & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 \\ \widetilde{\delta}'^E_1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \widetilde{\delta}'^E_2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \widetilde{\delta}'^E_3 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \widetilde{\delta}'^E_4 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \widetilde{\delta}'^E_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \widetilde{\delta}'^E_6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}$$

The new matrix $\underline{\widetilde{\Phi}}'^E$ representing the traffic load carried by the links is given by:

$$\underline{\widetilde{\Phi}}'^{E} = \underline{\widetilde{\Delta}}'^{E^T} \times \underline{\widetilde{R}}'^{E} = \begin{array}{c} \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{array} \begin{array}{c} \begin{array}{ccccccc} e_{2-1} & e_{1-5} & e_{2-3} & e_{3-4} & e_{4-7} & e_{5-6} & e_{7-8} \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

The number $\gamma'^{E}_{o_1}$ of o_1 optical ports needed in the network is evaluated as:

$$\begin{aligned} \gamma'^{E}_{o_1} &= 2 \times \sum_{l=1}^{L} \max_{1 \leq t \leq \mathbb{E}} \widetilde{\phi}'^{E}_{t,l} \\ &= 2 \times (1+1+1+1+1+1+1) = 14 \end{aligned}$$

The new matrix $\underline{\widetilde{\Psi}}'^{E,Em}$ representing the use of switching ports from the EXC to the OXC is given by:

$$\underline{\widetilde{\Psi}}'^{E,Em} = \underline{\widetilde{\Delta}}'^{E^T} \times \underline{\widetilde{S}}'^{E} = \begin{array}{c} \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{array} \begin{array}{c} \begin{array}{ccccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

The number $\gamma'^{E}_{e_3}$ of emitting e_3 electrical ports needed in the network is evaluated as:

$$\begin{aligned} \gamma'^{E}_{e_3} &= \sum_{n=1}^{N} \max_{1 \leq t \leq \mathbb{E}} \psi'^{E,Em}_{t,n} \\ &= 0+1+1+0+0+0+1+0+0 = 3 \end{aligned}$$

Similarly, the new matrix $\underline{\widetilde{\Psi}}'^{E,Re}$ representing the use of switching ports from the OXC to the EXC is given by:

$$\underline{\widetilde{\Psi}}'^{E,Re} = \underline{\widetilde{\Delta}}'^{E^T} \times \underline{\widetilde{D}}'^{E} = \begin{array}{c} \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{array} \begin{array}{c} \begin{array}{ccccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

The number $\gamma'^{E}_{r_3}$ of receiving r_3 electrical ports needed in the network is evaluated as:

$$\begin{aligned} \gamma'^{E}_{r_3} &= \sum_{n=1}^{N} \max_{1 \leq t \leq \mathbb{E}} \psi'^{E,Re}_{t,n} \\ &= 0+0+1+0+0+1+1+1+0 = 4 \end{aligned}$$

Consequently, the number $\gamma'^{E}_{o_3}$ of o_3 optical ports is evaluated as:

$$\gamma'^{E}_{o_3} = \gamma'^{E}_{e_3} + \gamma'^{E}_{r_3} = 3 + 4 = 7$$

Finally, the total number of optical ports ϱ'^{E} and the total number of electrical ports ε'^{E} are:

$$\varrho'^{E} = \gamma'^{E}_{o_1} + \gamma'^{E}_{o_3} = 14 + 7 = 21$$
$$\varepsilon'^{E} = \gamma'^{E}_{e_1} + \gamma'^{E}_{e_3} + \gamma'^{E}_{r_1} + \gamma'^{E}_{r_3} = 2 + 3 + 3 + 4 = 12$$

With κ being equal to 5, the total cost ζ'^{E} of routing and grooming the SED requests is evaluated as:

$$\zeta'^{E} = \varrho'^{E} + \kappa \times \varepsilon'^{E} = 21 + 5 \times 12 = 81$$

To sum up, by enabling the grooming in the network, the number of required o_1 optical ports has decreased from 18 to 14. This result confirms the values estimated in Example 5.2. Furthermore, the number of optical ports has been reduced from 23 to 21 at the price of an increase of the number of electrical ports from 10 to 12. As we mentioned before, this example shows that grooming does not drive systematically to a cost benefit. Consequently, when designing a grooming algorithm, aggregation is applied only in case of cost benefit.

5.4.3 Resource Sharing between SLDs and SEDs

In the two previous examples, the resources reserved for SLD requests and those reserved for SED requests are considered independently. However, it could be advantageous, when routing SED requests, to re-use some electrical ports and optical ports reserved for SLD requests if unused for some period of time. Going into details of the SLD and the SED routing solutions, only o_1 optical ports can be shared between these two types of requests. To this end, we make use of the matrices $\underline{\varPhi}^{\,L}$ and $\underline{\widetilde{\varPhi}}^{\,E}$ already introduced. An element $\phi^{L}_{t,l}$ of the former matrix represents the amount of o_1 optical ports needed at each end of the link e_l in order to satisfy all the SLDs at instant ϵ_t. Similarly, an element $\widetilde{\phi}^{E}_{t,l}$ of the latter matrix represents the amount of o_1 optical ports needed at each end of the link e_l in order to satisfy all the SEDs at instant ϵ_t. By suitably adding these two matrices (*i.e.,* these matrices are computed using the same set $\mathcal{E}$ of set-up/tear-down dates, and consequently they have the same size), the resulting matrix $\underline{\varPhi}^{\,G} = \{\phi^{G}_{t,l}\}$ represents the traffic load over time carried by the network links while satisfying both SLDs and SEDs. An element $\phi^{G}_{t,l}$ of such a matrix is an integer value representing the number of WDM optical channels carried by link e_l at instant ϵ_t. For a given link e_l, let φ^{G}_{l} be the maximum value of this traffic $\phi^{G}_{t,l}$ evaluated over the whole observation period. φ^{G}_{l} gives the number of WDM optical channels used on this link. As the o_1 optical ports go by pair, one port at each end of a WDM channel, φ^{G}_{l} represents also the number of o_1 optical port pairs installed at both ends of the link e_l. The number $\gamma^{G}_{o_1}$ of o_1 optical ports in the network is twice the sum of φ^{G}_{l} over all the network links.

$$\underline{\varPhi}^{\,G} = \underline{\varPhi}^{\,L} + \underline{\widetilde{\varPhi}}^{\,E} \tag{5.32a}$$

$$\varphi^{G}_{l} = \max_{1 \le t \le \mathbb{E}} \phi^{G}_{t,l} \tag{5.32b}$$

$$\gamma^{G}_{o_1} = 2 \times \sum_{l=1}^{L} \varphi^{G}_{l} \tag{5.32c}$$

It should be noted that the overall number of o_1 optical ports needed to route both the SLDs and the SEDs is, in most cases, smaller than the number of o_1 optical ports needed to route the SLDs and the SEDs separately.

$$\gamma_{o_1}^G \leqslant \gamma_{o_1}^L + \gamma_{o_1}^E \tag{5.33}$$

5.4.4 Lower Bound on the Network Cost in the Case of SEDs

By analyzing the network design problem, a lower bound on the network cost can be driven. One notices that the number of required o_1 optical ports reaches a minimum when each SED request passes through the EXC at every intermediate node along its path. Indeed, one proceed to a systematic aggregation at each node, optimizing by this way the bandwidth utilization of each optical channel. Moreover, the number of required electrical ports reaches a minimum when the SEDs are groomed together without the need to additional ports at the intermediate nodes. This is achieved when multiple SEDs reuse the same resources thanks to there time disjointness. As a result, one can compute a lower bound on the overall network cost. This lower bound depends only on the route assigned to each request. It is to be noted that this lower bound is rarely achievable because it is almost impossible to groom all the SEDs without the need to additional ports at the intermediate nodes. Hence, this theoretical lower bound solution can only be used to evaluate the performance of the various algorithms without being practically feasible. Such comparison is driven in Section 5.7 dedicated to the numerical results.

5.5 The Linear Programming Approach

Let us recall that, when routing SLDs, the number of emitting e_2 electrical ports, the number of receiving r_2 electrical ports, and the number of o_2 optical ports are independent of the routing solution. Consequently, the cost of the network is expressed as a function of only the number of o_1 optical ports. However, when routing SEDs, only the number of emitting e_1 electrical ports and the number of receiving r_1 electrical ports remain invariant. Consequently, the cost of the network is expressed as a function of the number of emitting e_3 electrical ports, the number of receiving r_3 electrical ports, and the number of o_1 and o_3 optical ports. In this section, we consider three separated problems. The first problem consists in determining the cheapest routing for SLDs assuming a set of precomputed K-shortest paths for each source-destination node pair. In this way, we reduce the number of variables to be considered when solving the Integer Linear Programming (ILP) formulation. The second problem also deals with SLDs routing but without any path pre-computation. For the same problem scenario, the latter approach drives to a more cost-effective solution compared to the former approach. However, this second approach is more complex, which makes it hard to get some results for practical networks size within reasonable computation time. Finally, as a third approach, we deal with the routing and grooming problem of a set of SEDs assuming precomputed K-shortest paths.

5.5.1 Mathematical ILP Formulation of the SLDs' Routing Problem assuming K-shortest Path Pre-computation

In this section, we consider a set of K-shortest paths computed off-line for each source-destination node pair. Consequently, the problem can be formulated as follows:

Parameters

- The network physical topology is represented by a set $V = \{v_i, i = 1 \cdots N\}$ of N EXC/OXC nodes and a set $E = \{e_i, i = 1 \cdots L\}$ of L unidirectional fiber-links interconnecting these nodes.

- The set $\mathcal{D}^L = \{\delta_i^L, i = 1 \cdots M^L\}$ of M^L SLD requests $\delta_i^L\big(s_i, d_i, \alpha_i, \beta_i, n_i = 1.0\big)$.

- For each SLD δ_i^L, the set $\mathcal{P}_i = \{P_{i,j}, j = 1 \cdots K\}$ of K available shortest paths connecting its source node s_i to its destination node d_i. $P_{i,j}$ is an L-dimensional vector ($P_{i,j} = [P_{i,j}^k, k = 1 \cdots L]$) denoting whether the j^{th}-shortest path traverses the directed link e_k ($P_{i,j}^k = 1$), or not ($P_{i,j}^k = 0$).

- The ordered set $\mathcal{E} = \{\epsilon_i, i = 1 \cdots \mathbb{E}\}$ of the set-up dates of all the SLDs.

- The binary request matrix $\underline{\mathit{\Delta}}^L = \{\delta_{i,j}^L, i = 1 \cdots M^L, j = 1 \cdots \mathbb{E}\}$. $\delta_{i,j}^L = 1$, if the SLD δ_i^L is active at instant ϵ_j. $\delta_{i,j}^L = 0$, otherwise.

Solution

For each SLD δ_i^L, determine the value of ρ_i ($1 \leqslant \rho_i \leqslant K$) and assign the corresponding ρ_i^{th}-shortest path to this SLD. This routing solution is represented by $\mathit{\Pi}_{\mathcal{D}^L,\rho} = \{P_{i,\rho_i}, i = 1 \cdots M^L\}$.

Variables

- The binary variables $\mathfrak{p}_{i,j}$, $i = 1 \cdots M^L$, $j = 1 \cdots K$.
  $\mathfrak{p}_{i,j} = 1$, if the j^{th}-shortest path between s_i and d_i is assigned to request δ_i^L. $\mathfrak{p}_{i,j} = 0$, otherwise.

- The binary routing matrix $\underline{R}^L = \{r_{i,j}^L, i = 1 \cdots M^L, j = 1 \cdots L\}$.
  $r_{i,j}^L = 1$, if the fiber-link e_j is part of the path assigned to SLD δ_i^L. $r_{i,j}^L = 0$, otherwise.

- The integer variables φ_i^L, $i = 1 \cdots L$.
  φ_i^L is the number of o_1 optical port pairs installed at both ends of the fiber-link e_i.

Constraints

- A unique path must be assigned to each SLD request δ_i^L.

$$\sum_{j=1}^{K} \mathfrak{p}_{i,j} = 1, \qquad \forall i = 1 \cdots M^L \tag{5.34}$$

- The variable $r_{i,j}^L$ defined in Equation (5.6) is computed as follows:

$$r_{i,j}^L = \sum_{k=1}^{K} \mathfrak{p}_{i,k} \times P_{i,k}^j, \qquad \forall i = 1 \cdots M^L, \forall j = 1 \cdots L \tag{5.35}$$

- The variable φ_i^L defined in Equation (5.9) is computed as follows:

$$\varphi_i^L \geqslant \sum_{k=1}^{M^L} \delta_{k,j}^L \times r_{k,i}, \qquad \forall i = 1 \cdots L, \forall j = 1 \cdots \mathbb{E} \tag{5.36}$$

Optional Constraints

- When needed, the congestion in the network can be limited to a given value $\beth$ by:

$$\beth \geqslant \varphi_i^L, \qquad \forall i = 1 \cdots L \tag{5.37}$$

Objective

Minimize the network cost ζ^L already formulated by Equation (5.15):

$$\zeta^L \propto \gamma_{o_1}^L = 2 \times \sum_{i=1}^{L} \varphi_i^L \tag{5.38}$$

5.5.2 General Mathematical ILP Formulation of the SLDs' Routing Problem

Parameters

- The network physical topology is represented by a set $V = \{v_i, i = 1 \cdots N\}$ of N EXC/OXC nodes and a set $E = \{e_i, i = 1 \cdots L\}$ of L unidirectional fiber-links interconnecting these nodes. e_i^s denotes the source node of the link e_i, while e_i^d denotes the destination node of this same link.

- The set $\mathcal{D}^L = \{\delta_i^L, i = 1 \cdots M^L\}$ of M^L SLD requests $\delta_i^L\big(s_i, d_i, \alpha_i, \beta_i, n_i = 1.0\big)$.

- The ordered set $\mathcal{E} = \{\epsilon_i, i = 1 \cdots \mathbb{E}\}$ of the set-up dates of all the SLDs.

- The binary request matrix $\underline{\Delta}^L = \{\delta_{i,j}^L, i = 1 \cdots M^L, j = 1 \cdots \mathbb{E}\}$. $\delta_{i,j}^L = 1$, if the SLD δ_i^L is active at instant ϵ_j. $\delta_{i,j}^L = 0$, otherwise.

Solution

For each SLD δ_i^L, determine the set of consecutive physical links that this SLD passes through.

Variables

- The binary routing matrix $\underline{R}^L = \{r_{i,j}^L, i = 1 \cdots M^L, j = 1 \cdots L\}$.
  $r_{i,j}^L = 1$, if the fiber-link e_j is part of the path assigned to SLD δ_i^L. $r_{i,j}^L = 0$, otherwise.

- The integer variables φ_i^L, $i = 1 \cdots L$.
  φ_i^L is the number of o_1 optical port pairs installed at both ends of the fiber-link e_i.

Constraints

- Except for the source node and the destination node of the SLD δ_i^L, each node must satisfy the flow conservation constraint with regard to the traffic flow carried by this SLD. In other words, the traffic flow of a request δ_i^L entering a node is equal to its traffic flow leaving this same node (other than the source node and the destination node of the request). At the source node of δ_i^L, the traffic flow of the request is only leaving the node. Similarly, at the destination node of δ_i^L, the traffic flow of the request is only entering the node. This can be formulated as follows:

$$\sum_{k=1 \backslash j=e_k^s}^{L} r_{i,k}^L - \sum_{k=1 \backslash j=e_k^d}^{L} r_{i,k}^L = \begin{cases} 0 & \text{if } j = 1 \cdots N \backslash \; j \neq s_i \text{ and } j \neq d_i, \\ 1 & \text{if } j = s_i, \\ -1 & \text{if } j = d_i, \end{cases} \quad , \forall i = 1 \cdots M^L \tag{5.39}$$

- The variable φ_i^L defined in Equation (5.9) is computed as follows:

$$\varphi_i^L \geqslant \sum_{k=1}^{M^L} \delta_{k,j}^L \times r_{k,i}, \qquad \forall i = 1 \cdots L, \forall j = 1 \cdots \mathbb{E} \tag{5.40}$$

Optional Constraints

- When needed, the congestion in the network can be limited to a given value $\beth$ by:

$$\beth \geqslant \varphi_i^L, \qquad \forall i = 1 \cdots L \tag{5.41}$$

Objective

Minimize the network cost ζ^L already formulated by Equation (5.15):

$$\zeta^L \propto \gamma_{o_1}^L = 2 \times \sum_{i=1}^{L} \varphi_i^L \tag{5.42}$$

5.5.3 Mathematical ILP Formulation of the SEDs' Routing and Grooming Problem assuming K-shortest Path Pre-computation

In this section, we consider a set of K-shortest paths computed off-line for each source-destination node pair. Moreover, we compute off-line all possible decompositions of these shortest paths into their inherent sub-paths as introduced at the beginning of Section 5.4. The problem is formulated as follows:

Parameters

- The network physical topology is represented by a set $V = \{v_i, i = 1 \cdots N\}$ of N EXC/OXC nodes and a set $E = \{e_i, i = 1 \cdots L\}$ of L unidirectional fiber-links interconnecting these nodes.

- The set $\mathcal{D}^E = \{\delta_i^E, i = 1 \cdots M^E\}$ of M^E SED requests $\delta_i^E\big(s_i, d_i, \alpha_i, \beta_i, n_i\big)$.

- For each SED δ_i^E, the set $\mathcal{P}_i = \{P_{i,j}, j = 1 \cdots K\}$ of K available shortest paths connecting its source node s_i to its destination node d_i. $P_{i,j}$ is an L-dimensional vector ($P_{i,j} = [P_{i,j}^k,\ k = 1 \cdots L]$) denoting whether the j^{th}-shortest path traverses the directed link e_k ($P_{i,j}^k = 1$), or not ($P_{i,j}^k = 0$).

- Let $\mathcal{P}$ be the set of all the shortest paths in the network. $\mathcal{P} = \bigcup P_{i,j},\ i = 1 \cdots M^E,\ j = 1 \cdots K$.

- Let ℓ_{max} be the maximum number of hops of the paths in the set $\mathcal{P}$.

- For each SED δ_i^E, each potential physical shortest path $P_{i,j} \in \mathcal{P}_i$ is divided into one or several sub-paths; each sub-path corresponding to a potential GL. Let us recall that a GL is a direct connection between two nodes (which are not necessarily adjacent) acting as a logical one-hop link. A possible decomposition $L_{i,j,m}$ of this physical path $P_{i,j}$ into its inherent sub-paths corresponds to a potential logical path followed by the SED δ_i^E in the logical topology. There exist $\mathbb{L}_{i,j}$ different decompositions. As mentioned in Section 5.4, $\mathbb{L}_{i,j}$ depends on the length of the physical path $P_{i,j}$ expressed in number of hops. If $P_{i,j}$ is composed of $\ell_{i,j}$ hops, $\mathbb{L}_{i,j}$ is equal to $2^{\ell_{i,j}-1}$.

- For a given SED δ_i^E and a given physical path $P_{i,j}$, let $\mathcal{L}_{i,j} = \{L_{i,j,m},\ m = 1 \cdots \mathbb{L}_{i,j}\}$ be the set of all the $\mathbb{L}_{i,j}$ possible decompositions of this physical path into its inherent GLs. The size $\mathbb{L}_{i,j}$ of such a set $\mathcal{L}_{i,j}$ varies from one request to another and from one path to another. However, the largest set $\mathcal{L}_{i,j}$ is composed of $\mathbb{L}_{max} = 2^{\ell_{max}-1}$ different logical paths $L_{i,j,m}$.

- Let $\mathcal{L} = \{\Game_n,\ n = 1 \cdots \Bbbk\}$ be the set of all the possible grooming lightpaths $\Game_n$ in the network. These GLs are defined based on all the physical paths $P_{i,j}$ $(i = 1 \cdots M^E,\ j = 1 \cdots K)$ and all their corresponding sub-paths $L_{i,j,m}$ $(i = 1 \cdots M^E,\ j = 1 \cdots K,\ m = 1 \cdots \mathbb{L}_{i,j})$. Let $\Bbbk$ be the size of this set ($\Bbbk = |\mathcal{L}|$). $\Bbbk$ represents the maximum number of GLs that can be established in the logical topology.

- A logical path $L_{i,j,m}$ $(i = 1 \cdots M^E,\ j = 1 \cdots K,\ m = 1 \cdots \mathbb{L}_{i,j})$ can be represented by a $\Bbbk$-dimensional vector ($L_{i,j,m} = [L_{i,j,m}^n, n = 1 \cdots \Bbbk]$) denoting whether the GL $\Game_n$ is a sub-path in $L_{i,j,m}$ ($L_{i,j,m}^n = 1$), or not ($L_{i,j,m}^n = 0$).

- A GL $\Game_n$ $(n = 1 \cdots \Bbbk)$ can be represented by an L-dimensional vector ($\Game_n = [\Game_n^k, k = 1 \cdots L]$) denoting whether the GL traverses through the directed link e_k ($\Game_n^k = 1$), or not ($\Game_n^k = 0$).

- The ordered set $\mathcal{E} = \{\epsilon_i, i = 1 \cdots \mathbb{E}\}$ of the set-up/tear-down dates of all the SEDs.

- κ $(\kappa > 1)$ the multiplicative factor representing the ratio of the cost of an electrical port to the cost of an optical port.

- The binary request matrix $\underline{\mathit{\Delta}}^E = \{\delta_{i,j}^E, i = 1 \cdots M^E, j = 1 \cdots \mathbb{E}\}$. $\delta_{i,j}^E = 1$, if the SED δ_i^E is active at instant ϵ_j. $\delta_{i,j}^E = 0$, otherwise.

- The binary groomed source matrix $\underline{\widetilde{S}}^E = \{\widetilde{s}_{i,n}^E, i = 1 \cdots \Bbbk, n = 1 \cdots N\}$. $\widetilde{s}_{i,n}^E = 1$, if the GL $\Game_i$ has node v_n as a source node. $\widetilde{s}_{i,n}^E = 0$, otherwise.

- The binary groomed destination matrix $\underline{\widetilde{D}}^E = \{\widetilde{d}_{i,n}^E, i = 1 \cdots \Bbbk, n = 1 \cdots N\}$. $\widetilde{d}_{i,n}^E = 1$, if the GL $\Game_i$ has node v_n as a destination node. $\widetilde{s}_{i,n}^E = 0$, otherwise.

Solution

For each SED δ_i^E, first determine the value of ρ_i $(1 \leqslant \rho_i \leqslant K)$ and assign the corresponding ρ_i^{th}-shortest path to this SED. Then, determine the value of λ_{i,ρ_i} corresponding to the adopted logical path in the logical topology. This routing and grooming solution is represented by $\mathit{\Pi}_{\mathcal{D}^E,\rho,\lambda} = \{(P_{i,\rho_i}, L_{i,\rho_i,\lambda_{i,\rho_i}}),\ i = 1 \cdots M^E)\}$.

Variables

- The binary variables $\mathfrak{p}_{i,j}$, $i = 1 \cdots M^E$, $j = 1 \cdots K$.
  $\mathfrak{p}_{i,j} = 1$, if the j^{th}-shortest path between s_i and d_i is assigned to request δ_i^E. $\mathfrak{p}_{i,j} = 0$, otherwise.

- The binary variables $\mathfrak{L}_{i,j,m}$, $i = 1 \cdots M^E$, $j = 1 \cdots K$, $m = 1 \cdots \mathbb{L}_{i,j}$.
  $\mathfrak{L}_{i,j,m} = 1$, if the m^{th}-decomposition into sub-paths of the physical path $P_{i,j}$ is adopted for the request δ_i^E. $\mathfrak{L}_{i,j,m} = 0$, otherwise.

- The groomed request matrix $\underline{\widetilde{\varDelta}}^E = \{\widetilde{\delta}_{i,t}^E, i = 1 \cdots \Bbbk, t = 1 \cdots \mathbb{E}\}$. $\widetilde{\delta}_{i,t}^E$ is an integer variable representing the number of WDM channels needed to carry the traffic load of the GL $\Game_i$ at time instant ϵ_t.

- The integer variables $\widetilde{\varphi}_i^E$, $i = 1 \cdots L$.
  $\widetilde{\varphi}_i^E$ is the number of o_1 optical port pairs installed at both ends of the fiber-link e_i.

- The integer variables $\widetilde{\upsilon}_i^{E,Em}$, $i = 1 \cdots N$.
  $\widetilde{\upsilon}_i^{E,Em}$ is the number of emitting e_3 electrical ports installed at node v_i.

- The integer variables $\widetilde{\upsilon}_i^{E,Re}$, $i = 1 \cdots N$.
  $\widetilde{\upsilon}_i^{E,Re}$ is the number of receiving r_3 electrical ports installed at node v_i.

Constraints

- A unique physical path must be assigned to each SED request δ_i^E.

$$\sum_{j=1}^{K} \mathfrak{p}_{i,j} = 1, \qquad \forall i = 1 \cdots M^E \tag{5.43}$$

- A unique logical path must be assigned to each SED request δ_i^E. This is achieved by selecting an appropriate decomposition of the chosen physical path into a set of GLs.

$$\sum_{m=1}^{\mathbb{L}_{i,j}} \mathfrak{L}_{i,j,m} = \mathfrak{p}_{i,j}, \qquad \forall i = 1 \cdots M^E, \forall j = 1 \cdots K. \tag{5.44}$$

- The variable $\widetilde{\delta}_{i,t}^E$ defined in Equation (5.19) is computed as follows:

$$\sum_{\substack{1 \leqslant j \leqslant M^E \\ 1 \leqslant k \leqslant K \\ 1 \leqslant m \leqslant \mathbb{L}_{j,k}}} n_j \times \delta_{j,t}^E \times \mathfrak{L}_{j,k,m} \times L_{j,k,m}^i \leqslant \widetilde{\delta}_{i,t}^E < \sum_{\substack{1 \leqslant j \leqslant M^E \\ 1 \leqslant k \leqslant K \\ 1 \leqslant m \leqslant \mathbb{L}_{j,k}}} n_j \times \delta_{j,t}^E \times \mathfrak{L}_{j,k,m} \times L_{j,k,m}^i + 1, \quad \begin{array}{l} \forall i = 1 \cdots \Bbbk, \\ \forall t = 1 \cdots \mathbb{E}. \end{array} \tag{5.45}$$

  Let us recall that the required data rate n_j of SED δ_j^E is a real number whereas it is equal to 1 in the case of SLDs.

- The variable $\widetilde{\upsilon}_i^{E,Em}$ defined in Equation (5.23) is computed as follows:

$$\widetilde{\upsilon}_i^{E,Em} \geqslant \sum_{k=1}^{\Bbbk} \widetilde{\delta}_{k,j}^E \times \widetilde{s}_{k,i}^E, \qquad \forall i = 1 \cdots N, \forall j = 1 \cdots \mathbb{E}. \tag{5.46}$$

- The variable $\widetilde{\upsilon}_i^{E,Re}$ defined in Equation (5.24) is computed as follows:

$$\widetilde{\upsilon}_i^{E,Re} \geqslant \sum_{k=1}^{\Bbbk} \widetilde{\delta}_{k,j}^E \times \widetilde{d}_{k,i}^E, \qquad \forall i = 1 \cdots N, \forall j = 1 \cdots \mathbb{E}. \tag{5.47}$$

- The variable $\widetilde{\varphi}_i^E$ defined in Equation (5.25) is computed as follows:

$$\widetilde{\varphi}_i^E \geqslant \sum_{k=1}^{\Bbbk} \widetilde{\delta}_{k,j}^E \times \Game_k^i, \qquad \forall i = 1 \cdots L, \forall j = 1 \cdots \mathbb{E}. \tag{5.48}$$

Objective

Minimize the network cost ζ^E already formulated by Equation (5.31):

$$\begin{aligned}\zeta^E &\propto \kappa \times (\gamma_{e_3}^E + \gamma_{r_3}^E) + \gamma_{o_1}^E + \gamma_{o_3}^E \\ &\propto (\kappa + 1) \times \sum_{i=1}^{L} \left(\widetilde{\upsilon}_i^{E,Em} + \widetilde{\upsilon}_i^{E,Re}\right) + 2 \times \sum_{i=1}^{L} \widetilde{\varphi}_i^E \end{aligned} \tag{5.49}$$

5.6 The Heuristic Approach

It is well known that the RWA optimization problem is *NP*-Complete [C49, J134]. If we assume that each connection request requires the full capacity of an optical channel, the traffic grooming problem we are studying reduces to the standard RWA optimization problem. Therefore, traffic grooming is inherently more difficult than the well-known *NP*-Complete RWA problem. Even when all the network nodes are equipped with wavelength converters (in which case wavelength assignment is trivial), the traffic grooming problem remains intractable.

We have formulated the optimization problem as an Integer Linear Program (ILP). When the network size is small and/or the number of requests is limited, some commercial softwares [W168] can be used to solve the ILP equations and obtain an optimal solution. The limitation of the ILP approach is that the number of variables and the number of equations increase explosively as the size of the network increases and/or the number of requests increases. This computational complexity makes it hard to be useful for networks of practical size. However, by means of efficient heuristic algorithms, it may be possible to get some results for large size networks. These results are sub-optimal solutions. The longer the computation time (number of iterations), the closer the results of these heuristics to the optimal solution.

In our proposed heuristic approach, designing a grooming-capable network is divided into two sub-problems which are solved separately. The first sub-problem routes the SLDs and the SEDs over the physical topology. Due to their deterministic aspect, the requests (SLDs and SEDs) can be routed by means of global optimization tools such as the Simulated Annealing heuristic (SA). For an instance of the SEDs routing solution, the second sub-problem deals with grooming several SEDs onto GLs. The adopted strategy is an Iterative process based on a Greedy algorithm (IG). The proposed IG algorithm considers single-hop grooming as well as multi-hop grooming. Such off-line optimization techniques may need Central Processing Unit (CPU) times ranging from a few minutes to several hours depending on the network size and the amount of traffic demands. The objective is to route all the SxDs at the lowest cost expressed as the number of ports used. When routing the SLDs, the cost benefit is obtained by means of the network resource reutilization thanks to the knowledge of time and space correlation between the demands. However, when routing the SEDs, the cost benefit is obtained by means of the network resource reutilization as well as by means of the grooming process already discussed in Section 5.2. These centralized off-line computations are carried out by the management plane which operates in the long term. Once the SxDs are routed, the management plane informs each node of the availability of the equipment (electrical ports and optical ports) within the network.

5.6.1 Simulated Annealing Algorithm for SLD and SED Routing

Simulated Annealing [J135, W169] (along with genetic algorithms) has been found to provide good solutions for complex optimization problems while avoiding local minima. It is based on the idea of taking a random walk through the solution space at successively lower temperatures, where the probability of taking a step is given by a Boltzmann distribution.

In our context, the simulated annealing process starts by assigning to each request the shortest path between its source node and its destination node. This routing solution determines the initial size of each EXC/OXC node in the physical topology. Path-exchange operations are used to generate neighboring configurations. In a path-exchange operation, the paths assigned to a random number of demands are simultaneously modified. For instance, we can choose to change the route assigned to 4 different requests. The traffic load of each of the selected requests is shifted from its current path to another alternate path. These alternate paths are chosen from the set of K-shortest paths already computed off-line for each source-destination node pair. It is to be noted that the higher the iteration index, the lower the number of path reassignments (size of the neighborhood). When the requests to be routed have sub-wavelength components, this neighboring configuration includes also the optimization of the grooming process as explained later. New neighboring configurations which give better results (lower network cost expressed in terms of electrical ports and optical ports) than the current solution are accepted automatically. Solutions with higher cost than the current one are accepted with a certain probability which is determined by a system control parameter. The probability with which these more expensive configurations are chosen, however, decreases as the algorithm progresses in time so as to simulate the "cooling" process associated with annealing. This probability is based on a negative exponential factor and is inversely proportional to the difference between the cost of the current solution and the cost of the new solution.

During the initial phase of the annealing process (high temperature value), we examine random configurations in the search space so as to obtain different initial starting configurations without getting stuck at a local minimum as in a greedy approach. However, as time progresses, the probability of accepting more expensive solutions decreases, and the algorithm settles down into a minimum after several iterations. The state becomes "frozen" when there is no improvement in the objective function of the solution even after a large number of iterations.

A more general problem is the dimensioning of the size of the EXC and the OXC switches while satisfying a limited number of optical channels per fiber-link. The previous problem ignores this constraint on the network congestion and thus can be viewed as an aspect of the more general problem. The flowchart in Figure 5.15 shows the structure of the whole algorithm. In this flowchart, we use the following notations:

- X_0 denotes the initial value of the variable X at the beginning of the algorithm.
- X_c denotes the current value of the variable X.
- X_n denotes a new value assigned to the variable X.
- X_b denotes the best value obtained so far for the variable X during the algorithm.

Additional details are provided by its corresponding pseudo-code in Algorithm 5.1.

5.6.2 Iterative Greedy Algorithm for SED Grooming

As shown in Section 5.2, the grooming process implies the use of additional e_3, r_3, and o_3 ports within the network but it reduces the number of required o_1 ports. Our optimization problem is then based on a trade-off between the cost penalty due to the additional ports and the cost benefit due to the reduction in the number of the o_1 ports. It is to be noted that the overall cost of the network can be reduced thanks to the resource reutilization between different requests. Section 5.2 also points out

Algorithm 5.1 Simulated Annealing Algorithm

```
Input: $\mathcal{D}$ (* set of requests *) ,
       $\daleth_{max}$ (* maximim allowable congestion *) ,
       $T_0$ and $Cnt_{max}$ (* parameters of the algorithm *)
Output: The path assigned to each request and the number of electrical ports and optical ports to be installed
        at each node
Begin
Step 1. Initializing
1          Initial Solution $\Pi_0$ constructed by assigning the shortest path to each request. $\zeta_0$ is its cost and $\daleth_0$ is its
           congestion.
2          Current Solution $\Pi_c$ := Initial Solution $\Pi_0$
3          $Cnt := 0$
Step 2. Congestion Control
4          while ($\daleth_c > \daleth_{max}$) do
4.1            if $Cnt > Cnt_{max}$ then
4.1.1              return Unable to satisfy the congestion constraint.
               else
4.1.2              Apply perturbation. New Solution: $\Pi_n(\zeta_n, \daleth_n)$
4.1.3              $Cnt := Cnt + 1$
4.1.4              case ($\daleth_n - \daleth_c$) of
                     <0 :
4.1.4.1                  $Cnt := 0$
4.1.4.2                  Accept new solution $\Pi_c := \Pi_n$
                     =0 :
4.1.4.3                  case ($\zeta_n - \zeta_c$) of
                           <0 :
4.1.4.3.1                      Accept new solution $\Pi_c := \Pi_n$
                           =0 :
4.1.4.3.2                      Randomly accept/reject the new solution
                         endcase
                   endcase
               endif
           endwhile
Step 3. Cost Optimization
5          $Cnt := 0$
6          $T := T_0$
7          Best Solution $\Pi_b$ := Current Solution $\Pi_c$
8          while ($Cnt < Cnt_{max}$) do
8.1            Apply Perturbation. New Solution: $\Pi_n(\zeta_n, \daleth_n)$
8.2            $Cnt := Cnt + 1$
8.3            if ($\daleth_n < \daleth_{max}$) then
8.3.1              case ($\zeta_n - \zeta_c$) of
                     =0 :
8.3.1.1                  Randomly accept/reject the new solution
                     >0 :
8.3.1.2                  Generate a random number $A$
8.3.1.3                  if $\left(A < \exp^{((\daleth_c - \daleth_n)/T)}\right)$ then
8.3.1.3.1                    Accept new solution $\Pi_c := \Pi_n$
                         endif
                     otherwise
8.3.1.4                  $Cnt := 0$
8.3.1.5                  Accept new solution $\Pi_c := \Pi_n$
8.3.1.6                  if ($\daleth_n < \daleth_b$) then
8.3.1.6.1                    Save as best solution. $\Pi_b := \Pi_n$
                         endif
                   endcase
               endif
8.4            $T := \beta \times T$
           endwhile
9          return Output Best Solution $\Pi_b$
End.
```

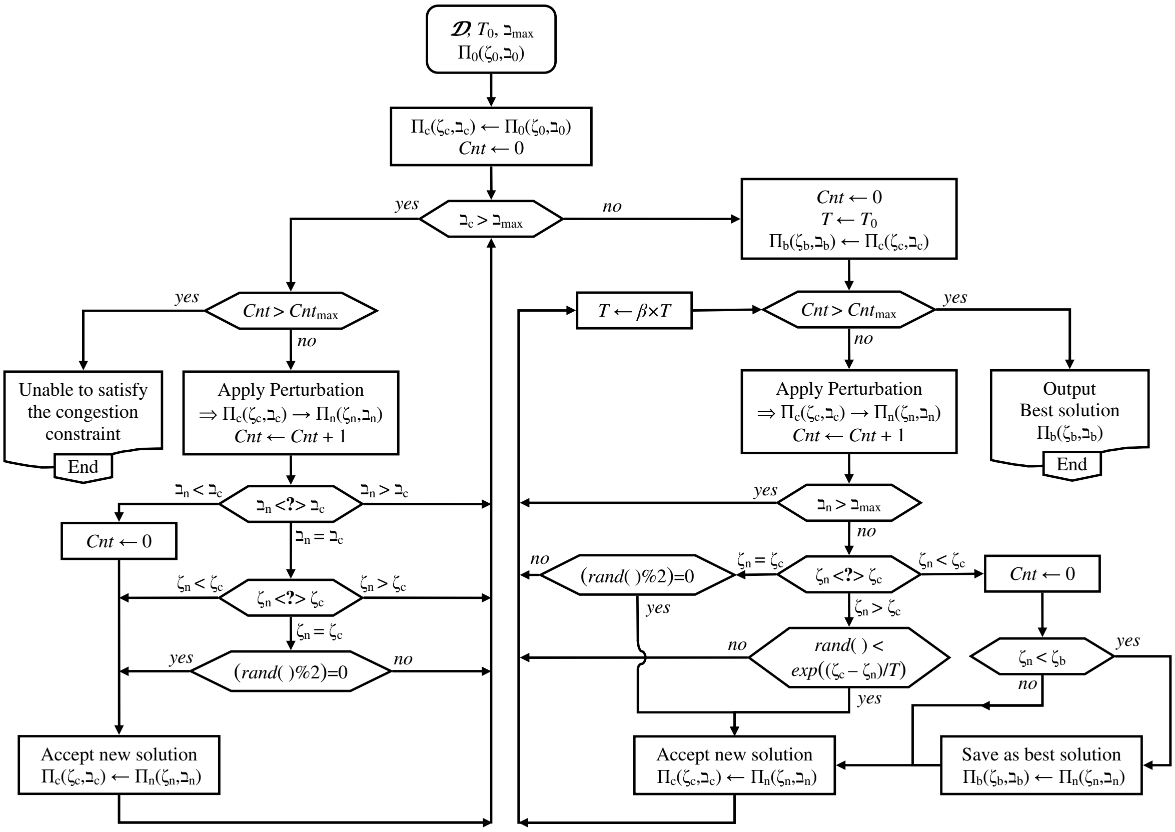


Fig. 5.15. Flowchart of the Simulated Annealing algorithm

that the grooming process does not alter the paths assigned to the requests being groomed. Hence, it is necessary to assign to each request a path between its source node and its destination node before optimizing the grooming. The paths assigned to some SEDs are modified by means of the SA algorithm just described in Section 5.6.1. Once the paths are modified, the grooming optimization will be re-initiated.

An exhaustive algorithm may be used to groom each possible pair of demands. But this method is time consuming and requires considerable memory resources. In order to reduce the complexity of the problem, we attempt to groom the requests two by two at a time. By a series of consecutive grooming processes inherent to an iterative approach, multiple traffic requests can be groomed together on the same GL. The depth of the aggregation process (*i.e.,* the maximum number of demands that can be aggregated on the same GL) is directly linked to the required capacity of each demand as well as to the computation time of the algorithm. The number of iterations carried out by the adopted algorithm is not infinite, and the algorithm stops when certain criteria are satisfied. This limited number of iterations impacts the depth of the aggregation process.

Before detailing the proposed algorithm, it is useful to define some expressions and concepts:

Common Path Length ($CPL_{i,j}$): it is a function that indicates if two demands δ_i^E and δ_j^E satisfy the grooming constraints or not. It is mainly based on the number of physical links on which δ_i^E and δ_j^E overlap. The overlapping time period can be another additional criteria to compute $CPL_{i,j}$.

$$CPL_{i,j} = \begin{cases} 0 & \text{if } \delta_i^E \text{ and } \delta_j^E \text{ are not simultaneous in time} \\ & \text{or they do not share any common fiber-link.} \\ \mathbf{p} & \mathbf{p} \text{ being the maximum number of consecutive} \\ & \text{common links/hops used by } \delta_i^E \text{ and } \delta_j^E. \end{cases} \tag{5.50}$$

Successful Grooming Pair (SGP): it is a pair of demands (δ_i^E, δ_j^E) with a positive CPL ($CPL_{i,j} > 0$) such that, after grooming together these two demands, the overall network cost is smaller than or equal to the cost of the current routing and grooming solution.

Unsuccessful Grooming Pair (UGP): it is a pair of demands (δ_i^E, δ_j^E) with a positive CPL ($CPL_{i,j} > 0$) such that, after grooming together these two demands, the overall network cost is higher than the cost of the current routing and grooming solution.

We aim to compare our iterative greedy strategy with other grooming algorithms proposed in the literature. The criteria of such comparison are mainly the computation time and the network cost obtained at the last iteration. Before detailing our proposed Iterative Greedy algorithm, we introduce two retained algorithms referred to as *"Greedy1"* and *"Greedy2"*.

Greedy1 Algorithm

By grooming the pair of demands (δ_i^E, δ_j^E) of common path length $CPL_{i,j}$, the reduction in the number of required o_1 optical ports is equal to $(2 \times CPL_{i,j})$. However, this grooming process requires additional e_3, r_3, and o_3 ports at the ingress and the egress grooming points. This increase in the number of e_3, r_3, and o_3 ports can be reduced thanks to the resource reutilization/sharing between several SEDs. Moreover, this increase in the number of e_3, r_3, and o_3 ports can be compensated by the decrease in the number of o_1 optical ports. The longer the common path length (high $CPL_{i,j}$ values), the higher the reduction of the number of required o_1 ports, and the more important the expected gain. The basic idea of the Greedy1 approach is intuitive where we always try to groom pair of demands with large CPL value before we try to groom pair of demands with smaller CPL value. The flowchart in Figure 5.16 shows the structure of the Greedy1 algorithm. Additional details are provided by its corresponding pseudo-code in Algorithm 5.2.

As the grooming process does not modify the path assigned to each request, the algorithm starts by assigning a path to each request. In this initial solution $\Pi_0(\zeta_0, \beth_0)$, each request uses separate network resources without undergoing any grooming process. For each possible pair of requests, we compute its CPL value. The pair with the largest CPL is selected and its requests are groomed together. Consequently, the set $\mathcal{D}'$ of groomed demands is updated by removing the original requests and by adding the groomed request and the set of marginal demands as already introduced in Section 5.2. The network cost ζ_n and the observed congestion $\beth_n$ of this new routing and grooming solution Π_n are computed. If the cost ζ_n of the new solution is smaller than or equal to the cost ζ_c of the current one, the new solution Π_n is retained; otherwise, this pair of requests is marked as Unsuccessful Grooming Pair (UGP) and is stored in T_list. It is to be noted that some resources may be added to the network when we groom two requests. Thus, a pair of demands marked as UGP may benefit from the additional resources and becomes a Successful Grooming Pair (SGP). For this reason, when we accept a new solution, the list T_list of all the UGPs is cleared. In addition, grooming two requests yields a new set of groomed demands. This implies that the set of all possible pairs of requests as well as their corresponding CPL values must be updated accordingly. A new solution is initiated by choosing another pair of demands with the largest CPL value and that is not already marked as UGP.

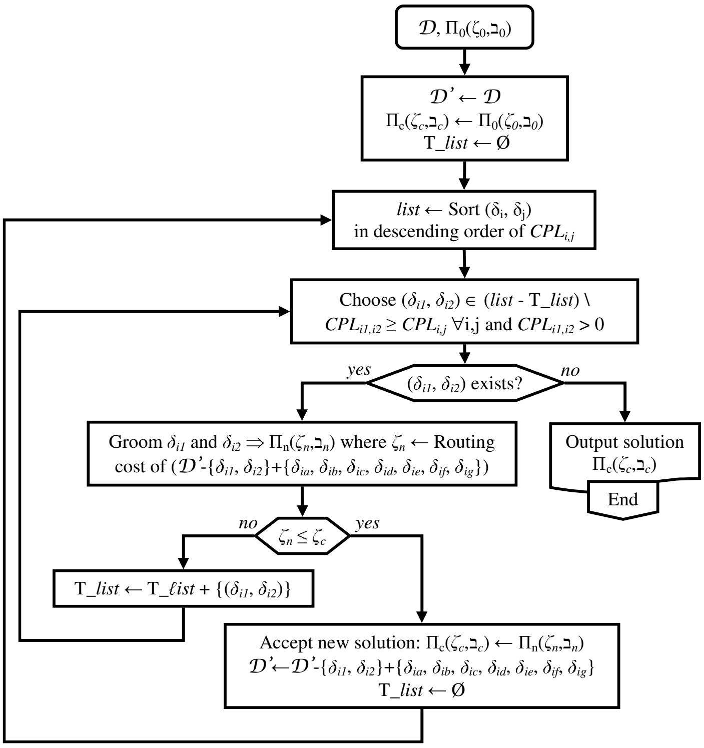

𝒟, Π0(ζ0,ℶ0)
𝒟' ← 𝒟
Πc(ζc,ℶc) ← Π0(ζ0,ℶ0)
T_list ← Ø
list ← Sort (δi, δj)
in descending order of CPLi,j
Choose (δi1, δi2) ∈ (list - T_list) \
CPLi1,i2 ≥ CPLi,j ∀i,j and CPLi1,i2 > 0
yes
(δi1, δi2) exists?
no
Groom δi1 and δi2 ⇒ Πn(ζn,ℶn) where ζn ← Routing
cost of (𝒟'-{δi1, δi2}+{δia, δib, δic, δid, δie, δif, δig})
Output solution
Πc(ζc,ℶc)
End
no
ζn ≤ ζc
yes
T_list ← T_list + {(δi1, δi2)}
Accept new solution: Πc(ζc,ℶc) ← Πn(ζn,ℶn)
𝒟'←𝒟'-{δi1, δi2}+{δia, δib, δic, δid, δie, δif, δig}
T_list ← Ø

Fig. 5.16. Flowchart of the Greedy1 algorithm

Algorithm 5.2 Greedy1 Algorithm

```
Input: $\mathcal{D}$ (* set of requests *)
Output: The best routing and grooming solution that satisfies this set of requests $\mathcal{D}$ at the lowest cost as
        well as the number of electrical ports and optical ports to be installed at each node
Begin
Step 1. Initializing
1      Initial Solution $\Pi_0$ constructed by assigning a path to each request without undergoing any grooming
       process. $\zeta_0$ is its cost and $\beth_0$ is its congestion.
2      Current Solution $\Pi_c$ := Initial Solution $\Pi_0$
3      The set of groomed demands $\mathcal{D}' := \mathcal{D}$
4      $T\_list := \emptyset$
Step 2. Grooming Process
5      Form $list$ by sorting $(\delta_i, \delta_j)$ in descending order of $CLP_{i,j}$.
6      Choose $(\delta_{i1}, \delta_{i2}) \in (list - T\_list) \setminus CLP_{i1,i2} \geqslant CLP_{i,j}\ \forall i,j$ and $CLP_{i1,i2} > 0$.
7      while $(\delta_{i1}, \delta_{i2})$ exists do
7.1        Groom $\delta_{i1}$ and $\delta_{i2}$. New solution $\Pi_n(\zeta_n, \beth_n)$.
           $\zeta_n$ routing cost of $\big(\mathcal{D}' - \{\delta_{i1}, \delta_{i2}\} + \{\delta_{ia}, \delta_{ib}, \delta_{ic}, \delta_{id}, \delta_{ie}, \delta_{if}, \delta_{ig}\}\big)$
7.2        if $(\zeta_n \leq \zeta_c)$ then
7.2.1          Accept new solution. $\Pi_c := \Pi_n$
7.2.2          $\mathcal{D}' := \mathcal{D}' - \{\delta_{i1}, \delta_{i2}\} + \{\delta_{ia}, \delta_{ib}, \delta_{ic}, \delta_{id}, \delta_{ie}, \delta_{if}, \delta_{ig}\}$
7.2.3          $T\_list := \emptyset$
7.2.4          Update $list$ by sorting $(\delta_i, \delta_j)$ in descending order of $CLP_{i,j}$.
           else
7.2.5          $T\_list := T\_list + \{(\delta_{i1}, \delta_{i2})\}$
           endif
7.3        Choose $(\delta_{i1}, \delta_{i2}) \in (list - T\_list) \setminus CLP_{i1,i2} \geqslant CLP_{i,j}\ \forall i,j$ and $CLP_{i1,i2} > 0$.
       endwhile
8      return Output Solution $\Pi_c$.
End.
```

This procedure is repeated until no new pair of demands with positive CPL value can be found or until the T_list has reached its maximum capacity.

Greedy2 Algorithm

In the Greedy1 approach, as grooming two requests is carried out frequently, the set $\mathcal{D}'$ of groomed demands is often updated. After each update, we need to recompute the CPL value for all the possible pair of requests in the new set $\mathcal{D}'$, and then we need to choose the pair of requests with the largest CPL. It appears that this operation is time consuming. In addition, in order to accomplish an acceptable number of grooming iterations, the size of the list T_list must be set to a high value. As a result, the Greedy1 algorithm requires also considerable memory resources. In order to get around these drawbacks, the Greedy2 algorithm is proposed. Instead of attempting to groom the pair of demands with the largest CPL value, the Greedy2 approach attempts to groom any pair of requests with a positive CPL value. At each iteration of this algorithm, we analyze the pair of demands in a random order. The first pair of demands with a positive CPL value is selected and its requests are groomed together. If the cost ζ_n of the new solution is smaller than or equal to the cost ζ_c of the current one, this aggregation is accepted; otherwise, this pair of requests is marked as UGP and is stored in T_list. Then, we continue analyzing the remaining pair of demands. The flowchart in Figure 5.17 shows the structure of the Greedy2 algorithm. Additional details are provided by its corresponding pseudo-code in Algorithm 5.3.

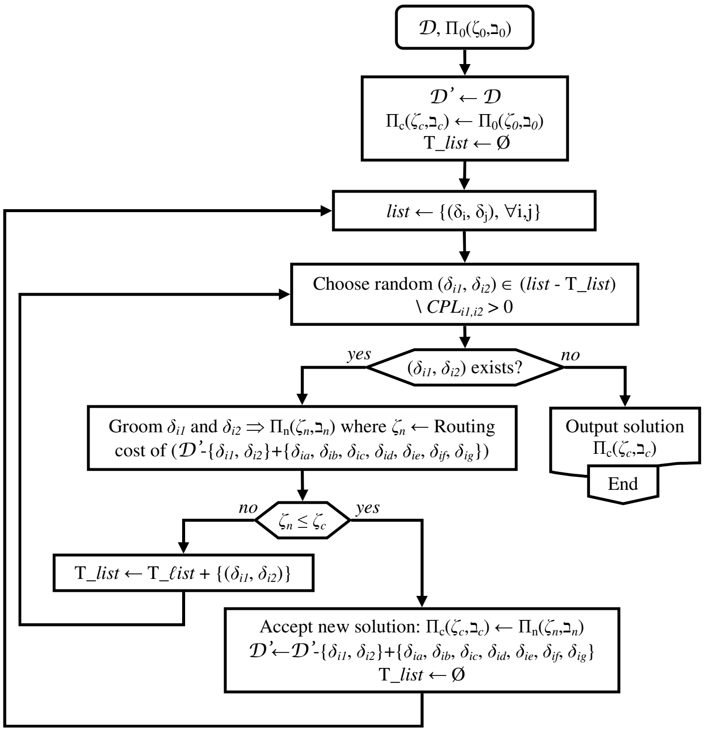

$\mathcal{D}$, $\Pi_0(\zeta_0,\beth_0)$
$\mathcal{D}' \leftarrow \mathcal{D}$
$\Pi_c(\zeta_c,\beth_c) \leftarrow \Pi_0(\zeta_0,\beth_0)$
$T\_list \leftarrow \emptyset$
$list \leftarrow \{(\delta_i, \delta_j), \forall i,j\}$
Choose random $(\delta_{i1}, \delta_{i2}) \in (list - T\_list) \setminus CPL_{i1,i2} > 0$
yes
$(\delta_{i1}, \delta_{i2})$ exists?
no
Groom δ_{i1} and $\delta_{i2} \Rightarrow \Pi_n(\zeta_n,\beth_n)$ where $\zeta_n \leftarrow$ Routing cost of $(\mathcal{D}'$-$\{\delta_{i1}, \delta_{i2}\}+\{\delta_{ia}, \delta_{ib}, \delta_{ic}, \delta_{id}, \delta_{ie}, \delta_{if}, \delta_{ig}\})$
Output solution $\Pi_c(\zeta_c,\beth_c)$
End
no
$\zeta_n \leq \zeta_c$
yes
$T\_list \leftarrow T\_\ell ist + \{(\delta_{i1}, \delta_{i2})\}$
Accept new solution: $\Pi_c(\zeta_c,\beth_c) \leftarrow \Pi_n(\zeta_n,\beth_n)$
$\mathcal{D}' \leftarrow \mathcal{D}'$-$\{\delta_{i1}, \delta_{i2}\}+\{\delta_{ia}, \delta_{ib}, \delta_{ic}, \delta_{id}, \delta_{ie}, \delta_{if}, \delta_{ig}\}$
$T\_list \leftarrow \emptyset$

Fig. 5.17. Flowchart of the Greedy2 algorithm

The Proposed Iterative Greedy (IG) Algorithm

On one hand, the Greedy1 algorithm is time consuming and requires considerable memory resources. On the other hand, the Greedy2 algorithm does not guarantee to achieve near-optimal solution. We

Algorithm 5.3 Greedy2 Algorithm

```
Input: $\mathcal{D}$ (* set of requests *)
Output: The best routing and grooming solution that satisfies this set of requests $\mathcal{D}$ at the lowest cost as
        well as the number of electrical ports and optical ports to be installed at each node
Begin
Step 1. Initializing
Initial Solution $\mathit{\Pi}_0$ constructed by assigning a path to each request without undergoing any grooming
    process. $\zeta_0$ is its cost and $\beth_0$ is its congestion.
Current Solution $\mathit{\Pi}_c$ := Initial Solution $\mathit{\Pi}_0$
The set of groomed demands $\mathcal{D}' := \mathcal{D}$
$T\_list := \emptyset$
Step 2. Grooming Process
$list := \{(\delta_i, \delta_j),\ \forall i, j\}$.
Choose a random $(\delta_{i1}, \delta_{i2}) \in (list - T\_list) \setminus CLP_{i1,i2} > 0$.
while $(\delta_{i1}, \delta_{i2})$ exists do
    Groom $\delta_{i1}$ and $\delta_{i2}$. New solution $\mathit{\Pi}_n(\zeta_n, \beth_n)$.
    $\zeta_n$ routing cost of $\big(\mathcal{D}' - \{\delta_{i1}, \delta_{i2}\} + \{\delta_{ia}, \delta_{ib}, \delta_{ic}, \delta_{id}, \delta_{ie}, \delta_{if}, \delta_{ig}\}\big)$
    if $(\zeta_n \leq \zeta_c)$ then
        Accept new solution. $\mathit{\Pi}_c := \mathit{\Pi}_n$
        $\mathcal{D}' := \mathcal{D}' - \{\delta_{i1}, \delta_{i2}\} + \{\delta_{ia}, \delta_{ib}, \delta_{ic}, \delta_{id}, \delta_{ie}, \delta_{if}, \delta_{ig}\}$
        $T\_list := \emptyset$
        Update $list := \{(\delta_i, \delta_j),\ \forall i, j\}$.
    else
        $T\_list := T\_list + \{(\delta_{i1}, \delta_{i2})\}$
    endif
    Choose a random $(\delta_{i1}, \delta_{i2}) \in (list - T\_list) \setminus CLP_{i1,i2} > 0$.
endwhile
return Output Solution $\mathit{\Pi}_c$.
End.
```

propose the Iterative Greedy (IG) algorithm as a new approach that achieves near-optimal grooming solutions in a very short time compared to the Greedy1 approach. We can distinguish two main steps:

- **Step 1:** The Greedy1 approach always attempts to groom the pair of requests with the largest CPL value. However, grooming pairs of demands with small CPL values can also be beneficent due to resource reutilization by non-simultaneous requests. To this end, we group the pair of requests into sub-groups; each sub-group is characterized by a constant CPL value. Starting with the sub-group characterized by the largest CPL value, we attempt to groom pairs of requests within this set as described by the Greedy1 algorithm. When all the pairs of demands within this sub-group are tested and marked as UGP or when the T_list has reached its maximum capacity, we consider the next sub-group characterized by a one unit smaller CPL value. For this new sub-group, we reset the list T_list and retry to groom additional pairs of demands within this sub-group. This is repeated until all the possible values of the CPL are considered. It is to be noted that for a given sub-group, the Greedy1 algorithm and the Greedy2 algorithm are almost the same because all the pair of requests in this sub-group have the same CPL value.

  Another advantage of this decomposition is that the size of the subgroups are smaller than the original group. Consequently:

  - The size of the list T_list in our IG approach can be set to a smaller value than in the case of the Greedy1 approach. This results in considerable savings in memory resources. Let L_1 be the size of the T_list in the IG approach.
  - When grooming two requests, the number of CPL values that need to be recomputed is smaller than in the Greedy1 approach. Thus, our approach is faster than the Greedy1 approach.

- **Step 2:** Till now, pairs of demands with large CPL values cannot take advantage of the resources added by pairs of demands with smaller CPL values groomed later. For this reason, we make a last attempt to groom pairs of demands without fixing any value for the CPL and starting from its largest value as is the case of the Greedy1 approach. Let L_2 be the size of the T_list for this last attempt.

Finally, one can choose to repeat **Step 1** several times before going on to **Step 2**. Let N_1 be the number of times the **Step 1** is repeated. The flowchart in Figure 5.18 shows the structure of the Iterative Greedy algorithm. Additional details are provided by its corresponding pseudo-code in Algorithm 5.4.

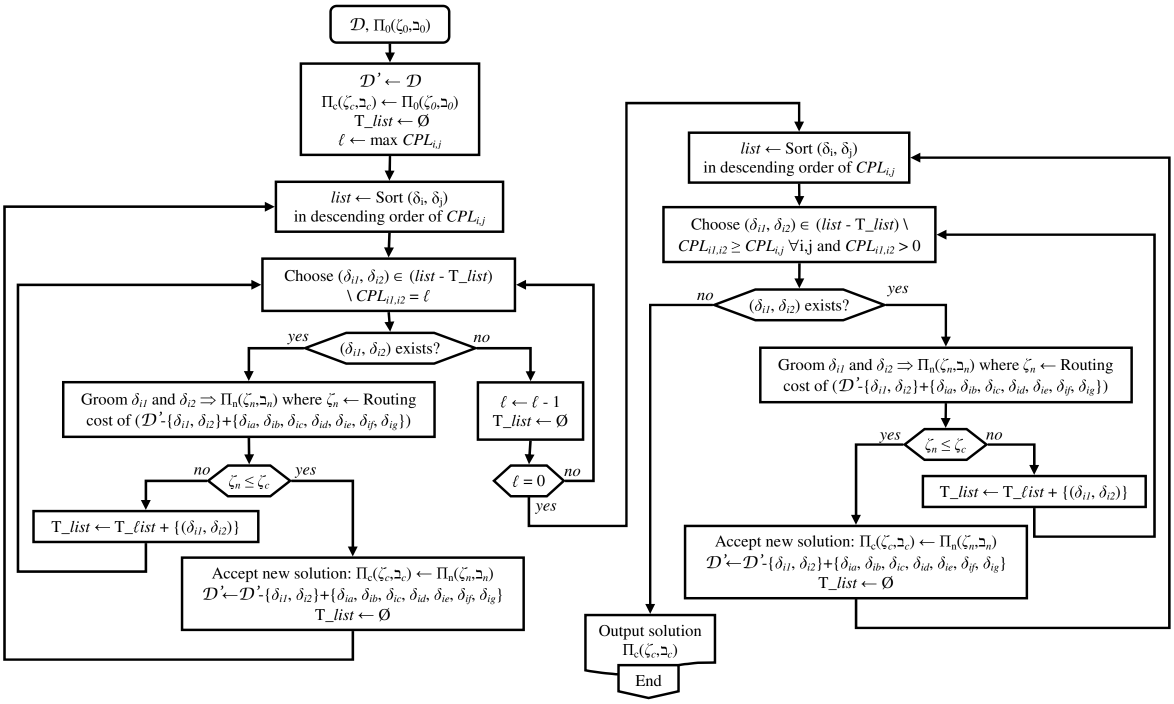

$\mathcal{D}$, $\Pi_0(\zeta_0,ב_0)$
$\mathcal{D}' \leftarrow \mathcal{D}$
$\Pi_c(\zeta_c,ב_c) \leftarrow \Pi_0(\zeta_0,ב_0)$
T_*list* ← Ø
$\ell \leftarrow \max CPL_{i,j}$
list ← Sort (δ_i, δ_j)
in descending order of $CPL_{i,j}$
Choose $(\delta_{i1}, \delta_{i2}) \in$ (*list* - T_*list*)
\ $CPL_{i1,i2} = \ell$
yes
$(\delta_{i1}, \delta_{i2})$ exists?
no
Groom δ_{i1} and $\delta_{i2} \Rightarrow \Pi_n(\zeta_n,ב_n)$ where $\zeta_n \leftarrow$ Routing
cost of $(\mathcal{D}'\text{-}\{\delta_{i1}, \delta_{i2}\}+\{\delta_{ia}, \delta_{ib}, \delta_{ic}, \delta_{id}, \delta_{ie}, \delta_{if}, \delta_{ig}\})$
$\ell \leftarrow \ell$ - 1
T_*list* ← Ø
no
$\zeta_n \leq \zeta_c$
yes
$\ell = 0$
no
yes
T_*list* ← T_*ℓist* + $\{(\delta_{i1}, \delta_{i2})\}$
Accept new solution: $\Pi_c(\zeta_c,ב_c) \leftarrow \Pi_n(\zeta_n,ב_n)$
$\mathcal{D}' \leftarrow \mathcal{D}'\text{-}\{\delta_{i1}, \delta_{i2}\}+\{\delta_{ia}, \delta_{ib}, \delta_{ic}, \delta_{id}, \delta_{ie}, \delta_{if}, \delta_{ig}\}$
T_*list* ← Ø
list ← Sort (δ_i, δ_j)
in descending order of $CPL_{i,j}$
Choose $(\delta_{i1}, \delta_{i2}) \in$ (*list* - T_*list*) \
$CPL_{i1,i2} \geq CPL_{i,j}$ ∀i,j and $CPL_{i1,i2} > 0$
no
$(\delta_{i1}, \delta_{i2})$ exists?
yes
Groom δ_{i1} and $\delta_{i2} \Rightarrow \Pi_n(\zeta_n,ב_n)$ where $\zeta_n \leftarrow$ Routing
cost of $(\mathcal{D}'\text{-}\{\delta_{i1}, \delta_{i2}\}+\{\delta_{ia}, \delta_{ib}, \delta_{ic}, \delta_{id}, \delta_{ie}, \delta_{if}, \delta_{ig}\})$
yes
$\zeta_n \leq \zeta_c$
no
T_*list* ← T_*ℓist* + $\{(\delta_{i1}, \delta_{i2})\}$
Accept new solution: $\Pi_c(\zeta_c,ב_c) \leftarrow \Pi_n(\zeta_n,ב_n)$
$\mathcal{D}' \leftarrow \mathcal{D}'\text{-}\{\delta_{i1}, \delta_{i2}\}+\{\delta_{ia}, \delta_{ib}, \delta_{ic}, \delta_{id}, \delta_{ie}, \delta_{if}, \delta_{ig}\}$
T_*list* ← Ø
Output solution
$\Pi_c(\zeta_c,ב_c)$
End

Fig. 5.18. Flowchart of the Iterative Greedy algorithm

5.7 Simulation Results and Analysis

Extensive simulations have been carried out in order to compare the performance of the various algorithms proposed in this chapter. In the following, and according to the current technology, the ratio κ representing the cost of an electrical port to the cost of an optical port is fixed to 5. We define the computation time as the period of time required to complete the optimization process on a 3 GHz Intel Pentium IV processor with 1 Giga Bytes of RAM memory.

5.7.1 SLD Routing: ILP vs SA

The purpose of the routing optimization of the SLD demands is to outline the cost benefit obtained by means of channel reuse thanks to the knowledge of time and space correlation between the SLDs. This cost benefit only depends on the number of required optical ports. Indeed, the number of electrical

Algorithm 5.4 Iterative Greedy Algorithm

```
Input: $\mathcal{D}$ (* set of requests *)
Output: The best routing and grooming solution that satisfies this set of requests $\mathcal{D}$ at the lowest cost as
        well as the number of electrical ports and optical ports to be installed at each node
Begin
Step 1. Initializing
1    Initial Solution $\Pi_0$ constructed by assigning a path to each request without undergoing any grooming
     process. $\zeta_0$ is its cost and $\beth_0$ is its congestion.
2    Current Solution $\Pi_c$ := Initial Solution $\Pi_0$
3    The set of groomed demands $\mathcal{D}' := \mathcal{D}$
4    $T\_list := \emptyset$
Step 2. Grooming by fixed $CPL$ values
5    Form $list$ by sorting $(\delta_i, \delta_j)$ in descending order of $CLP_{i,j}$.
6    for $\ell := \max_{i,j} CPL_{i,j}$ to $1$ do
6.1        Choose $(\delta_{i1}, \delta_{i2}) \in (list - T\_list) \setminus CLP_{i1,i2} := \ell$.
6.2        while $(\delta_{i1}, \delta_{i2})$ exists do
6.2.1            Groom $\delta_{i1}$ and $\delta_{i2}$. New solution $\Pi_n(\zeta_n, \beth_n)$.
                 $\zeta_n$ routing cost of $\big(\mathcal{D}' - \{\delta_{i1}, \delta_{i2}\} + \{\delta_{ia}, \delta_{ib}, \delta_{ic}, \delta_{id}, \delta_{ie}, \delta_{if}, \delta_{ig}\}\big)$
6.2.2            if $(\zeta_n \leq \zeta_c)$ then
6.2.2.1              Accept new solution. $\Pi_c := \Pi_n$
6.2.2.2              $\mathcal{D}' := \mathcal{D}' - \{\delta_{i1}, \delta_{i2}\} + \{\delta_{ia}, \delta_{ib}, \delta_{ic}, \delta_{id}, \delta_{ie}, \delta_{if}, \delta_{ig}\}$
6.2.2.3              $T\_list := \emptyset$
6.2.2.4              Update $list$ by sorting $(\delta_i, \delta_j)$ in descending order of $CLP_{i,j}$.
                 else
6.2.2.5              $T\_list := T\_list + \{(\delta_{i1}, \delta_{i2})\}$
                 endif
6.2.3            Choose $(\delta_{i1}, \delta_{i2}) \in (list - T\_list) \setminus CLP_{i1,i2} := \ell$.
           endwhile
6.3        $T\_list := \emptyset$
     endfor
Step 3. Grooming by descending $CPL$ values
7    Update $list$ by sorting $(\delta_i, \delta_j)$ in descending order of $CLP_{i,j}$.
8    Choose $(\delta_{i1}, \delta_{i2}) \in (list - T\_list) \setminus CLP_{i1,i2} \geqslant CLP_{i,j}\ \forall i,j$ and $CLP_{i1,i2} > 0$.
9    while $(\delta_{i1}, \delta_{i2})$ exists do
9.1        Groom $\delta_{i1}$ and $\delta_{i2}$. New solution $\Pi_n(\zeta_n, \beth_n)$.
           $\zeta_n$ routing cost of $\big(\mathcal{D}' - \{\delta_{i1}, \delta_{i2}\} + \{\delta_{ia}, \delta_{ib}, \delta_{ic}, \delta_{id}, \delta_{ie}, \delta_{if}, \delta_{ig}\}\big)$
9.2        if $(\zeta_n \leq \zeta_c)$ then
9.2.1          Accept new solution. $\Pi_c := \Pi_n$
9.2.2          $\mathcal{D}' := \mathcal{D}' - \{\delta_{i1}, \delta_{i2}\} + \{\delta_{ia}, \delta_{ib}, \delta_{ic}, \delta_{id}, \delta_{ie}, \delta_{if}, \delta_{ig}\}$
9.2.3          $T\_list := \emptyset$
9.2.4          Update $list$ by sorting $(\delta_i, \delta_j)$ in descending order of $CLP_{i,j}$.
           else
9.2.5          $T\_list := T\_list + \{(\delta_{i1}, \delta_{i2})\}$
           endif
9.3        Choose $(\delta_{i1}, \delta_{i2}) \in (list - T\_list) \setminus CLP_{i1,i2} \geqslant CLP_{i,j}\ \forall i,j$ and $CLP_{i1,i2} > 0$.
     endwhile
10   return Output Solution $\Pi_c$.
End.
```

ports used at the source node and the destination node of each demand remains unchanged for the various routing solutions. For this purpose, we mainly consider two sets of traffic requests: SxD_2 and SxD_3 (*c.f.* Section 3.6). The former is a set of 1000 SLDs corresponding to connections to be established on the 9-node, 12-link NSF network, while the latter is a set of 2000 SLDs corresponding to connections to be established on the 29-node, 44-link NSF network. For each set of requests, we dimension a network able to handle these traffic demands using three different algorithms: an ILP model based on a set of pre-computed K-shortest path, a general ILP model without any path pre-computation, and a heuristic approach based on Simulated Annealing algorithm. The networks with the lowest costs are retained from the different algorithms and are compared between them. Table 5.7 and Table 5.8 summarize the costs of these networks for the traffic sets SxD_2 and SxD_3, respectively.

Table 5.7. Network cost in the case of 1000 SLDs based on the 9-node, 12-link NSF network topology

		Shortest Path	ILP Model with K-shortest path pre-computation			General ILP Model	SA with K-shortest path pre-computation		
			$K = 2$	$K = 4$	$K = 10$		$K = 2$	$K = 4$	$K = 10$
Optical Ports	o_1-ports	1744	1706	1664	1664	1664	1706	1674	1676
	o_2-ports	858	858	858	858	858	858	858	858
	o_3-ports	-	-	-	-	-	-	-	-
Emitting Electrical Ports	e_1-ports	-	-	-	-	-	-	-	-
	e_2-ports	431	431	431	431	431	431	431	431
	e_3-ports	-	-	-	-	-	-	-	-
Receiving Electrical Ports	r_1-ports	-	-	-	-	-	-	-	-
	r_2-ports	427	427	427	427	427	427	427	427
	r_3-ports	-	-	-	-	-	-	-	-
Overall Cost		6892	6854	6812	6812	6812	6854	6822	6824
Network Congestion		80					75	61	63
Computation Time		-					217 sec	374 sec	452 sec

When routing the set of 1000 SLDs, the number of required electrical e_2-ports is equal to 431 and the number of required electrical r_2-ports is equal to 427 independently of the chosen routing solution. This is due to the fact that several SLDs share the same source and/or the same destination with at least another time disjoint SLD. When these 1000 SLDs are routed according to the shortest path between the various source-destination node pairs, the overall cost of the network is evaluated to 6892. In this case, only the time correlation is considered. By allowing some requests to be routed along alternate paths, we improve the resource sharing between non-simultaneous SLDs. Consequently, both space and time correlations are considered in this case. For instance, when we pre-compute a set of 4 shortest paths for each source-destination node pair and by using an exact approach, the number of required o_1 optical ports decreases from 1744 to 1664. This represents a gain of 4.59%. However, it is to be noted that the time needed to obtain this result using the exact approach is of 1 to 2 days long. In addition, we highlight that the ILP approach and the heuristic approach perform similarly. Indeed, compared to the shortest path routing solution, the proposed SA algorithm combined with a set of 4 shortest paths pre-computed for each source-destination node pair allows the number of required o_1 optical ports to decrease from 1744 to 1674. This represents a gain of 4.01%. One notices also that the congestion decreases from 80 to 61. It is to be noted that the SA computing time to get this result is of 6 to 7 minutes long.

Similarly, when routing the set of 2000 SLDs, the number of required electrical e_2-ports is equal to 806 and the number of required electrical r_2-ports is equal to 808 independently of the chosen routing solution. When these 2000 SLDs are routed according to the shortest path between the various source-destination node pairs, the overall cost of the network is evaluated to 15734. By allowing some requests to be routed along alternate paths, we improve the resource sharing between non-simultaneous SLDs. For instance, when we pre-compute a set of 4 shortest paths for each source-destination node pair and by using an exact approach, the number of required o_1 optical ports decreases from 6050 to 5386. This

Table 5.8. Network cost in the case of 2000 SLDs based on the 29-node, 44-link NSF network topology

		Shortest Path	ILP Model with K-shortest path pre-computation			General ILP Model	SA with K-shortest path pre-computation		
			$K = 2$	$K = 4$	$K = 10$		$K = 2$	$K = 4$	$K = 10$
Optical Ports	o_1-ports	6050	5568	5386	5328	x	5584	5480	5524
	o_2-ports	1614	1614	1614	1614	x	1614	1614	1614
	o_3-ports	-	-	-	-	x	-	-	-
Emitting Electrical Ports	e_1-ports	-	-	-	-	x	-	-	-
	e_2-ports	806	806	806	806	x	806	806	806
	e_3-ports	-	-	-	-	x	-	-	-
Receiving Electrical Ports	r_1-ports	-	-	-	-	x	-	-	-
	r_2-ports	808	808	808	808	x	808	808	808
	r_3-ports	-	-	-	-	x	-	-	-
Overall Cost		15734	15252	15070	15012	x	15268	15164	15208
Network Congestion		101				x	83	70	68
Computation Time		-				$\geqslant$ 11 days	6898 sec	7300 sec	15117 sec

represents a gain of 10.98%. However, the time needed to obtain this result using the exact approach is of 3 to 4 days long. It is to be noticed that the general ILP model has not reached any near-optimal solution even after 11 days of computation. Likewise, compared to the shortest path routing solution, the proposed SA algorithm combined with a set of 4 shortest paths pre-computed for each source-destination node pair allows the number of required o_1 optical ports to decrease from 6050 to 5480. This represents a gain of 9.42%. In this latter case, the congestion decreases from 101 to 70. It is to be noted that the SA computing time to get this result is roughly 2 hours long.

From these results, we conclude that the use of alternate shortest paths improves the resource sharing between SLD requests. The higher the number K of pre-computed shortest paths, the lower the number of required o_1-ports, and the higher the gain achieved by means of resource sharing. The highest gain is expected for the general ILP approach at the price of an extremely long computation time. However, having as few as four alternate routes provides the major benefit of resource sharing. In addition, our proposed heuristic approach achieves near-optimal solution in significantly reduced computation time. For these reasons, when routing SLD requests, we mainly consider, in the following, our SA approach combined with a set of 4 shortest paths pre-computed off-line for each source-destination node pair.

5.7.2 SLD Routing under Limited Number of Wavelengths per Fiber: ILP vs SA

In this section, we mainly focus on the network dimensioning problem for SLDs under limited network capacity. This limited network capacity is implemented by limiting the number of wavelengths per fiber. For this purpose, we consider only the set SxD_3 of 2000 SLD requests (*c.f.* Section 3.6). This set of SLDs is routed by means of an ILP model using pre-computed K-shortest paths and a SA algorithm using also pre-computed K-shortest paths. When $K = 4$, the minimal number of wavelengths per fiber achievable by the ILP model and by the SA algorithm without rejecting any connection request are 57 and 58, respectively. These values decrease to 54 for the ILP model and 55 for the SA algorithm when 10 shortest paths are considered between each source-destination node pair. Different values for the upper-bound on the number of wavelengths per fiber are considered. Table 5.9 and Table

5.10 summarize the network cost for different upper-bound values assuming 4 and 10 pre-computed shortest paths, respectively.

Table 5.9. ILP model versus SA algorithm with 4-shortest paths and additional constraint on the congestion

		Unconstrained	Constrained Model (ILP/SA)				
		Model (ILP/SA)	$\gimel \leqslant 57$	$\gimel \leqslant 58$	$\gimel \leqslant 60$	$\gimel \leqslant 65$	$\gimel \leqslant 70$
Optical Ports	o_1-ports	5386/5480	5434/x	x/5580	5406/5560	5388/5510	5386/5480
	o_2-ports	1614	1614	1614	1614	1614	1614
	o_3-ports	-	-	-	-	-	-
Emitting Electrical Ports	e_1-ports	-	-	-	-	-	-
	e_2-ports	806	806	806	806	806	806
	e_3-ports	-	-	-	-	-	-
Receiving Electrical Ports	r_1-ports	-	-	-	-	-	-
	r_2-ports	808	808	808	808	808	808
	r_3-ports	-	-	-	-	-	-
Overall Cost		15070/15164	15118/x	x/15264	15090/15244	15072/15194	15070/15164
Network Congestion		x/70	57	58	60	65	70

When dealing with networks with limited number of wavelengths per fiber, the number of o_1 optical ports obtained by means of both the ILP model and the SA algorithm increases compared to the case of unlimited network capacity. This increase is due to the fact that longer paths may be adopted to satisfy this capacity limitation. For instance, when considering the ILP model with 4 pre-computed shortest paths, limiting the number of wavelengths per fiber to 54 yields an increase in the network cost from 15070 to 15118. The penalty due to the network capacity constraint is thus equal to 0.32%. Similarly, when considering the SA algorithm combined with a set of 4 pre-computed shortest paths, limiting the number of wavelengths per fiber to 58 yields an increase in the network cost from 15164 to 15264. In this case, the penalty due to the network capacity constraint is equal to 0.66%. It is to be noted that the computation time for the ILP model is of 3 to 4 days long, while the computation time for the SA algorithm is of 2 to 3 hours long.

Table 5.10. ILP Model versus SA algorithm with 10-shortest paths and additional constraint on the congestion

		Unconstrained	Constrained Model (ILP/SA)				
		Model (ILP/SA)	$\gimel \leqslant 54$	$\gimel \leqslant 55$	$\gimel \leqslant 60$	$\gimel \leqslant 65$	$\gimel \leqslant 70$
Optical Ports	o_1-ports	5328/5524	5368/x	x/5726	5332/5618	5328/5528	5328/5524
	o_2-ports	1614	1614	1614	1614	1614	1614
	o_3-ports	-	-	-	-	-	-
Emitting Electrical Ports	e_1-ports	-	-	-	-	-	-
	e_2-ports	806	806	806	806	806	806
	e_3-ports	-	-	-	-	-	-
Receiving Electrical Ports	r_1-ports	-	-	-	-	-	-
	r_2-ports	808	808	808	808	808	808
	r_3-ports	-	-	-	-	-	-
Overall Cost		15012/15208	15052/x	x/15410	15016/15302	15012/15212	15012/15208
Network Congestion		x/68	54	55	60	65	68

Similar observations can be noticed when one considers 10 shortest paths computed off-line for each source-destination node pair. As in Section 5.7.1, we highlight in this section the small gap that exists between the performance of the ILP model and that of the SA algorithm. In addition, we outline the short computation time needed by the SA algorithm compared to that of the ILP model.

5.7.3 SED Routing: ILP vs SA

When routing the SEDs, the cost benefit is obtained by means of resource sharing between the SEDs themselves and by means of the grooming procedure. In this section, we focus only on the gain obtained due to channel reuse thanks to the knowledge of time and space correlation between SEDs. This cost benefit only depends on the number of required optical ports. Indeed, the number of electrical ports used at the source node and the destination node of each demand remains unchanged for the various routing solutions. For this purpose, we consider the set SxD_1 of 250 SED traffic requests (*c.f.* Section 3.6). These SEDs correspond to connection requests to be established on the 9-node, 12-link NSF network. A network able to handle these traffic demands is dimensioned by means of an ILP model based on a set of pre-computed K-shortest path without any grooming consideration and the proposed heuristic approach based on Simulated Annealing algorithm. Table 5.11 summarizes the cost of the network for both algorithms and for different sets of pre-computed shortest paths.

Table 5.11. ILP Model versus SA algorithm when routing a set of 250 SED requests

		Shortest Path	ILP Model with K-shortest path pre-computation		SA with K-shortest path pre-computation	
			$K = 2$	$K = 4$	$K = 2$	$K = 4$
Optical Ports	o_1-ports	552	534	516	534	516
	o_2-ports	-	-	-	-	-
	o_3-ports	251	251	251	251	251
Emitting Electrical Ports	e_1-ports	131	131	131	131	131
	e_2-ports	-	-	-	-	-
	e_3-ports	120	120	120	120	120
Receiving Electrical Ports	r_1-ports	131	131	131	131	131
	r_2-ports	-	-	-	-	-
	r_3-ports	120	120	120	120	120
Overall Cost		3313	3295	3277	3295	3277
Network Congestion		27			24	20
Computation Time		-			30 sec	30 sec

Compared to the shortest path without any grooming consideration, the best solution obtained thanks to resource reutilization ($K = 4$) represents a gain of 1.09% on the overall network cost. It is to be noted that the computational time of the ILP is of several hours, while the time needed by the SA algorithm to obtain these results is about 30 sec. Once more, we highlight the small gap that exists between the performance of the ILP model and that of the SA algorithm. In addition, we outline the short computation time needed by the SA algorithm compared to that of the ILP model.

5.7.4 SED Grooming: Various Approaches

Many algorithms dealing with the grooming problem are investigated in the literature. We retain two grooming algorithms, namely Greedy1 and Greedy2, that are initially designed for static traffic demands, and we adapt them to the case of dynamic scheduled demands (*c.f.* Section 5.6.2). Then, we compare the performance of these two algorithms with an exact approach based on an ILP model and with our proposed Iterative Greedy (IG) algorithm. The size of the T_list for both the Greedy1 and the Greedy2 algorithms is fixed to 100000 in order to achieve a sufficiently high number of iterations. However, the parameters L_1, L_2, and N_1 of our IG algorithm are fixed to 100, 1000, and 1, respectively (*c.f.* Section 5.6.2).

First, we reconsider the set SxD_1 of 250 SEDs introduced in Section 5.7.3. Without loss of generality, we suppose that these requests are routed along the shortest path between their source-destination node pairs. Table 5.12 summarizes the resources required by the various grooming algorithms in order to handle this set of SEDs.

Table 5.12. Network cost when grooming a set of 250 SEDs

		without grooming	ILP model	Greedy1 algorithm	Greedy2 algorithm	IG algorithm	Lower-bound
Optical Ports	o_1-ports	552	334	356	348	366	300
	o_2-ports	-	-	-	-	-	-
	o_3-ports	251	178	194	210	195	171
Emitting Electrical Ports	e_1-ports	131	131	131	131	131	131
	e_2-ports	-	-	-	-	-	-
	e_3-ports	131	93	98	106	99	89
Receiving Electrical Ports	r_1-ports	120	120	120	120	120	120
	r_2-ports	-	-	-	-	-	-
	r_3-ports	120	85	96	104	96	82
Overall Cost		3313	2657	2775	2863	2791	2581
Network Congestion		27	15	17	17	17	13
Computation Time		-	14 days	11660 sec	87 sec	4 sec	-

When no grooming is performed, the overall cost of the network is evaluated to 3313. By allowing the aggregation of low speed connections into high capacity GLs, the cheapest achievable solution is obtained by the ILP model and is equal to 2657. This represents a gain of 19.80%. A near-optimal solution (gain ≈ 16.24%) can be achieved by the Greedy1 algorithm, but the time needed for this task is of several hours. The Greedy2 algorithm decreases this computational time to 87 sec at the price of a higher overall network cost. Our IG approach seems a good compromise between the time needed to achieve a given solution and the cost penalty due to the sub-optimality of the solution. In our case, the IG algorithm achieves a gain of 15.76% in almost 4 sec. Note that the number of required optical o_1-ports reaches a minimum when each SED request passes through the EXC at each intermediate node that it traverses. When routing according to the shortest path algorithm, this minimum is equal to 300 o_1-ports. Moreover, the number of required electrical ports reaches a minimum when the SEDs are groomed together without any need to additional ports at intermediate nodes. As a result, one can compute a theoretical lower bound on the overall network cost. This lower bound depends only on the route assigned to each request. In our case, the lower bound on the network cost is equal to

2581. It is to be noted that this lower bound is rarely achievable because it is almost impossible to groom all the SEDs without the need to additional ports at the intermediate nodes. Indeed, when the grooming is performed at each intermediate node, the number of required optical ports is equal to 600 ports, while the number of required electrical ports is equal to 551. The cost of such a network is equal to 3355 which is higher than the estimated lower bound.

Now, let us consider a larger set of 5000 SED requests (SxD_4). In this case, no optimal solution can be obtained in reasonable computational time. Table 5.13 summarizes the resources required by the Greedy1, the Greedy2 and the IG algorithms in order to handle this set of 5000 SEDs.

Table 5.13. Network cost when grooming a set of 5000 SEDs

		without grooming	Greedy1 algorithm	Greedy2 algorithm	IG algorithm	Lower-bound
Optical Ports	o_1-ports	15176	14232	14322	10690	7742
	o_2-ports	-	-	-	-	-
	o_3-ports	4046	3939	3981	3331	2279
Emitting Electrical Ports	e_1-ports	2012	2012	2012	2012	2012
	e_2-ports	-	-	-	-	-
	e_3-ports	2012	1963	1989	1660	1131
Receiving Electrical Ports	r_1-ports	2034	2034	2034	2034	2034
	r_2-ports	-	-	-	-	-
	r_3-ports	2034	1976	1992	1671	1148
Overall Cost		59682	58096	58438	50906	41646
Network Congestion		254	230	244	168	126
Computation Time		-	12771 sec	10964 sec	3939 sec	-

When no grooming is performed, the overall cost of the network is evaluated to 59682. By grooming the SEDs according to the Greedy1 algorithm, the network cost decreases to 58096. This represents a gain of around 2.66%. In this case, the program has stopped after 12771 sec because the T_list has reached is maximum capacity. Thus, higher gains could have been achieved by increasing the size of the T_list. Similarly, by grooming the SEDs according to the Greedy2 algorithm, the overall network cost decreases from 59682 to 58438 during a 10964 sec time interval. This represents a gain of about 2.08%. Under the same traffic scenario, our IG approach achieves a gain of 14.70% in about 3939 sec of computation time.

To conclude, our proposed IG algorithm enables to obtain smaller network costs than those obtained with basic greedy strategies proposed in the literature in a considerably shorter computation time. In addition, our approach does not require a lot of memory resources. Indeed, the size of the T_list is fixed to 1000 for the IG algorithm, while it is fixed to 100000 for the Greedy1 and the Greedy2 algorithms.

5.7.5 Resource Sharing between SLDs and SEDs

Let us consider the set SxD_5 of 7000 requests. This set is decomposed into a subset of 2000 SLD requests and another subset of 5000 SED requests. The 2000 SLDs are routed by means of the SA algorithm. However, the 5000 SEDs are routed along the shortest path between their source-destination node pairs and are groomed using the IG algorithm. Table 5.14 shows the network cost corresponding to this routing solution.

Table 5.14. Network cost when considering both SLDs and SEDs

		Shortest Path without Grooming			SA (K=4) + IG		
		SLD	SED	SxD	SLD	SED	SxD
Optical Ports	o_1-ports	6050	15176	20594	5480	10690	15704
	o_2-ports	1614	-	1614	1614	-	1614
	o_3-ports	-	4046	4046	-	3331	3309
Emitting Electrical Ports	e_1-ports	-	2012	2012	-	2012	2012
	e_2-ports	806	-	806	806	-	806
	e_3-ports	-	2012	2012	-	1660	1648
Receiving Electrical Ports	r_1-ports	-	2034	2034	-	2034	2034
	r_2-ports	808	-	808	808	-	808
	r_3-ports	-	2034	2034	-	1671	1661
Overall Cost		15734	59682	74784	15164	50906	65472
Network Congestion		101	254	345	70	168	245
Computation Time		-	-	-	7300 sec	3938 sec	13597 sec

Compared to the case where the SLDs are routed along the shortest path, routing the SLDs according to the SA algorithm allows the network cost to decrease from 15734 to 15164. This represents a gain of 3.62%. Compared to the case where no grooming is performed, grooming the SEDs allows the network cost to decrease from 59682 to 50906. This represents a gain of 14.70%. By allowing some network resources reserved for SLDs to be re-used by SED requests, the network cost decreases from 66070 $(15164 + 50906)$ to 65472. This is possible since the SLD resources remain free for a period of time, and it yields an additional gain of 0.91%. To conclude, the network resources reutilization among SLDs, in addition to the grooming of the SEDs, and the resource sharing between SLDs and SEDs allow the network cost to decrease from 75416 $(15734+59682)$ to 65472. This represents a global gain of 13.19%.

5.7.6 Evolution over Time of the Heuristic Approaches

Figure 5.19 illustrates the routing cost of the traffic set SxD_3 expressed in terms of required electrical ports and optical ports over the SA computing time. The flat shape of the curve during the first iterations is due to the fact that the first perturbations applied to the initial solution do not result in a cost reduction. In addition, due to the large initial value of the parameter T (referred to as "Temperature"), the SA algorithm accepts a lot of more expensive routing solutions during this first stage. The SA computing time to get this curve is of 7300 sec. We note that WDM channel reuse reduces the network cost by 3.62% between the first and the last SA iteration. If the SA algorithm is stopped after 3000 sec of computation time, the number of required o_1 optical ports for the best solution obtained during this computational time is of 5568 ports. Thus, the overall network cost is of 15252 which represents a gain of 3.06%. The network congestion of this best solution is equal to 76.

Without loss of generality, we consider the set SxD_4 of 5000 SEDs where the requests are routed along the shortest path between their source-destination node pairs. When no grooming is performed, the global network cost is equal to 59682. In our Iterative Greedy approach, we limit the size of the

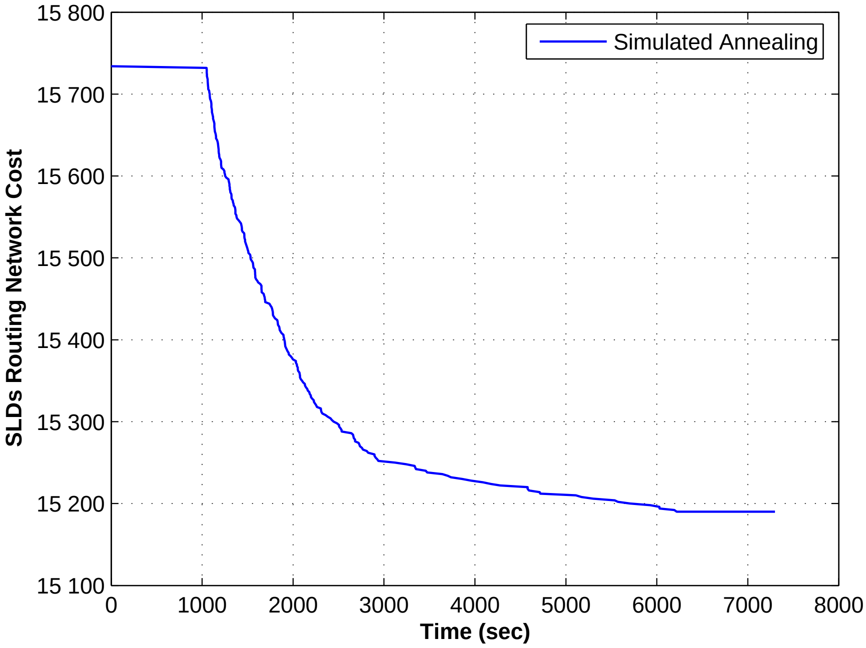


Fig. 5.19. The best solution over time obtained by means of the SA algorithm

T_list to 100 and 1000 in the **Step 1** and the **Step 2**, respectively. However, we limit the size of the T_list to 100000 for the Greedy1 and the Greedy2 algorithms. Figure 5.20 illustrates the grooming cost expressed in terms of required electrical ports and optical ports as a function of the computation time for the three algorithms. One notices the outperformance of our proposed algorithm compared to the Greedy1 and the Greedy2 algorithms. Indeed, the network cost at the last iteration of the IG algorithm is of 50906. This represents a gain of 14.70%. In addition, if the algorithms are stopped after 2700 sec of computation time, the network cost at this imposed ending time is of 51000, 58628, and 58924 for the IG, the Greedy1, and the Greedy2 approaches, respectively.

5.7.7 Impact of the SED Routing Solution on the Grooming Procedure

Let us consider a new set SxD_6 of 5000 SEDs (*c.f.* Section 3.6). This set is characterized by an average traffic flow of $\phi_{SED} = 1.52\ Tbps$ entering the network at each instant and a peak flow of $\pi_{SED} = 2.18\ Tbps$. The time correlation between the requests of this set is equal to 22.17%.

First, we assign to each request the shortest path between its source node and its destination node. When no grooming is performed, a network able to handle this set of 5000 SEDs is composed of 19811 optical ports and 8106 electrical ports (*c.f.* Table 5.15). Given that $\kappa = 5$, the overall cost of the network is equal to 60341. The congestion, defined as the number of wavelengths used on the most loaded link, is equal to 246.

Without changing the path assigned to the requests and by only applying our IG algorithm, the number of optical ports required to handle this set of SEDs is reduced to 14559 ports, while the number of electrical ports is reduced to 7308 ports. This result is achieved when the parameters L_1, L_2, and N_1 of our proposed iterative greedy algorithm are fixed to 100, 1000, and 1, respectively. As a result, the overall cost of the network has decreased to 51099 which represents a gain of 15.31%. The congestion has also decreased to 152. The processing time to obtain this result is roughly one hour.

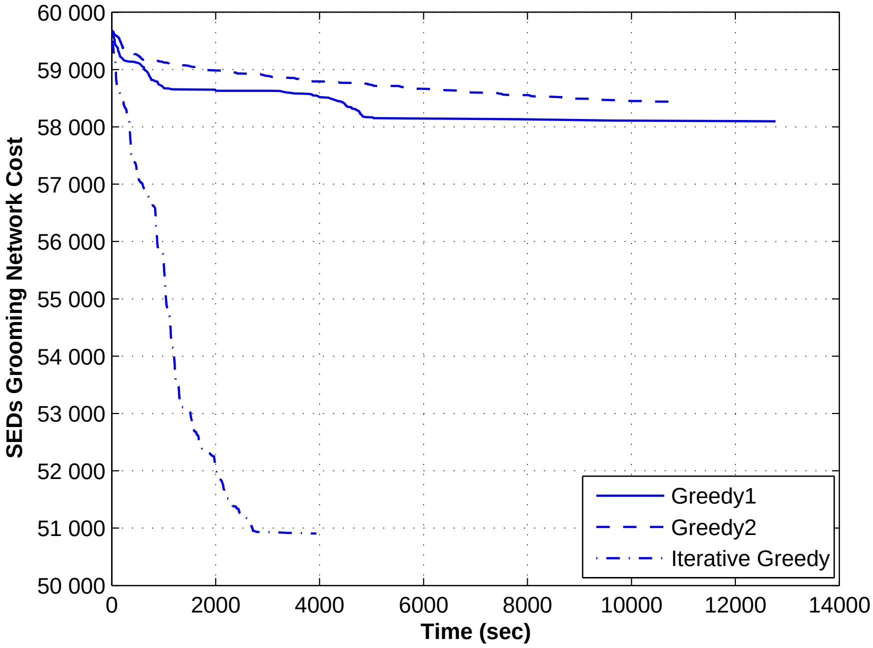


Fig. 5.20. The best solution over time obtained by means of the grooming process for three different algorithms: Greedy1, Greedy2, and Iterative Greedy

Finally, we evaluate the impact of different routing solutions on the network cost. For this task, we have randomly generated 50 different routing solutions. For each routing solution, we have applied our IG algorithm with the parameters L_1, L_2, and N_1 being fixed to 100, 1000, and 1, respectively. In average over these 50 routing solutions, a network able to handle this set of SEDs is composed of 14520 optical ports and 7323 electrical ports. As a result, the overall cost of the network is equal to 51135 which represents a gain of 15.25%. The cheapest network observed during our simulations is composed of 14123 optical ports and 7294 electrical ports corresponding to a global cost of 50593. Table 5.15 shows the resources required for the different routing solutions of the set SxD_6 of 5000 SEDs.

Table 5.15. Impact of the SED routing solution on the network cost

SEDs' Routing and Grooming Cost								
Shortest Path without Grooming	o_1-ports	15758	e_1-ports	2022	r_1-ports	2031	Optical ports	19811
	o_2-ports	-	e_2-ports	-	r_1-ports	-	Electrical ports	8106
	o_3-ports	4053	e_3-ports	2022	r_3-ports	2031	Congestion	246
Shortest Path with Grooming	o_1-ports	11304	e_1-ports	2022	r_1-ports	2031	Optical ports	14559
	o_2-ports	-	e_2-ports	-	r_2-ports	-	Electrical ports	7308
	o_3-ports	3255	e_3-ports	1626	r_3-ports	1629	Congestion	152
Average Cost with Grooming	o_1-ports	11250	e_1-ports	2022	r_1-ports	2031	Optical ports	14520
	o_2-ports	-	e_2-ports	-	r_2-ports	-	Electrical ports	7323
	o_3-ports	3270	e_3-ports	1632	r_3-ports	1638	Congestion	154
Best Cost with Grooming	o_1-ports	10882	e_1-ports	2022	r_1-ports	2031	Optical ports	14123
	o_2-ports	-	e_2-ports	-	r_2-ports	-	Electrical ports	7294
	o_3-ports	3241	e_3-ports	1621	r_3-ports	1620	Congestion	152

In average over the 50 routing solutions, the overall network cost varies from 60341 at the beginning of the simulation to about 51135 after one hour and a half of computation time. Figure 5.21 plots the network cost over the computation time. The continuous curve refers to the average network cost over time observed over the 50 routing solutions, while the vertical lines correspond to the mean square variation of the network cost over these routing solutions.

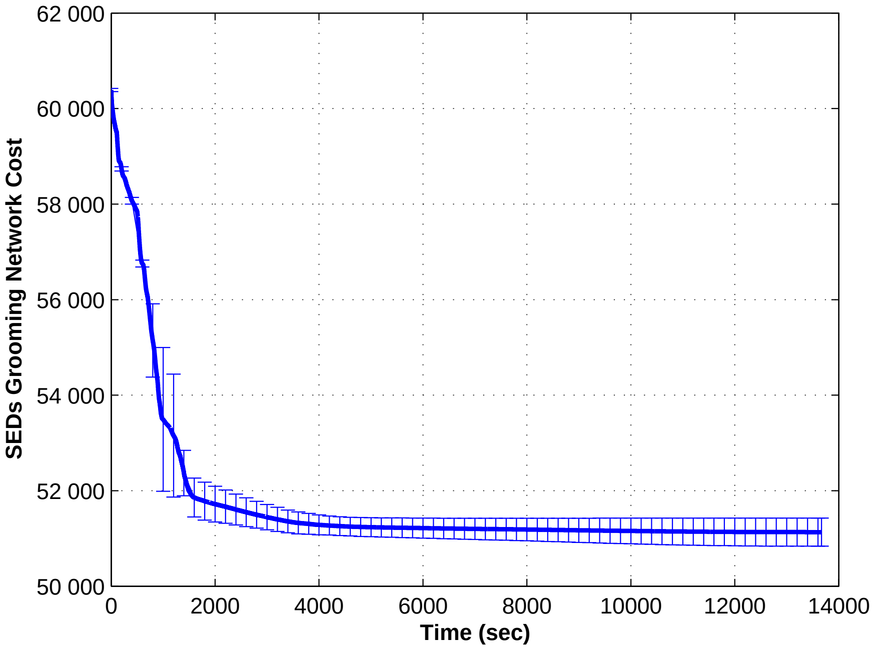


Fig. 5.21. The overall network cost over the computation time (set of 50 SED routing solutions)

5.7.8 Impact of the Size of the T_list (Parameters: L_1 and L_2) on the IG Algorithm

In this section, we evaluate the impact of the size of the T_list on the performance of our IG algorithm. We still make use of the set SxD_6 of 5000 SEDs (*c.f.* Section 3.6). Table 5.16 shows the number of required electrical ports and optical ports for different values of the parameters L_1 and L_2 at different time instances. Note that these results are obtained when the SEDs are routed according to the shortest path algorithm. Figure 5.22 plots the overall network cost over the computation time.

It is to be noted that, at the beginning of the simulation, the slope of the network cost is steeper for small values of L_1 than for large values. Thus, we can reach better network cost in shorter time. However, the network cost obtained at the end of the IG algorithm is smaller for large values of L_1 but it requires extensive computation time. Finally, the impact of the L_2 parameter becomes negligible for large values of the L_1 parameter. To conclude, ($L_1 = 100$, $L_2 = 1000$) seems a good trade-off between the size of the T_list, the computation time, and the cost obtained at the end of the grooming optimization.

5.7.9 Impact of the Number of Iterations (Parameter: N_1) on the IG Algorithm

We have stated that the **Step 1** of our proposed IG algorithm can be repeated several times. In this section, without changing the considered set SxD_6 of 5000 SEDs (*c.f.* Section 3.6), we evaluate the

Table 5.16. Impact of the parameters L_1 and L_2

		Step 1: after 500s	End of **Step 1**	**Step 2**: $L_2 = L_1$	**Step 2**: $L_2 = 1000$
$L_1 = 10$	Elapsed time	500	780	823	10508
	Optical ports	17715	16143	16051	14410
	Electrical ports	7836	7436	7414	7263
	Congestion	204	188	187	152
$L_1 = 50$	Elapsed time	500	1384	1426	5642
	Optical ports	18462	15433	15407	14573
	Electrical ports	7955	7386	7388	7310
	Congestion	208	169	169	152
$L_1 = 100$	Elapsed time	500	977	1009	3583
	Optical ports	18152	15413	15401	14559
	Electrical ports	7927	7378	7380	7308
	Congestion	199	169	169	152
$L_1 = 250$	Elapsed time	500	2826	2997	3274
	Optical ports	18407	14547	14511	14482
	Electrical ports	7950	7312	7312	7313
	Congestion	208	154	152	151
$L_1 = 500$	Elapsed time	500	5366	5689	6952
	Optical ports	18471	13509	13499	13476
	Electrical ports	7954	7254	7254	7253
	Congestion	208	146	146	146
$L_1 = 1000$	Elapsed time	500	7302	8325	8325
	Optical ports	18454	13223	13186	13186
	Electrical ports	7955	7242	7243	7243
	Congestion	208	143	143	143

impact of the number of times N_1 this step is repeated on the performance of the grooming algorithm. Table 5.17 shows the overall cost of the network for different values of the parameter N_1 as well as the time needed to reach this solution. Figure 5.23 plots the overall network cost over the computation time.

From these results, we can conclude that the performance of our algorithm increases as the number of iterations increases. In addition, for large N_1 values, the impact of the size L_2 of the *T_list* during the **Step 2** becomes negligible. However, $N_1 = 3$ seems a good trade-off between the computation time and the cost obtained at the end of the grooming optimization.

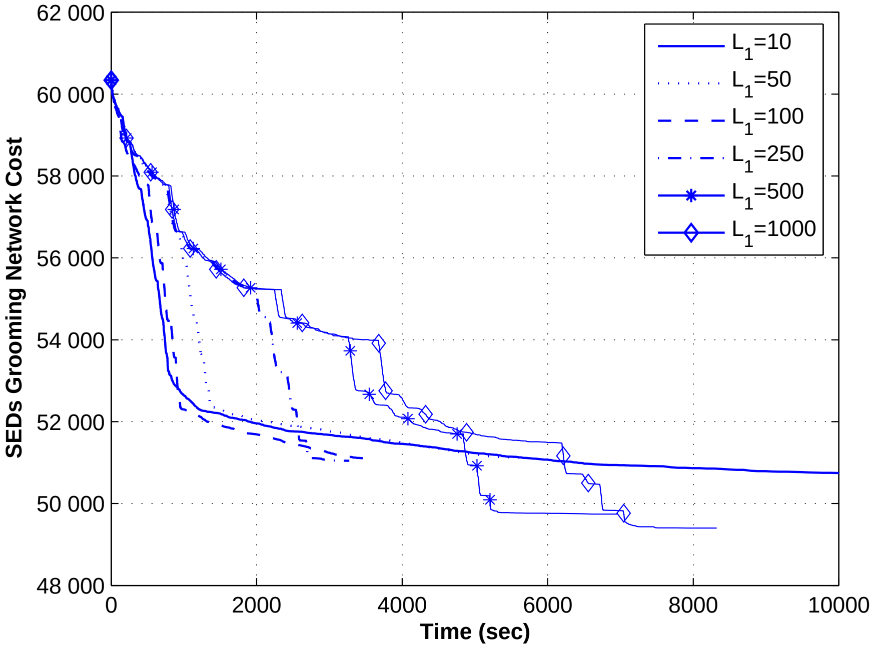


Fig. 5.22. The overall network cost over time for different values of L_1

Table 5.17. Impact of the parameter N_1

		$L_1 = 50,\ L_2 = 250$		$L_1 = 100,\ L_2 = 250$		$L_1 = 250,\ L_2 = 250$	
		Step 1	**Step 2**	**Step 1**	**Step 2**	**Step 1**	**Step 2**
$N_1 = 1$	Network Cost	52363	51147	52303	51113	51107	51071
	Elapsed Time	1385	5349	1547	5405	2860	3033
$N_1 = 2$	Network Cost	49993	49695	49629	49599	49159	49153
	Elapsed Time	2518	3574	2965	3159	4689	4808
$N_1 = 3$	Network Cost	49627	49591	48917	48899	48411	48399
	Elapsed Time	2776	3052	3745	3922	6104	6306
$N_1 = 4$	Network Cost	49285	49283	48559	48557	47951	47951
	Elapsed Time	2986	3090	4303	4420	7017	7138
$N_1 = 5$	Network Cost	48807	48803	48101	48099	47607	47607
	Elapsed Time	3563	3668	5134	5236	8048	8125

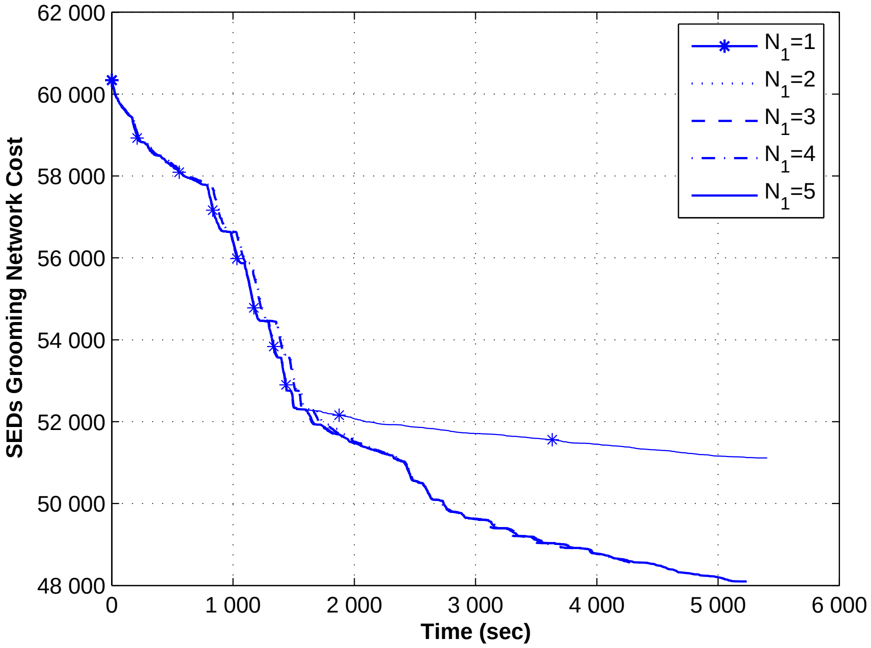


Fig. 5.23. The overall network cost over time for $L_1 = 100$, $L_2 = 250$ and different values of N_1

6

Routing and Grooming of Random Demands (sRxD/RxD)

Abstract: *After having described in Chapter 5 our approach for SxDs routing, this chapter is dedicated to the problem of routing random traffic demands. Two types of such demands are considered: pure random demands (RxDs) with unknown date of arrival and unknown life duration, and semi-random demands (sRxDs) with random date of arrival but pre-known life duration. In this chapter, we also propose grooming strategies for both sRxDs and RxDs with a granularity lower than the wavelength capacity. For this purpose, we define a dynamic weight assignment scheme for the various edges of the auxiliary graph presented in Chapter 4. We conclude this chapter by a set of numerical results outlining the benefits of our routing strategy.*

In the previous chapter, we have dimensioned a network that is able to carry all the SxDs at the lowest cost. Once the network is dimensioned, the SxDs (SLDs and SEDs) are routed by reserving their required resources during their active period. This is carried out by the management plane that informs each node of the availability of the various equipment (electrical ports and optical ports) within the network. At this stage, the network can be over-dimensioned by providing additional resources in order to route some random traffic demands. The control plane is in charge of routing the random demands that arrive one at a time at the network and need to be routed on the fly. The control plane operates in real time and is distributed in all the nodes of the network. We assume that GMPLS signaling, still under specification, is used to share information between the two planes.

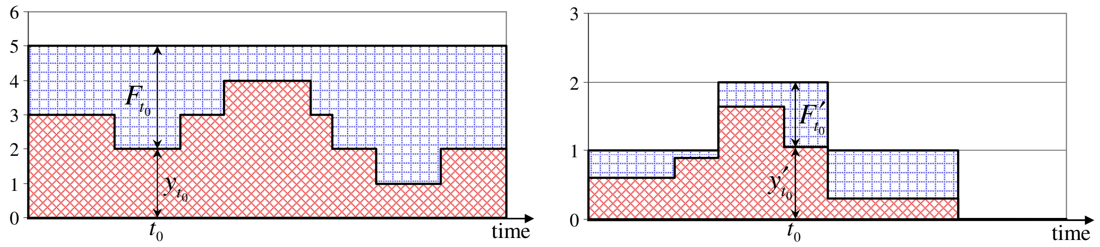
6.1 Network Over-dimensioning

Once the network is dimensioned, it can be over-dimensioned by providing additional resources in order to route random traffic demands. This resource provisioning consists in:

1. providing each node by $\ddot{\alpha}$ additional e_1-ports and $\ddot{\alpha}$ additional r_1-ports. These ports are used to add/drop REDs/sREDs.
2. providing each node by $\ddot{\beta}$ additional e_2-ports, $\ddot{\beta}$ additional r_2-ports, and $2 \times \ddot{\beta}$ additional o_2-ports. These ports are used to add/drop RLDs/sRLDs.
3. providing each node by $\ddot{\gamma}$ additional e_3-ports, $\ddot{\gamma}$ additional r_3-ports, and $2 \times \ddot{\gamma}$ additional o_3-ports. These ports connect the EXC to the OXC and are used to groom the REDs/sREDs together.

4. providing each node by additional o_1 optical ports according to an over-dimensioning factor $\ddot{\eta}$ applied to the already dimensioned network.

At the end of the over-dimensioning procedure and before dealing with random demands, we have a full knowledge of the state of the network at each instant in the future. The state of the network is defined by both the installed capacity of each network component and the amount of resources consumed by the SxDs at each instant. In other words, we know the number of electrical ports and the number of optical ports installed at each node. This determines the maximum capacity of a node to add/drop full-wavelength and sub-wavelength traffic requests as well as its switching and its grooming capacities. In addition, we know for each grooming lightpath (GL) its instant of establishment, its life duration, its exact route on the physical topology, and the load it carries at each instant. For instance, let us consider the full-wavelength add ports of a particular node. After dimensioning and over-dimensioning the network, let us assume that the number of emitting e_2 electrical ports is equal to $C = 5$. This represents the maximum capacity of the node to add full-wavelength xLD requests. According to the routes assigned to a given set of SLDs, the variation of the number of SLDs added at this node over time is represented by Figure 6.1(a). Let y_{t_0} be the number of xLDs added at this node at instant t_0. The difference $F_{t_0} = C - y_{t_0}$ represents the free capacity/number of the e_2 ports at instant t_0. Similarly, let us consider a given GL connecting two nodes along a particular path. According to the routes and the grooming schemes assigned to a given set of SEDs, the maximum capacity C'_{t_0} of the GL at instant t_0 and the traffic rate y'_{t_0} that this GL carries at this same instant are given by Figure 6.1(b). The difference $F'_{t_0} = C'_{t_0} - y'_{t_0}$ represents the free capacity of the GL at instant t_0. Let us restate, that the traffic load y'_{t_0} of a GL is a step function that takes its values in $\mathbb{R}^+$ due to the fact that the required rate of the xEDs is smaller than the capacity of an optical channel.



(a) Physical network resources: e_1, e_2, e_3, r_1, r_2, r_3, o_1, o_2, and o_3 ports

(b) Logical network resources: Grooming Lightpath

Fig. 6.1. Load and Free Capacity of the various resources in the Network

6.2 Traffic Engineering

In this chapter, we try to route a set of RxDs/sRxDs over the dimensioned/over-dimensioned network. Due to their random nature, the RxDs/sRxDs must be routed online. More specifically, the REDs/sREDs must also be aggregated on the fly. The current well-known routing approaches are fixed routing, fixed-alternate routing, and adaptive routing [J136].

In fixed routing, the connections are always routed through the same pre-defined fixed route for a given source-destination node pair. An example of such an approach is fixed shortest-path routing. The shortest-path route between each source-destination node pair is calculated off-line using standard shortest-path algorithms such as Dijkstra's algorithm [J137, W170] or Bellman-Ford's algorithm [J138, W170]. This approach to route connections is very simple; however, it can potentially lead to high blocking probabilities because the traffic is distributed to the links that belong to some shortest paths. Also, fixed routing may be unable to handle fault situations in which one or more links in the network fail. To handle link faults, the routing scheme must either consider alternate paths to the destination or be able to find a new route dynamically.

To alleviate the drawback of fixed shortest-path routing algorithm, Harai *et al.* proposed the fixed-alternate routing algorithm [C50] and investigated its performance by means of an analytical model. The fixed-alternate routing algorithm can improve the blocking performance by introducing multiple fixed routes between each pair of nodes. In this approach, each node in the network is required to maintain a routing table that contains an ordered list of a number of fixed routes to each destination node. For example, these routes may include the shortest-path, the second shortest-path, etc and are ordered by the number of hops or the physical length to the destination. When a new connection request arrives, if there is no free resources on the primary route, an alternative route is investigated. If no resource-free route is found from the list of alternate routes, then the connection request is blocked and lost. Fixed-alternate routing provides simplicity of control for setting-up and tearing-down lightpaths. It may also be used to provide some degree of fault tolerance upon link failures. Another advantage of fixed-alternate routing is that it can significantly reduce the connection blocking probability compared to fixed routing because the traffic is potentially distributed to more fiber-links. It has been shown that, for certain networks, having as few as two alternate routes provides significantly lower blocking probabilities than having full wavelength conversion at each node with fixed routing [D157].

The main problem of static routing strategies is that the route decision does not consider the current network state. In other terms, the static routing is not suited to traffic engineering which is very important in random traffic scenarios. On the contrary, adaptive routing algorithms are good candidates for traffic engineering, and they can further improve the blocking performance significantly [J139, J140]. In adaptive routing, the route decision is based on the current network state. The network state is determined by the set of all the connections that are currently in progress and can be managed in either a distributed manner or a centralized manner. For scalability, distributed management is often preferred [J141]. When a new connection request arrives, the current shortest path between the source node and the destination node is computed based on the available resources in the network (the current network state); then the connection is established through this route. Adaptive routing requires extensive support from the control protocol and the management protocol to continuously update the current network state. In addition, adaptive routing requires more computation and a longer set-up time than fixed and fixed-alternate routing. However, adaptive routing is more flexible and results in lower connection blocking probabilities.

In our approach, we make use of the adaptive routing strategy applied to the auxiliary graph model derived from [C11, J62, B150] and already introduced in Section 4.4. This graph model allows for different number of electrical ports and optical ports at each node, different number of wavelengths on each fiber-link, as well as different wavelength conversion capabilities and different traffic grooming capabilities at each node. One associates to each edge in this auxiliary graph a property tuple $P(C_t, y_t)$, where C_t denotes the capacity limiting the use of this edge at instant t and y_t denotes the load of this edge at the same instant t. This property tuple $P(C_t, y_t)$ applied to all the edges of the network auxiliary graph reflects the network state at the time instant t. Based on these information, a weight is appropriately computed and assigned to each edge. The capability of easily adjusting the edges' weight makes the graph model very suitable for dynamic traffic grooming. By means of the well-known Dijkstra algorithm combined with an appropriate dynamic cost assignment to each edge of the equivalent auxiliary graph, we can simultaneously and in real time determine the logical topology that consists of lightpaths, route the lightpaths over the physical topology, and finally route the traffic on the virtual topology.

Example 6.1.

In order to solve the dynamic routing and traffic grooming problem, we first construct an auxiliary graph representation of each node in the network. Let us review the basic concepts of this graph representation largely detailed in Section 4.4. An auxiliary graph is a three-layer graph. The top-most layer is the access layer where traffic flows originate and terminate. The middle layer is the lightpath layer where low-speed traffic streams are multiplexed (demultiplexed) onto (from) the grooming lightpaths. The lower layer is the wavelength layer and represents the wavelength switching capability of the node. Each layer is represented by two vertices; one acting as an entry point to the layer and the other acting as an exit point from the layer. These vertices are connected by a set of inter-node and intra-node edges. In figure 6.2, we redraw the auxiliary graph representation corresponding to a network node with full wavelength conversion capability already illustrated in Figure 4.5(a). Let us recall the characteristics of the various edges:

Type *'a'* edges: An ***'a'*** edge is used to add a sub-wavelength channel request (xED) at the node. Its capacity is limited by the number of emitting e_1-ports present at this node.

Type *'b'* edges: A ***'b'*** edge represents the ingress point of a GL. Its capacity is limited by the number of emitting e_3-ports/receiving o_3-ports present at this node.

Type *'c'* edges: A ***'c'*** edge is used to add a full-wavelength channel request (xLD) at the node. Its capacity is limited by the number of emitting e_2-ports/receiving o_2-ports present at this node.

Type *'d'* edges: A ***'d'*** edge is used to drop a sub-wavelength channel request (xED) at the node. Its capacity is limited by the number of receiving r_1-ports present at this node.

Type *'e'* edges: An ***'e'*** edge represents the egress point of a GL. Its capacity is limited by the number of receiving r_3-ports/emitting o_3-ports ports present at this node.

Type *'f'* edges: An ***'f'*** edge is used to drop a full-wavelength channel request (xLD) at the node. Its capacity is limited by the number of receiving r_2-ports/emitting o_2-ports ports present at this node.

Type *'g'* edges: A ***'g'*** edge represents the grooming capability of the node.

Type *'h'* edges: An ***'h'*** edge represents the wavelength switching capability of the node.

Type *'i'* edges: An ***'i'*** edge represents a pre-established GL connecting two nodes. The capacity of this edge is equal to the number of wavelengths required to groom the requests carried by its corresponding GL. It is to be noted that this capacity can vary in time as shown in Figure 6.1(b).

Type '*o*' edges: An '*o*' edge represents a physical fiber-link connecting two nodes. The capacity of this edge is equal to the number of o_1-ports present at each end of the physical link.

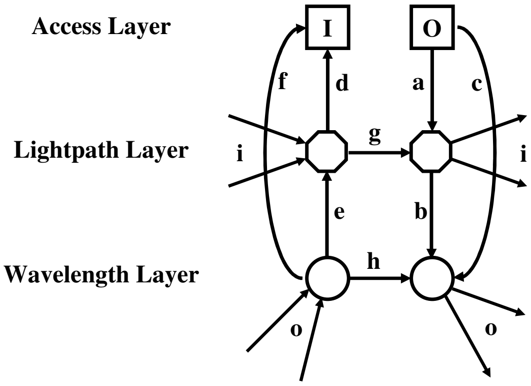


Fig. 6.2. Auxiliary graph representation of a network node

Based on the auxiliary graph model of a node, we construct the auxiliary graph of the whole network according to the given network configuration. For simplicity and without loss of generality, we consider a very simple network topology composed of three nodes connected by four unidirectional fiber-links. We suppose that a grooming lightpath is already established between node A and node C in the logical topology. The auxiliary graph of the network is constructed as in Figure 6.3.

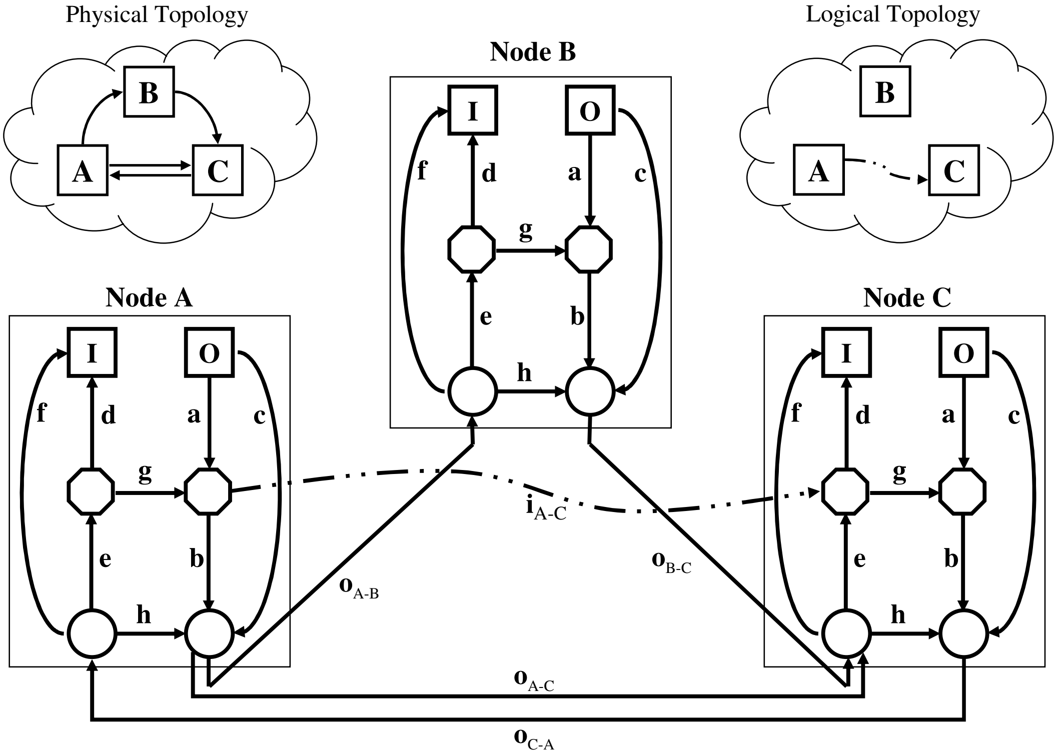


Fig. 6.3. Auxiliary graph representation of a simple network topology

6.3 Weight Assignment in the Case of sRLDs

Let us consider a new incoming sRLD connection request defined by $\delta_i^L\ \big(s_i, d_i, \alpha_i, \beta_i, n_i = 1.0\big)$. The required rate n_i of this connection is normalized with respect to the capacity C_ω of an optical channel.

6.3.1 Weight Assigned to *'a'* and *'d'* edges

Used respectively to add and to drop sub-wavelength connection requests (xEDs), the ***'a'*** and ***'d'*** edges are assigned infinite weights when routing sRLDs.

6.3.2 Weight Assigned to *'b'* and *'e'* edges

The sRLDs do not need any grooming functionality. Thus, when routing sRLDs, the ***'b'*** and ***'e'*** edges, that are used respectively to initiate and to terminate grooming lightpaths, are assigned infinite weights.

6.3.3 Weight Assigned to *'c'* and *'f'* edges

The ***'c'*** edges are used to add full-wavelength connection requests (xLDs), while the ***'f'*** edges are used to drop these connection requests. The capacity C of a ***'c'*** edge, respectively of an ***'f'*** edge is given by the number of e_2-ports, respectively r_2-ports installed at the node. The current load y_t of these edges is the number of xLDs added, respectively dropped at this node at time instant t. y_t is a function of already established sRLDs as well as of past and future SLDs. Let $Y_{[\alpha_i,\beta_i]}$ be the maximum number of ports used during the life duration of the request $\delta_i^L\ \big(Y_{[\alpha_i,\beta_i]} = \max\limits_{\alpha_i \leqslant t < \beta_i} y_t\big)$. $Y_{[\alpha_i,\beta_i]}$ is computable since the life duration $(\beta_i - \alpha_i)$ of the incoming sRLD is known at its instant of arrival α_i. The difference $F_{[\alpha_i,\beta_i]} = C - Y_{[\alpha_i,\beta_i]}$ represents the number of e_2/r_2 ports that remain free during the life duration of the request. The weight $w_{c,f}$ assigned to these edges is given by:

$$w_{c,f} = \begin{cases} 0 & \text{if } F_{[\alpha_i,\beta_i]} > 0, \\ \infty & \text{if } F_{[\alpha_i,\beta_i]} = 0. \end{cases} \tag{6.1}$$

6.3.4 Weight Assigned to *'g'* and *'h'* edges

The ***'g'*** edges represent the grooming and sub-wavelength switching functions of the nodes. As the sRLDs do not undergo any grooming process, the ***'g'*** edges are assigned infinite weights. On the other hand, the ***'h'*** edges represent the wavelength switching functionality of the nodes. Assuming non-blocking OXCs, the ***'h'*** edges are assigned null weights.

6.3.5 Weight Assigned to *'o'* edges

The ***'o'*** edges represent the physical topology of the network. The maximum capacity C of an ***'o'*** edge is the number of wavelengths in the fiber-link or equivalently, the number of o_1-ports installed at each end of the fiber-link. The current load y_t of an ***'o'*** edge is the number of active wavelengths on its corresponding fiber-link. y_t is a function of already established sRLDs/sREDs as well as of past and future SLDs/SEDs. Let $Y_{[\alpha_i,\beta_i]}$ be the maximum number of optical channels used on this fiber-link during the life duration of the request $\delta_i^L\ \big(Y_{[\alpha_i,\beta_i]} = \max\limits_{\alpha_i \leqslant t < \beta_i} y_t\big)$. $Y_{[\alpha_i,\beta_i]}$ is computable since the life duration $(\beta_i - \alpha_i)$ of the incoming sRLD is known at its instant of arrival α_i. The difference $F_{[\alpha_i,\beta_i]} = C - Y_{[\alpha_i,\beta_i]}$ represents the number of optical channels that remain free during the life duration of the request. The weight w_o assigned to an ***'o'*** edge is then given by:

$$w_o = \begin{cases} C/F_{[\alpha_i,\beta_i]} & \text{if } F_{[\alpha_i,\beta_i]} > 0, \\ \infty & \text{if } F_{[\alpha_i,\beta_i]} = 0. \end{cases} \tag{6.2}$$

Such a weight assignment evenly distributes the traffic load across the network and guarantees that the most loaded links are avoided as much as possible. Indeed, the higher the utilization of a fiber-link (represented by the value of $Y_{[\alpha_i,\beta_i]}$), the higher its assigned weight. This approach is known in the literature as "Congestion Minimization".

6.3.6 Weight Assigned to *'i'* edges

The ***'i'*** edges represent the logical topology of the network. An ***'i'*** edge corresponds to an already established GL. As the sRLDs are routed directly on the physical topology, infinite weights are assigned to the ***'i'*** edges when routing an sRLD.

Example 6.2.

Let us consider the simple 3-node and 4-link network topology already introduced in Example 6.1 and shown in Figure 6.3. A new incoming sRLD connection request $\delta_i^E\left(s_i = A, d_i = C, \alpha_i = 3:00, \beta_i = 19:00, n_i = 1.0\right)$ arrives at the network and needs to be routed on the fly. All the ***'a'***, ***'b'***, ***'d'***, ***'e'***, ***'g'***, and ***'i'*** edges in the network auxiliary graph are assigned infinite weights,

$$w_a = w_b = w_d = w_e = w_g = w_i = \infty \tag{6.3}$$

while all the ***'h'*** edges are assigned null weights.

$$w_h = 0 \tag{6.4}$$

For instance, we assume that 5 emitting e_2 electrical ports are already installed at node A, and that the utilization of these ports varies in time as illustrated in Figure 6.4.

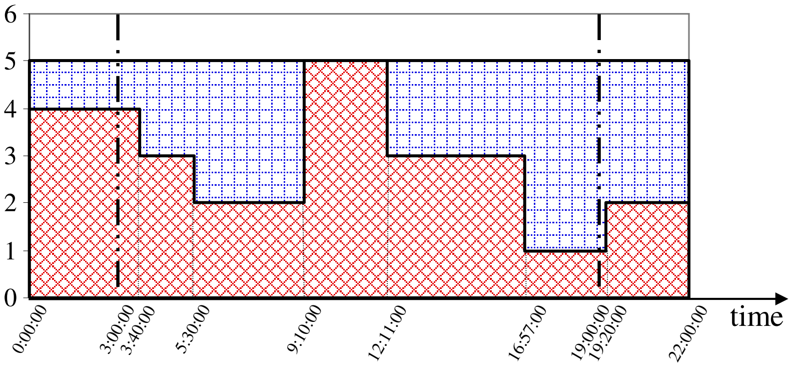


Fig. 6.4. Utilization of the e_2-ports at node A

During the life duration $\Theta_T = [3:00, 19:00]$ of the new sRLD, the maximum load $Y_{[3:00,19:00]}$ of the ***'c'*** edge at node A is evaluated to 5. The maximum capacity C of this edge being also equal to 5 (number of installed ports), its remaining free capacity is equal to $F_{[3:00,19:00]} = C - Y_{[3:00,19:00]} = 5 - 5 = 0$. As a result, we assign to this edge a weight w_c^A equal to:

$$w_c^A = \infty \tag{6.5}$$

Similar weight assignments can be applied to the remaining ***'c'*** and ***'f'*** edges of the network.

In addition, we assume that the number of WDM channels per fiber-link is set to 5. Figure 6.5 gives the WDM channels utilization over time of the fiber-link connecting the node A to the node C.

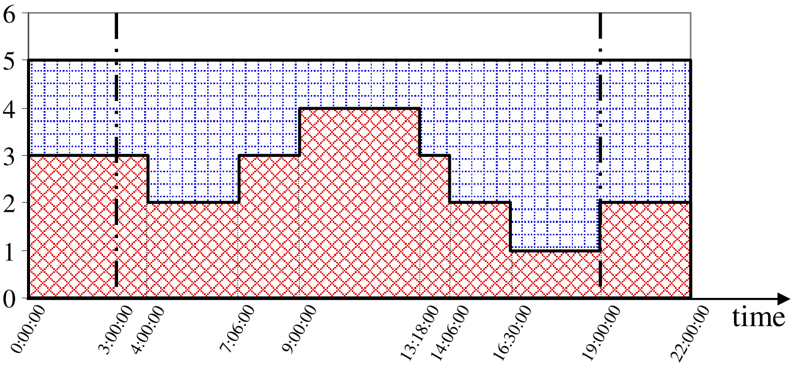


Fig. 6.5. WDM channel utilization on the fiber-link connecting node A and node C

During the life duration $\Theta_T = [3:00, 19:00]$ of the sRLD, the maximum load $Y_{[3:00,19:00]}$ of the ***'o'*** edge representing this fiber-link is evaluated to 4. The maximum capacity C of this edge being equal to 5 (number of WDM channels), its remaining free capacity is equal to $F_{[3:00,19:00]} = C - Y_{[3:00,19:00]} = 5 - 4 = 1$. As a result, we assign to this edge a weight w_o^{A-C} equal to:

$$w_o^{A-C} = 5/1 = 5 \tag{6.6}$$

Similar weight assignments can be applied to the remaining ***'o'*** edges of the network.

6.4 Weight Assignment in the Case of sREDs

Let us consider a new incoming sRED connection request defined by $\delta_i^E\left(s_i, d_i, \alpha_i, \beta_i, n_i < 1.0\right)$. The required rate n_i of this connection is normalized with respect to the capacity C_ω of an optical channel.

6.4.1 Weight Assigned to *'a'* and *'d'* edges

The ***'a'*** edges are used to add sub-wavelength connection requests (xEDs), while the ***'d'*** edges are used to drop these connection requests. The capacity C of an ***'a'*** edge, respectively a ***'d'*** edge is given by the number of e_1-ports, respectively r_1-ports installed at the node. The current load y_t of these edges is the number of xEDs added, respectively dropped at this node at time instant t. y_t is a function of already established sREDs as well as of past and future SEDs. Let $Y_{[\alpha_i,\beta_i]}$ be the maximum number of ports used during the life duration of the request δ_i^E $\left(Y_{[\alpha_i,\beta_i]} = \max\limits_{\alpha_i \leqslant t < \beta_i} y_t\right)$.

$Y_{[\alpha_i,\beta_i]}$ is computable since the life duration $(\beta_i - \alpha_i)$ of the incoming sRED is known at its instant of arrival α_i. The difference $F_{[\alpha_i,\beta_i]} = C - Y_{[\alpha_i,\beta_i]}$ represents the number of e_1/r_1 ports that remain free during the life duration of the request. The weight $w_{a,d}$ assigned to these edges is given by:

$$w_{a,d} = \begin{cases} 0 & \text{if } F_{[\alpha_i,\beta_i]} > 0, \\ \infty & \text{if } F_{[\alpha_i,\beta_i]} = 0. \end{cases} \tag{6.7}$$

6.4.2 Weight Assigned to *'b'* and *'e'* edges

The ***'b'*** edges are used to initiate new grooming lightpaths (GLs), while the ***'e'*** edges are used to terminate these GLs. The capacity C of a ***'b'*** edge, respectively an ***'e'*** edge is given by the number of e_3-ports, respectively r_3-ports installed at the node. The current load y_t of these edges is the number of GLs initiated, respectively terminated at this node at time instant t. y_t is a function of already established sREDs as well as of past and future SEDs. Let $Y_{[\alpha_i,\beta_i]}$ be the maximum number of ports used during the life duration of the request δ_i^E $\left(Y_{[\alpha_i,\beta_i]} = \max\limits_{\alpha_i \leqslant t < \beta_i} y_t\right)$. $Y_{[\alpha_i,\beta_i]}$ is computable since the life duration $(\beta_i - \alpha_i)$ of the incoming sRED is known at its instant of arrival α_i. The difference $F_{[\alpha_i,\beta_i]} = C - Y_{[\alpha_i,\beta_i]}$ represents the number of e_3/r_3 ports that remain free during the life duration of the request. The weight $w_{b,e}$ assigned to these edges is given by:

$$w_{b,e} = \begin{cases} 0 & \text{if } F_{[\alpha_i,\beta_i]} > 0, \\ \infty & \text{if } F_{[\alpha_i,\beta_i]} = 0. \end{cases} \tag{6.8}$$

6.4.3 Weight Assigned to *'c'* and *'f'* edges

Used respectively to add and to drop full-wavelength connection requests (xLDs), the ***'c'*** and ***'f'*** edges are assigned infinite weights when routing sREDs.

6.4.4 Weight Assigned to *'g'* and *'h'* edges

The ***'g'*** edges represent the grooming and sub-wavelength switching functions of the nodes. Assuming non-blocking EXCs, the ***'g'*** edges are assigned null weights. Similarly, the ***'h'*** edges represent the wavelength switching functionality of the nodes. Assuming non-blocking OXCs, the ***'h'*** edges are assigned null weights.

6.4.5 Weight Assigned to *'o'* edges

The ***'o'*** edges represent the physical topology of the network. The maximum capacity C of an ***'o'*** edge is the number of wavelengths in the fiber-link or equivalently, the number of o_1-ports installed at each end of the fiber-link. The current load y_t of an ***'o'*** edge is the number of active wavelengths on its corresponding fiber-link. y_t is a function of already established sRLDs/sREDs as well as of past and future SLDs/SEDs. Let $Y_{[\alpha_i,\beta_i]}$ be the maximum number of optical channels used on this fiber-link during the life duration of the request δ_i^E $\left(Y_{[\alpha_i,\beta_i]} = \max\limits_{\alpha_i \leqslant t < \beta_i} y_t\right)$. $Y_{[\alpha_i,\beta_i]}$ is computable since the life duration $(\beta_i - \alpha_i)$ of the incoming sRED is known at its instant of arrival α_i. The difference $F_{[\alpha_i,\beta_i]} = C - Y_{[\alpha_i,\beta_i]}$ represents the number of optical channels that remain free during the life duration of the request. The weight w_o assigned to an ***'o'*** edge is given by:

$$w_o = \begin{cases} C/F_{[\alpha_i,\beta_i]} + (1 - n_i) & \text{if } F_{[\alpha_i,\beta_i]} > 0, \\ \infty & \text{if } F_{[\alpha_i,\beta_i]} = 0. \end{cases} \tag{6.9}$$

where "1" is the normalized capacity of the WDM channel and n_i is the normalized required capacity of the incoming sRED δ_i^E. As for the sRLDs, this weight assignment guarantees that the most loaded links are avoided as much as possible. Indeed, the higher the utilization of a fiber-link (represented by the value of $Y_{[\alpha_i,\beta_i]}$), the higher its assigned weight. The additional weight $(1 - n_i)$ represents the fact that the sRED does not use the whole capacity of the optical channel. In this way, we privilege the use of already established GLs before creating new GLs using new WDM channels.

6.4.6 Weight Assigned to *'i'* edges

An ***'i'*** edge represents an already established GL in the logical topology. Let us recall that a GL is a direct connection between two nodes acting as a logical one-hop link, where all intermediate nodes are passed through at the OXC level. To this edge corresponds a path composed of:

1. a ***'b'*** edge (representing an e_3 port and an o_3 port) at its ingress node,
2. an ***'e'*** edge (representing an o_3 port and an r_3 port) at its egress node,
3. a series of consecutive ***'o'*** and ***'h'*** edges connecting its ingress node to its egress node and representing its physical path. Let k be the number of WDM channels (hops/ ***'o'*** edges) that the GL passes through.

For a given GL, we evaluate its capacity C'_t and its load y'_t observed at time instant t. The former (C'_t) stands for the number of wavelengths required on each of the ***'o'*** edges along the physical path of the GL to carry a set of sub-wavelength requests (xEDs) at time instant t. The latter (y'_t) is the sum of the xEDs' data rates carried by the GL at the same instant t. y'_t is normalized with respect to the capacity C_ω of an optical channel. Both C'_t and y'_t are functions of already established sREDs as well as of past and future SEDs. Let F'_t be the difference $C'_t - y'_t$. F'_t represents the free capacity of the GL at time instant t. Its minimum value evaluated during a time interval Θ $\left(\min_{t\in\Theta} F'_t\right)$ is denoted by $F'_{[\Theta]}$ and represents the highest rate that can be carried by the GL during this period Θ.

According to the relative value of the GL's free capacity F'_t compared to the sRED's required rate n_i, the active period $\Theta_T = [\alpha_i, \beta_i]$ of the sRED is divided into two disjoint time intervals Θ_H and Θ_E ($\Theta_T = \Theta_H \cup \Theta_E$ and $\Theta_H \cap \Theta_E = \emptyset$). The former ($\Theta_H \subset [\alpha_i, \beta_i]$) is the period of time where the GL has enough free capacity to carry the incoming sRED ($F'_t \geqslant n_i,\ \forall t \in \Theta_H$). The latter ($\Theta_E = [\alpha_i, \beta_i] - \Theta_H$) is the period of time where the free capacity offered by the GL is smaller than the required rate of the incoming sRED ($F'_t < n_i,\ \forall t \in \Theta_E$). We propose to extend the free capacity of the GL by reserving (if possible) additional resources during Θ_E in order to satisfy the incoming sRED during its whole active period $[\alpha_i, \beta_i]$. Let w_H be the weight for holding the new request during Θ_H and w_E be the weight for extending the GL for Θ_E. The weight w_i assigned to the ***'i'*** edge representing this GL is equal to:

$$w_i = \frac{\overline{\Theta}_H}{\overline{\Theta}_T} \times w_H + \frac{\overline{\Theta}_E}{\overline{\Theta}_T} \times w_E \tag{6.10}$$

where $\overline{\Theta}_H$, $\overline{\Theta}_E$, and $\overline{\Theta}_T$ are the length of the time intervals Θ_H, Θ_E, and Θ_T, respectively. Note that the life duration of the incoming sRED is equal to $\overline{\Theta}_T = \beta_i - \alpha_i = \overline{\Theta}_H + \overline{\Theta}_E$. When the GL has enough free capacity during the whole period $[\alpha_i, \beta_i]$, we have $\Theta_E = 0$, and the weight w_i assigned to the ***'i'*** edge is equal to the weight w_H for holding the request during $\Theta_H = \Theta_T$. Conversely, when the GL lacks of free capacity during the whole period $[\alpha_i, \beta_i]$, we have $\Theta_H = 0$, and the weight w_i assigned to the ***'i'*** edge is equal to the weight w_E for extending the GL for the period $\Theta_E = \Theta_T$. In this latter case, extending a GL for the whole life duration of a request is equivalent to create a new GL during this period.

Extending a GL for the period Θ_E can be viewed as creating a new GL along a given path during this period. Thus, the weight w_E of such an extension is equal to the sum of the weights of the various edges along the path of the GL. In other words, w_E is equal to the sum of:

1. the weight w_b of the ***'b'*** edge at its ingress node (*c.f.* Section 6.4.2),
2. the weight w_e of the ***'e'*** edge at its egress node (*c.f.* Section 6.4.2),
3. the weight of the various ***'o'*** edges along its physical path (*c.f.* Section 6.4.5)(***'h'*** edges being assigned null weights (*c.f.* Section 6.4.4)).

For instance, let us consider a GL connecting node m to node n and spanning over the ***'b**$_m$**'***, ***'o**$_{l_1}$**'***, ***'o**$_{l_2}$**'***, $\cdots$, ***'o**$_{l_k}$**'***, and ***'e**$_n$**'*** edges. The weight w_E for extending this GL for Θ_E is computed as follows:

$$w_E = w_{b_m} + \sum_{i=1}^{k} w_{o_{l_i}} + w_{e_n} \tag{6.11}$$

where the free capacity $F_{[\Theta_E]}$ of the ***'b**$_m$**'***, ***'o**$_{l_i}$**'***, and ***'e**$_n$**'*** edges and consequently their assigned weight are evaluated during the period Θ_E and not during the life duration Θ_T of the incoming sRED.

By analogy, the weight w_H for holding an sRED during a period Θ_H is computed using reduced weights for the various edges along the path of the GL. The reduced weights of these edges are defined as follows:

1. the weight w_b of the ***'b'*** edge at the ingress node of the GL is reduced to 0,
2. the weight w_e of the ***'e'*** edge at the egress node of the GL is reduced to 0,
3. the ratio $C/F_{[\Theta_H]}$ in the weight assigned to an ***'o'*** edge (*c.f.* Equation (6.9)) is reduced to its minimum value 1,
4. the incremental weight $1 - n_i$ in the weight assigned to an ***'o'*** edge (*c.f.* Equation (6.9)) is reduced to $F'_{[\Theta_H]} - n_i$, where $F'_{[\Theta_H]}$ $\left(F'_{[\Theta_H]} < 1\right)$ is the free capacity of the GL evaluated during the period Θ_H.

For the previous GL connecting node m to node n along the path represented by the ***'b**$_m$**'***, ***'o**$_{l_1}$**'***, ***'o**$_{l_2}$**'***, $\cdots$, ***'o**$_{l_k}$**'***, and ***'e**$_n$**'*** edges in the network auxiliary graph, the corresponding weight w_H is expressed as follows:

$$\begin{aligned} w_H &= w_{b_m} + \sum_{i=1}^{k} w_{o_{l_i}} + w_{e_n} \\ &= k \times (1 + F'_{[\Theta_H]} - n_i) \end{aligned} \tag{6.12}$$

where k is the number of hops of this GL.

It is to be noticed that w_H is smaller than w_E. Hence, the weight assigned to a GL that can carry an incoming sRED request during its whole life period Θ_T is smaller than the weight of a GL that can partially hold this sRED during Θ_H and needs to be extended for Θ_E. In addition, this last weight is, in its turn, smaller than the weight of creating an entirely new GL for the incoming sRED. In this way, we try to pack an incoming sRED to the GL that suitably fits its needs.

Example 6.3.

Let us consider again the simple 3-node and 4-link network topology already introduced in Example 6.1 and shown in Figure 6.3. We assume that the number of WDM channels on each fiber-link is set to 5. A new incoming sRED connection request $\delta_i^E\left(s_i = A, d_i = C, \alpha_i = 3:00, \beta_i = 19:00, n_i = 0.25 < 1.0\right)$ arrives at the network and needs to be routed and groomed on the fly. All the ***'c'*** and ***'f'*** edges in the network auxiliary graph are assigned infinite weights.

$$w_c = w_f = \infty \tag{6.13}$$

Similarly, all the ***'h'*** and ***'g'*** edges in the network auxiliary graph are assigned null weights.

$$w_h = w_g = 0 \tag{6.14}$$

In addition, we assume that the weights assigned to the ***'a'***, ***'b'***, ***'d'***, and ***'e'*** edges are computed using the same weight computation method as described in Example 6.2 for the ***'c'*** edge of node A. In our example and without loss of generality, all the ***'a'***, ***'b'***, ***'d'***, and ***'e'*** edges in the network auxiliary graph are assigned null weights.

$$w_a = w_b = w_d = w_e = 0 \tag{6.15}$$

If the WDM channels utilization over time of the fiber-link connecting the node A to the node B is given by Figure 6.6, the maximum load $Y_{[3:00,19:00]}$ of its corresponding ***'o'*** edge during $\Theta_T = [3:00, 19:00]$ is equal to 4. Thus, its remaining free capacity is equal to $F_{[3:00,19:00]} = C - Y_{[3:00,19:00]} = 5 - 4 = 1$. As a result, we assign to this edge a weight w_o^{A-B}:

$$w_o^{A-B} = \frac{5}{1} + (1 - 0.25) = 5.75 \tag{6.16}$$

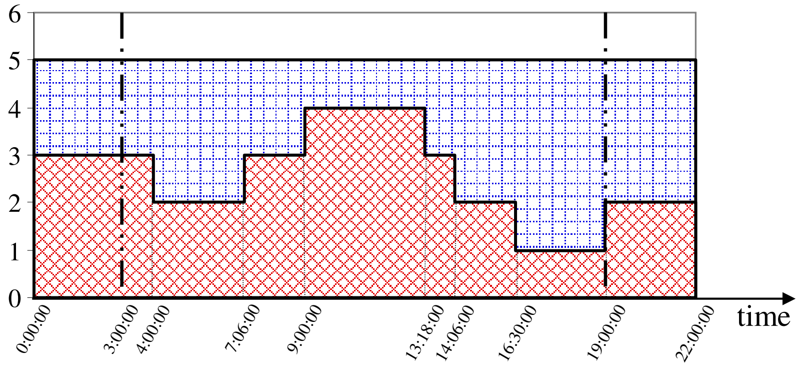


Fig. 6.6. WDM channel utilization on the fiber-link connecting node A and node B

Similarly, if the WDM channels utilization over time of the fiber-link connecting the node A to the node C is given by Figure 6.7, the maximum load $Y_{[3:00,19:00]}$ of its corresponding ***'o'*** edge during $\Theta_T = [3:00, 19:00]$ is equal to 5. Thus, its remaining free capacity is equal to $F_{[3:00,19:00]} = C - Y_{[3:00,19:00]} = 5 - 5 = 0$. As a result, we assign to this edge a weight w_o^{A-C}:

$$w_o^{A-C} = \infty \tag{6.17}$$

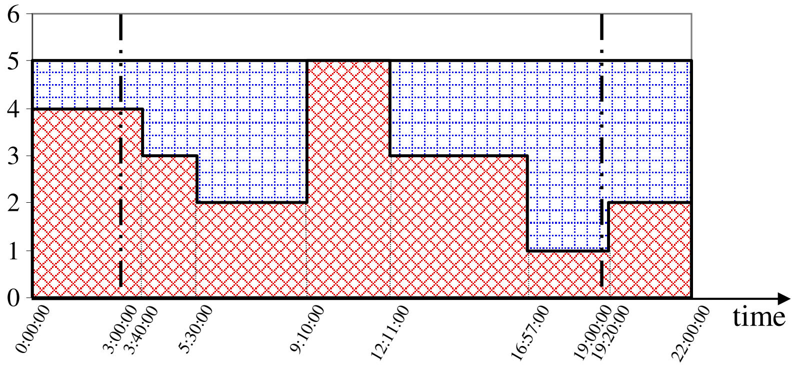


Fig. 6.7. WDM channel utilization on the fiber-link connecting node A and node C

Likewise, if the WDM channels utilization over time of the fiber-link connecting the node B to the node C is given by Figure 6.8, the maximum load $Y_{[3:00,19:00]}$ of its corresponding ***'o'*** edge during $\Theta_T = [3:00, 19:00]$ is equal to 5. Thus, its remaining free capacity is equal to $F_{[3:00,19:00]} = C - Y_{[3:00,19:00]} = 5 - 5 = 0$. As a result, we assign to this edge a weight w_o^{B-C}:

$$w_o^{B-C} = \infty \tag{6.18}$$

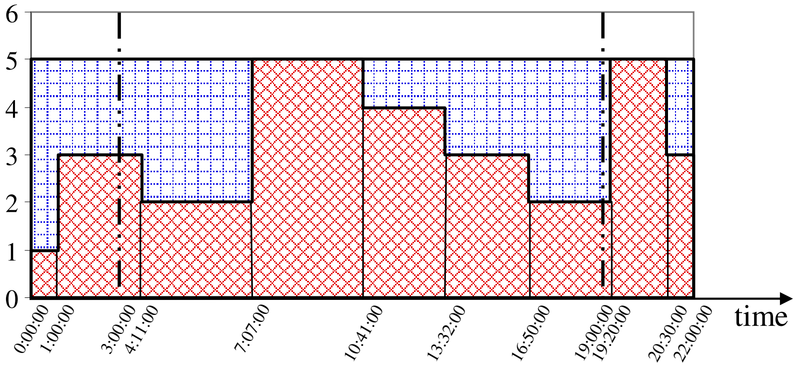


Fig. 6.8. WDM channel utilization on the fiber-link connecting node B and node C

One can notice that there is not any path with a finite weight that connects the node A to the node C using only ***'o'*** edges. Consequently, the network, in its current state, cannot satisfy the incoming sRED request by setting-up a new grooming lightpath directly connecting its source node to its destination node.

Now, let us consider the grooming lightpath connecting the node A to the node C. This GL is two hops long since it is routed along the path $A - B - C$. Let us assume that the maximum capacity C'_t and the carried data rate y'_t of this GL over time are illustrated in Figure 6.9. The free capacity $F'_{[3:00,19:00]}$ of its corresponding ***'i'*** edge evaluated during the period $\Theta_T = [3:00, 19:00]$ is equal to 0 (*c.f.* $C'_t - y'_t$ for $t \in [17:00, 19:00]$). Consequently, this existing GL, in its current state, cannot satisfy the incoming sRED request.

We divide the life duration $\Theta_T = [3:00, 19:00]$ of the sRED into two subintervals: Θ_H and Θ_E. The former subinterval corresponds to time instants where the GL has enough free capacity to satisfy the incoming request. Hence, $\Theta_H = [3:00, 4:00] \cup [6:00, 17:00]$. The latter subinterval corresponds to time instants where the free capacity offered by the GL is smaller than the required rate of the incoming sRED ($n_i = 0.25$). Hence, $\Theta_E = [4:00, 6:00] \cup [17:00, 19:00]$.

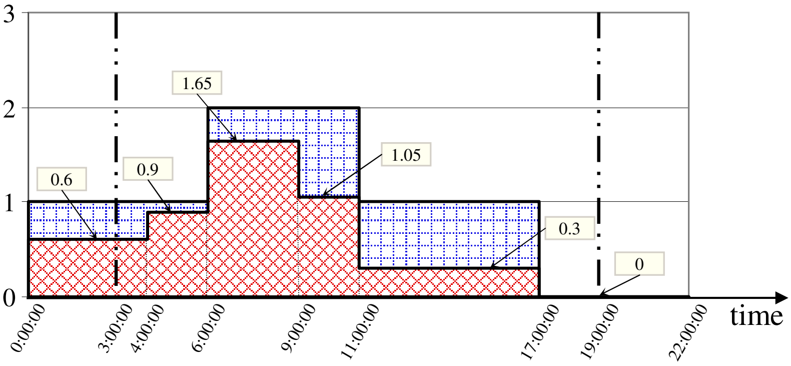


Fig. 6.9. Maximum capacity C'_t and traffic load y'_t of the GL connecting node A to node C

During Θ_H, the free capacity $F'_{[\Theta_H]}$ of this GL is equal to 0.35 (*c.f.* $C'_t - y'_t$ for $t \in [6:00, 9:00]$). As a result, the weight w_H for holding the new request during Θ_H is computed as follows:

$$\begin{aligned} w_H &= k \times (1 + F'_{[\Theta_H]} - n_i) \\ &= 2 \times (1 + 0.35 - 0.25) = 2.2 \end{aligned} \tag{6.19}$$

During Θ_E, additional resources must be reserved (if possible) along the path $A-B-C$ in order to extend the free capacity and/or the active duration of the GL. The free capacity $F_{[\Theta_E]}$ of the ***'o'*** edge along the fiber-link $A-B$ during Θ_E is 3 (*c.f.* Figure 6.6). We assign to this edge a weight w_o^{A-B}:

$$w_o^{A-B} = \frac{5}{3} + (1 - 0.25) = 2.4166 \tag{6.20}$$

Similarly, the free capacity $F_{[\Theta_E]}$ of the ***'o'*** edge along the fiber-link $B-C$ during Θ_E is 2 (*c.f.* Figure 6.8). We assign to this edge a weight w_o^{B-C}:

$$w_o^{B-C} = \frac{5}{2} + (1 - 0.25) = 3.25 \tag{6.21}$$

As a result, the weight w_E for extending the existing GL for a period Θ_E is computed as follows:

$$\begin{aligned} w_E &= w_b^A + w_o^{A-B} + w_o^{B-C} + w_e^C \\ &= 0 + 2.4166 + 3.25 + 0 = 5.66 \end{aligned} \tag{6.22}$$

The length of the subintervals Θ_H, Θ_E, and Θ_T are:

$$\overline{\Theta}_H = 43200 \ sec. \tag{6.23a}$$

$$\overline{\Theta}_E = 14400 \ sec. \tag{6.23b}$$

$$\overline{\Theta}_T = \overline{\Theta}_H + \overline{\Theta}_E = 57600 \ sec. \tag{6.23c}$$

Finally, the weight w_i^{A-C} assigned to the ***'i'*** edge is:

$$\begin{aligned} w_i^{A-C} &= \frac{\overline{\Theta}_H}{\overline{\Theta}_T} \times w_H + \frac{\overline{\Theta}_E}{\overline{\Theta}_T} \times w_E \\ &= \frac{43200}{57600} \times 2.2 + \frac{14400}{57600} \times 5.66 = 3.066 \end{aligned} \tag{6.24}$$

To conclude, the incoming sRED can be satisfied by the combination of an existing GL during the time period Θ_H and a new GL during the remaining period Θ_E.

6.5 Weight Assignment in the Case of RLDs

Let us suppose that a new RLD request δ_i^L arrives at the network and needs to be routed on the fly. This new request is characterized by: $\delta_i^L\ \big(s_i, d_i, \alpha_i, \beta_i =?, n_i = 1.0\big)$. The required rate n_i of this connection is normalized with respect to the capacity C_ω of an optical channel. Due to the fact that the life duration of the RLD is unknown, one can suppose that this RLD remains active till the end of the observation (simulation) period ($\beta_i = \infty$). In this case, the weights assigned to the various edges of the network follow the same rules as for sRLDs described in Section 6.3, where the maximum load $Y_{[\alpha_i,\beta_i=\infty]}$ and consequently the free capacity $F_{[\alpha_i,\beta_i=\infty]}$ of an edge are evaluated till the end of the observation period.

6.6 Weight Assignment in the Case of REDs

Let us suppose that a new RED request δ_i^E arrives at the network and needs to be routed and groomed on the fly. This new request is characterized by: $\delta_i^E\ \big(s_i, d_i, \alpha_i, \beta_i =?, n_i < 1.0\big)$. The required rate n_i of this connection is normalized with respect to the capacity C_ω of an optical channel. Due to the fact that the life duration of the RED is unknown, one can suppose that this RED remains active till the end of the observation (simulation) period ($\beta_i = \infty$). In this case, the weights assigned to the various edges of the network follow the same rules as for sREDs described in Section 6.4 except for the weight assigned to ***'i'*** edges. Indeed, β_i being set to infinity, the life duration $\overline{\Theta}_T$ of the incoming RED is also equal to infinity ($\overline{\Theta}_T = \beta_i - \alpha_i = \infty$). Thus, the weight w_i already defined by Equation (6.10) is not computable. In this case, we postulate that a GL can satisfy the incoming RED if its free capacity at the end of the observation period is larger than the data rate required by this RED ($F'_{t=\infty} \geqslant n_i$). This statement guarantees a finite value for the time period $\overline{\Theta}_E$. Thus, the weight w_E for extending the GL for the period Θ_E is still computed as mentioned by Equation (6.11), where the maximum load $Y_{[\Theta_E]}$ and consequently the free capacity $F_{[\Theta_E]}$ of an edge are evaluated during the period Θ_E. However, the time period $\overline{\Theta}_H$ ($\overline{\Theta}_H = \overline{\Theta}_T - \overline{\Theta}_E$) is still infinite. The new proposed weight w_H for holding the incoming RED is defined as follows:

$$w_H = k \times (1 + F'_{t=\infty} - n_i) \tag{6.25}$$

Finally, the weight assigned to an ***'i'*** edge is computed as follows:

$$w_i = \begin{cases} w_H + w_E & \text{if } F'_{t=\infty} \geqslant n_i, \\ \infty & \text{if } F'_{t=\infty} < n_i. \end{cases} \tag{6.26}$$

Example 6.4.

We consider again the simple 3-node and 4-link network topology already introduced in Example 6.1 and shown in Figure 6.3. We assume that the number of WDM channels on each fiber-link is set to 5. A new incoming RED connection request $\delta_i^E \big(s_i = A, d_i = C, \alpha_i = 3:00, \beta_i = ?, n_i = 0.25 < 1.0\big)$ arrives at the network and needs to be routed and groomed on the fly. In this example, we focus mainly on the weight assigned to the ***'i'*** edge representing the grooming lightpath between node A and node C. We recall that this GL is two hops long, and it is routed along the path $A - B - C$. Let us assume that the maximum capacity C'_t and the data rate y'_t carried by this GL are illustrated in Figure 6.10. The free capacity $F'_{t=\infty}$ of its corresponding ***'i'*** edge evaluated at the end of the observation period is equal to 0.5 $($*c.f.* $C'_t - y'_t$ for the right most time value$)$. As $F'_{t=\infty} \geqslant n_i$, we assume that this GL can potentially carry the new request. This assumption must also investigate the possibility of extending the GL during the time period where its offered free capacity is smaller than the required rate of the incoming RED.

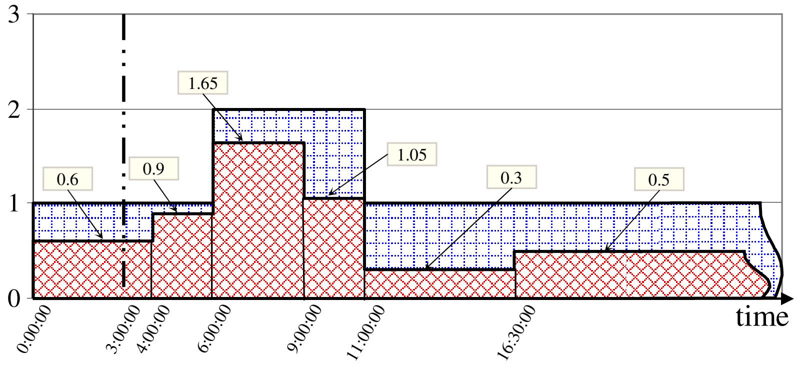


Fig. 6.10. Maximum capacity C'_t and traffic load y'_t of the GL connecting node A to node C

As in the previous Example 6.3, we divide the life duration $\Theta_T = [3:00, \infty]$ of the RED into two subintervals: Θ_H and Θ_E. In this example, $\Theta_H = [3:00, 4:00] \cup [6:00, \infty]$ and $\Theta_E = [4:00, 6:00]$. It is to be noted that the length of the subinterval Θ_E is finite, while the subinterval Θ_H has infinite duration.

If we consider the same load for the various edges as it was illustrated in the previous Example 6.3, the free capacity of the fiber-link $A - B$ during the period $\Theta_E = [4:00, 6:00]$ is $F_{[4:00,6:00]} = 3$ $($*c.f.* Figure 6.6$)$, while the free capacity of the fiber-link $B - C$ during this same period is $F_{[4:00,6:00]} = 2$ $($*c.f.* Figure 6.8$)$. Consequently, the weights w_o^{A-B} and w_o^{B-C} evaluated for this period Θ_E remain the same, and the weight w_E for extending the existing GL for the period Θ_E is still equal to:

$$\begin{aligned} w_E &= w_b^A + w_o^{A-B} + w_o^{B-C} + w_e^C \\ &= 0 + 2.4166 + 3.25 + 0 = 5.66 \end{aligned} \tag{6.27}$$

However, the weight w_H for holding the incoming RED during Θ_H is:

$$\begin{aligned} w_H &= k \times (1 + F'_{t=\infty} - n_i) \\ &= 2 \times (1 + 0.5 - 0.25) = 2.5 \end{aligned} \tag{6.28}$$

Finally, the weight assigned to the ***'i'*** edge is computed as follows:

$$\begin{aligned} w_i &= w_H + w_E \\ &= 2.5 + 5.66 = 8.16 \end{aligned} \tag{6.29}$$

6.7 Additional Issues

- When routing sREDs/REDs, and in order to take into account the complexity of the signaling phase, the null weight assigned to the ***'g'*** edges is replaced by a positive weight. This weight reduces as possible the number of intermediate nodes where the grooming is performed, thus privileging single-hop grooming over multiple-hop grooming.
- In order to implement the ratio κ of the cost of an electrical port to the cost of an optical port in the routing and grooming algorithm, the weights assigned to the ***'b'*** and ***'e'*** edges are incremented by a fixed positive value. This positive value is a parameter to be set in the algorithm.
- The difference between routing sRxDs and routing RxDs can be better outlined when the network carry a mix of scheduled and random demands. However, for a given set of random demands, the sRxDs routing algorithm and the RxDs routing algorithm have almost the same network performance in the absence of any scheduled demands. Indeed, when no preplanned demands are already routed on the network, there is not any network resources that are reserved for future use. In addition, the requests that are currently occupying the network cannot ask for additional resources, but they will free their occupied resources once they end. Consequently, the load carried by a network edge can only decrease with time. Hence, the weight assignment functions are based on the current network load (current network load = maximum network load) and are independent of the duration of the request to be routed. As a result, the weights assigned to the various edges of the network are identical for RxDs routing and sRxDs routing with minor changes for the weights assigned to the ***'i'*** edges. This explains the close performance behavior of the sRxDs routing and the RxDs routing for the same set of random demands in the absence of pre-routed scheduled demands.
- The Dijkstra algorithm combined with the dynamic cost assignment is initially designed to route dynamic traffic requests in a blocking scenario where we try to achieve some objectives depending on the adopted policy. This algorithm can be slightly modified in order to be used as a tool for network dimensioning. In this latter case, the objective of the algorithm is to minimize the number of wavelengths used per fiber. Given a set of requests, the modified algorithm fixes the maximum number of wavelengths per fiber and tries to route as much requests as possible. If some requests cannot be satisfied, the algorithm increases the number of wavelengths per fiber and tries to route additional requests. When all the requests are routed, we evaluate the global network cost expressed as a function of the number of electrical ports and optical ports used.

6.8 Simulation Results and Analysis

6.8.1 Network Dimensioning using the Dijkstra Algorithm

The Dijkstra algorithm is generally used to route traffic in a blocking scenario where we try to achieve some objectives depending on the adopted policy. This algorithm can be slightly modified in order to be used as a tool for network dimensioning. In this later case, the objective of the resulting algorithm is to minimize the congestion observed in the network with/without considering the impact of this strategy on the cost of the network (*c.f.* Section 6.7). For this purpose, let us consider the set SxD_7 of 7874 requests (*c.f.* Section 3.6). This set is decomposed into a subset of 2190 SLD requests and another subset of 5684 SED requests.

Network Dimensioning for SLDs

Our purpose is to compare three routing strategies. The first strategy assigns to each request the shortest path between its source node and its destination node. The second strategy optimizes the routing of the SLDs thanks to the knowledge of time and space correlation between the requests. This is achieved using the Simulated Annealing algorithm introduced in Section 5.6.1. The third strategy routes the set of SLDs while minimizing the congestion observed in the network. This strategy uses the modified Dijkstra algorithm combined with the weight assignment scheme for sRLD requests. It is to be noted that the number of electrical ports used at the source node and the destination node of each request remains unchanged for the various routing solutions. In our example with 2190 SLDs, the number of required electrical ports is of 1875 ports. This is due to the fact that several SLDs among the 2190 share the same source and/or the same destination with at least another time disjoint SLD. Table 6.1 illustrates the SLD routing cost expressed in terms of required electrical ports and optical ports for the three routing strategies.

Table 6.1. Comparing 3 strategies for routing the SLD requests

SLDs' Routing Cost								
Shortest Path Routing	o_1-ports	8850	e_1-ports	-	r_1-ports	-	Overall Cost	20100
	o_2-ports	1875	e_2-ports	944	r_2-ports	931	Network Congestion	161
	o_3-ports	-	e_3-ports	-	r_3-ports	-	Computation Time	0 sec
SA Algorithm K=4	o_1-ports	8404	e_1-ports	-	r_1-ports	-	Overall Cost	19654
	o_2-ports	1875	e_2-ports	944	r_2-ports	931	Network Congestion	136
	o_3-ports	-	e_3-ports	-	r_3-ports	-	Computation Time	5600 sec
Dijkstra Algorithm	o_1-ports	8970	e_1-ports	-	r_1-ports	-	Overall Cost	20220
	o_2-ports	1875	e_2-ports	944	r_2-ports	931	Network Congestion	91
	o_3-ports	-	e_3-ports	-	r_3-ports	-	Computation Time	0 sec

When routing according to the shortest path routing algorithm, the overall network cost is equal to 20100. The SA algorithm allows a request to be routed along one of the 4 shortest paths pre-computed off-line for each source-destination node pair. In this way, the cost of a network able to satisfy all the requests is reduced to 19654 at the last SA iteration. Thus, a reduction by 2.22% of the global network cost is observed between the shortest path routing and the SA routing. One notices that SA algorithm reduces the congestion, which is the number of optical channels used on the most loaded link, from 161 to 136. The SA computing time to get this result is about 5600 sec. On the other hand, it is interesting to note that when routing the SLDs by means of the modified Dijkstra algorithm, the global cost of the network has increased with respect to the cost obtained by means of the SA algorithm. This result is quite natural knowing that routing the SLDs sequentially is sub-optimal in comparison to a global routing strategy like the SA algorithm. In addition, in order to reduce the congestion observed on some links, some requests are routed on longer paths. Furthermore, the congestion observed in this case is equal to 91 which is far smaller than the congestion observed in the case of the SA algorithm.

Conversely, let us now consider the network dimensioned by the proposed SA algorithm. On this network, we try to route the 2190 SLDs sequentially using the Dijkstra algorithm combined with the weight assignment scheme for sRLD requests. In this case, 65 requests out of 2190 are blocked due to the lack of network resources. In other words, for the same network and the same set of SLD requests, using a global optimization tool can satisfy 2.97% more requests than a sequential optimization tool.

Network Dimensioning for SEDs

Our purpose is to compare four routing strategies. The first strategy assigns to each request the shortest path between its source node and its destination node, but it does not allow any grooming to be carried out. The second strategy assigns to each request the shortest path between its source node and its destination node, and then it proceeds to the grooming optimization using our proposed IG algorithm. The third strategy instantaneously routes and grooms each request using the modified Dijkstra algorithm combined with the weight assignment scheme for sRED requests. This strategy aims at minimizing the congestion observed in the network and does not implement the ratio κ of the cost of an electrical port to the cost of an optical port in the weight assignment function. The fourth strategy differs from the third strategy by implementing the ratio κ in the weight assigned to the ***'b'*** and ***'e'*** edges. Table 6.2 illustrates the routing and grooming cost expressed in terms of required electrical ports and optical ports for the four routing strategies.

Table 6.2. Comparing 4 strategies for routing and grooming the SED requests

SEDs' Routing and Grooming Cost								
Shortest Path Routing without Grooming	o_1-ports	26660	e_1-ports	3519	r_1-ports	3519	Overall Cost	104078
	o_2-ports	-	e_2-ports	-	r_2-ports	-	Network Congestion	452
	o_3-ports	7038	e_3-ports	3519	r_3-ports	3519	Computation Time	0 sec
Shortest Path Routing with IG Grooming	o_1-ports	12958	e_1-ports	3519	r_1-ports	3519	Overall Cost	73300
	o_2-ports	-	e_2-ports	-	r_2-ports	-	Network Congestion	206
	o_3-ports	4192	e_3-ports	2094	r_3-ports	2098	Computation Time	3686 sec
Dijkstra algorithm without implementing κ	o_1-ports	12242	e_1-ports	3519	r_1-ports	3519	Overall Cost	95450
	o_2-ports	-	e_2-ports	-	r_2-ports	-	Network Congestion	101
	o_3-ports	8003	e_3-ports	4003	r_3-ports	4000	Computation Time	44 sec
Dijkstra algorithm with κ implementation	o_1-ports	13184	e_1-ports	3519	r_1-ports	3519	Overall Cost	79364
	o_2-ports	-	e_2-ports	-	r_2-ports	-	Network Congestion	111
	o_3-ports	5165	e_3-ports	2581	r_3-ports	2584	Computation Time	56 sec

When routing the SED requests along the shortest path without undergoing any grooming process, the cost of the network is equal to 104078. Without altering the path assigned to these SEDs and by allowing some requests to be groomed together according to the IG algorithm, the network cost decreases to 73300. We notice also that the grooming procedure reduces the congestion from 452 to 206. These results are obtained when assuming that the ratio κ of the cost of an electrical port to the cost of an optical port is equal to 5. Thus, a reduction by 29.57% of the global network cost is observed when applying the grooming procedure. The computing time to get this result is roughly one hour.

On the other hand, it is interesting to note that when routing the SEDs by means of the modified Dijkstra algorithm combined with the weight assignment scheme for sRED requests, the global network cost is equal to 95450. This result is quite natural knowing that Dijkstra algorithm aims to use the existing grooming lightpaths to their maximum extents. In addition, in order to reduce the congestion observed on some links, we try to groom the requests together as much as possible. Furthermore, the congestion observed in this case is equal to 101 which is far smaller than the congestion observed in the other two cases. The time needed to get this result is of 44 sec. In order to reduce the number of required electrical ports, the ratio κ has been implemented in the weight assignment scheme. As a result, the global network cost decreases to 79364 but the congestion reaches 111. This result was expected knowing that in this case, the Dijkstra algorithm reduces the number of requests to be groomed. As a result, the number of o_1 optical ports increases, while the number of e_3 and r_3 electrical ports decreases.

Conversely, let us now consider the network dimensioned by the proposed IG algorithm. On this network, we try to route the 5684 SEDs sequentially using the Dijkstra algorithm combined with the weight assignment scheme for sRED requests. In this case, 444 requests out of 5684 are blocked due to the lack of network resources. In other words, for the same network and the same set of SED requests, using a global optimization tool can satisfy 7.81% more requests than a sequential optimization tool.

Network Dimensioning for SxDs

Table 6.3. Resource sharing between SLDs and SEDs

SxDs' Routing Cost									
Dijkstra Algorithm for sRLDs	o_1-ports	8970	e_1-ports	-	r_1-ports	-	Overall Cost	20220
	o_2-ports	1875	e_2-ports	944	r_2-ports	931	Network Congestion	91
	o_3-ports	-	e_3-ports	-	r_3-ports	-	Computation Time	0 sec
Dijkstra algorithm for sREDs with κ implementation	o_1-ports	13184	e_1-ports	3519	r_1-ports	3519	Overall Cost	79364
	o_2-ports	-	e_2-ports	-	r_2-ports	-	Network Congestion	111
	o_3-ports	5165	e_3-ports	2581	r_3-ports	2584	Computation Time	56 sec
Dijkstra algorithm for sRxDs with κ implementation	o_1-ports	21830	e_1-ports	3519	r_1-ports	3519	Overall Cost	98552
	o_2-ports	1875	e_2-ports	944	r_2-ports	931	Network Congestion	199
	o_3-ports	5047	e_3-ports	2519	r_3-ports	2528	Computation Time	61 sec

When dimensioning a network able to satisfy the SLDs alone by means of the Dijkstra algorithm combined with the weight assignment scheme for sRLDs, the overall network cost is equal to 20220. Similarly, when dimensioning a network able to satisfy the SEDs alone by means of the Dijkstra algorithm combined with the weight assignment scheme for sREDs and implementing the ratio κ, the overall network cost is equal to 79364. However, by simultaneously routing and grooming both SLDs and SEDs using the modified Dijkstra algorithm combined with the weight assignment scheme for sRxDs, and by allowing some network resources reserved for SLDs to be re-used by SED requests, the overall network cost is equal to 98552. Compared to the sum of the network cost of the SLDs alone and that of the SEDs alone ($20220 + 79364 = 99584$), this network cost represents a gain of 1.04%. This highlights the capacity of the Dijkstra algorithm to share resources between full-wavelength (xLDs) and sub-wavelength (xEDs) requests.

6.8.2 Traffic Engineering using the Dijkstra Algorithm

In this section, we want to evaluate the blocking ratio of the Dijkstra algorithm for different traffic scenarios. For this reason, we suppose that the network is already dimensioned using the SA algorithm to satisfy the set of 2190 SLD requests inherent to the traffic set SxD_7. Once the network is dimensioned, the SLDs are routed by reserving their required resources during their active period. Subsequently, the network can be over-dimensioned by providing additional resources in order to route random traffic demands. In our case study, the parameters $\ddot{\alpha}$, $\ddot{\beta}$, and $\ddot{\gamma}$ of the over-dimensioning procedure are set to 15, 15, and 25, respectively. These values are chosen in order to prevent any sRxD/RxD blocking due to the lack of electrical ports. Thus, blocking will only be due to the lack of o_1 optical ports. The number of o_1-ports is over-dimensioned according to the parameter $\ddot{\eta}$ chosen in the range $0\% - 20\%$.

At this stage, sRxDs/RxDs can be routed sequentially using the Dijkstra algorithm combined with a suitable weight assignment scheme corresponding to the type of the request to be routed. This traffic engineering approach takes into account the need of the sub-wavelength requests to an electrical grooming process. For this purpose, we have considered 3 sets of random requests (*c.f.* Section 3.6):

- The set $sRxD_1/RxD_1$ of 2659 sRLDs/RLDs. This corresponds to the arrival of a new incoming sRLD/RLD every 30 sec that asks for some network resources for an hour period.
- The set $sRxD_2/RxD_2$ of 8158 sREDs/REDs. This corresponds to the arrival of a new incoming sRED/RED every 10 sec that asks for some network resources for about 2400 sec.
- The set $sRxD_3/RxD_3$ of 10817 sRxDs/RxDs. This set is the combination of the sets $sRxD_1/RxD_1$ and $sRxD_2/RxD_2$.

It is to be noted that the CPU time needed to determine a routing path for a random request exceeds rarely 10 ms depending on the network load. Assuming that the time needed to establish a connection is of several hundreds of milliseconds including the signalling stage as well as the path determining stage, we assume that enough time remains available for the signalling process.

Traffic Engineering for RxDs: Our Approach vs Existing Approach

In this section, we compare our proposed routing and grooming algorithm with an existing approach. For instance, the authors in [J142] use a standard shortest path computation algorithm to route the RxD requests. They propose to provision a connection through the least cost route. The cost of a lightpath link is equal to the overall cost of the concatenated fiber-links that the lightpath traverses. Every free fiber-link has a unit cost.

Our algorithm differs from the previous algorithm not only by the cost assignment functions but also by the capability to handle the routing information of the SxDs. In other words, when a new RxD request arrives at the network, the existing algorithm tries to route it using only the additional resources provided when over-dimensioning the network. Thus, the existing algorithm uses dedicated resources for the RxDs and does not use any SxD resources even if they are free. Conversely, our proposed algorithm tries to route an incoming RxD request using the free resources provided by the over-dimensioning procedure as well as some free SxD resources. The life duration of the RxDs being unknown, we assume, at the instant of arrival of an RxD, that this RxD will remain active till the end of the observation period. Once it ends, we free its occupied resources.

RLDs Rejection Ratio

We investigate the routing of the RLDs alone assuming that the SxDs are already routed. Simulations are carried out to show the impact of the over-dimensioning factor on the RLDs rejection ratio. Figure 6.11 plots the rejection ratio for RLDs using our approach and the existing approach. The results show the outperformance of our proposed algorithm compared to the existing one. For instance, for small over-dimensioning factor, the rejection ratio of the RLDs is decreased by at least 10%.

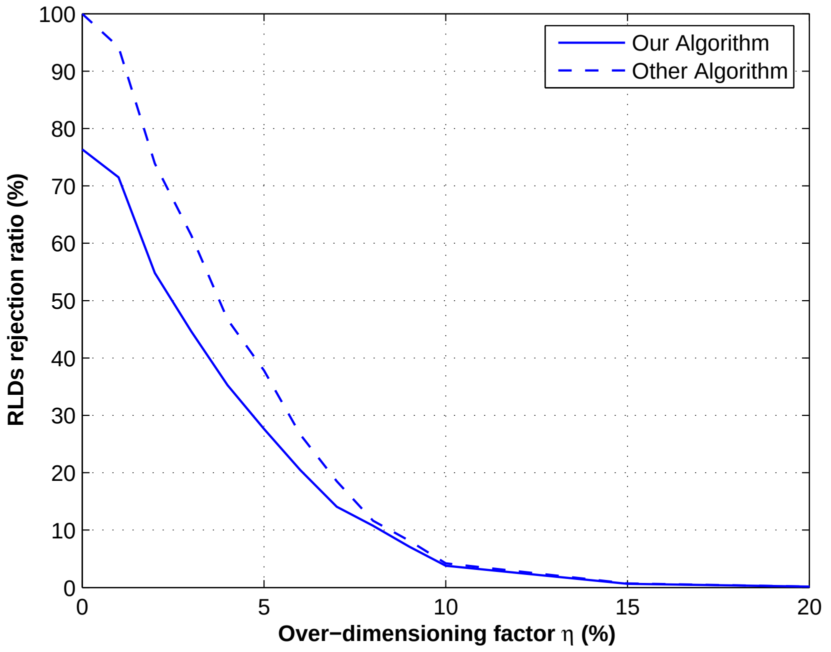


Fig. 6.11. RLDs rejection ratio as function of the network over-dimensioning

REDs Rejection Ratio

We investigate also the routing and grooming of the REDs alone assuming that the SxDs are already routed. In order to highlight the benefit of our proposed weight assignment scheme, an enhanced version of the already existing algorithm is developed. In this enhanced version, a new RED request can be routed using the free resources provided when over-dimensioning the network as well as some free SxD resources. However, the free fiber-links have retained their unitary weight.

Simulations are carried out to illustrate the gain brought out by means of traffic grooming. In addition, a comparison between our proposed algorithm and the algorithm proposed in [J142] is drawn in Figure 6.12. These simulations show the outperformance of our approach as well as the benefit of our weight assignment scheme. For instance, when $\ddot{\eta}$ is equal to 5%:

- When no grooming is applied, and allowing the REDs to use some SxD resources, about 46% of the REDs are rejected.
- With the proposed algorithm in [J142], about 41% of the REDs are rejected. This rejection ratio decreases to 31% by using the enhanced version of the algorithm.
- Finally, with our proposed algorithm, only 24% of the REDs are rejected.

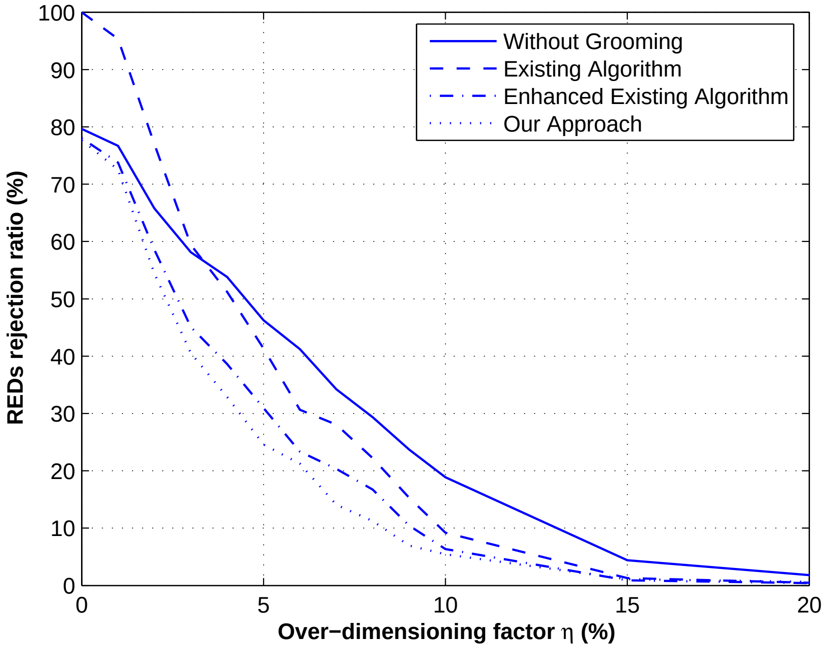


Fig. 6.12. REDs rejection ratio as function of the network over-dimensioning

RxDs Rejection Ratio

In this section, we consider both the RLDs and the REDs of the traffic set $sRxD_3/RxD_3$. Assuming that the SxDs are already routed, we try to route on the fly these RxDs using only our proposed algorithm. Figure 6.13 illustrates the rejection ratio of the RLDs alone, the rejection ratio of the REDs alone, and the rejection ratio of the global set of RxDs. For instance, when $\ddot{\eta}$ is equal to 10%, about 34% of the RLDs are rejected, while the rejection ratio of the REDs is about 13%. Thus, the overall rejection ratio is evaluated to about 16%. We note that when the network is overloaded, the already established GLs may still have some free capacity to route additional low-speed traffic requests. This explains the relative high rejection ratio of the RLDs compared to the rejection ratio of the REDs for the same over-dimensioning factor $\ddot{\eta}$.

Traffic Engineering: sRxDs vs RxDs

The aim of this section is to outline the impact of the predictability of the lifetime of a demand on the network performance. This impact is outlined when the network carries a mix of scheduled traffic and random traffic. For this purpose, we consider the network already dimensioned using the SA algorithm to satisfy the set of 2190 SLD requests inherent to the SxD_7 traffic set. Once the network is dimensioned, the SLDs are routed by reserving their required resources during their active period. Then, we try to route on the fly a set of random demands using the Dijkstra algorithm combined with an appropriate weight assignment scheme.

When the life duration of the random demands is unknown (RxDs), one may assume that these requests remain active till the end of the observation period. Thus, these random demands are routed using the Dijkstra algorithm combined with the weight assignment schemes introduced in Section 6.5 and Section 6.6. However, if we consider the same set of demands and we assume that their life duration is given at their instant of arrival (sRxDs), these requests can be routed using the Dijkstra algorithm

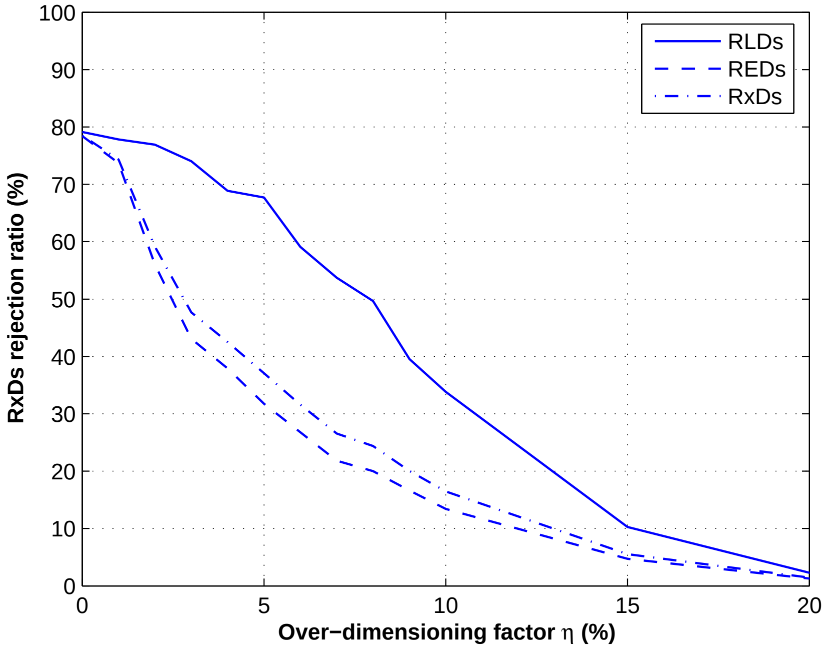


Fig. 6.13. RxDs rejection ratio as function of the network over-dimensioning

combined with the weight assignment schemes introduced in Section 6.3 and Section 6.4. Figure 6.14 and Figure 6.15 illustrate the rejection ratios of both cases (unknown/known life duration) for full-wavelength traffic requests and sub-wavelength traffic requests, respectively. From these numerical results, it can be observed that the network performance decreases drastically when we ignore the life duration of the random demands.

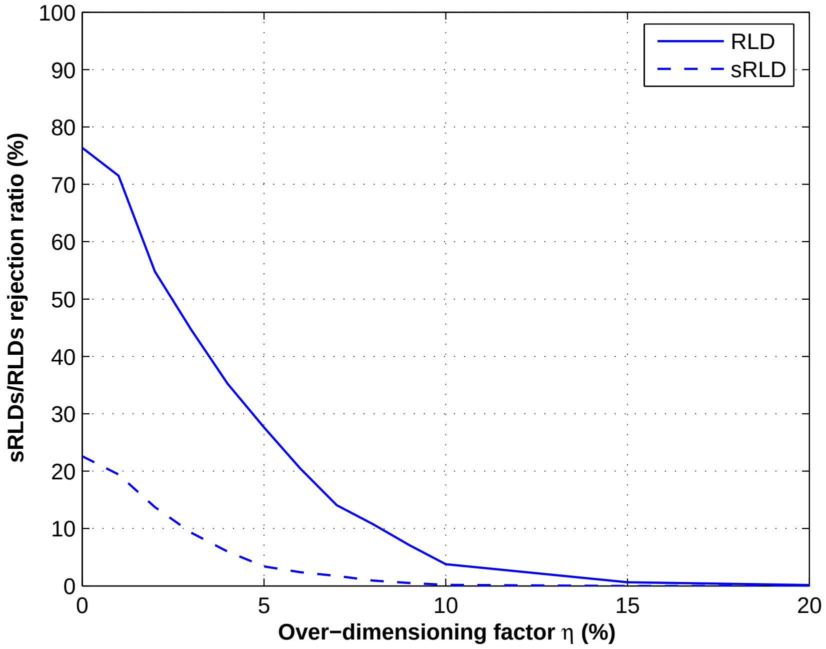


Fig. 6.14. RLDs rejection ratio vs sRLDs rejection ratio

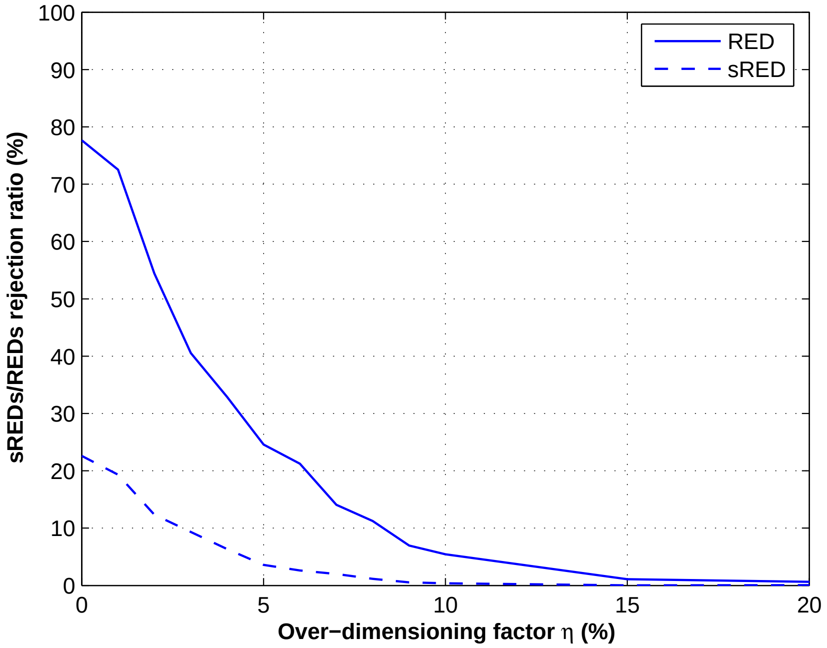


Fig. 6.15. REDs rejection ratio vs sREDs rejection ratio

7

Rerouting Strategies

Abstract: *In this last chapter, we consider the problem of the best effort Quality of Service provided to random traffic demands in WDM networks. In general, when a new random demand arrives at the network, this demand is accepted only based on the available network resources at its instant of arrival. In such a case, this random demand may be preempted if its assigned resources are reserved for future use by a scheduled demand. Consequently, random demands may be subject to high rejection ratio depending on the amount of reserved network resources. In order to improve the blocking performance and the network efficiency, rerouting techniques are addressed. Two approaches are proposed in this matter: Path Adjusting Rerouting (PAR) and Time Limited Resource Reservation (TLRR). The former extends the concept of wavelength retuning scheme implementing crossover edges. The resulting PAR approach is capable of rerouting a lightpath or connection by modifying its physical route. The latter (TLRR) is quite different from the traditional dynamic routing algorithms. TLRR consists in providing a time-limited guaranteed service to random demands instead of a pure best effort service.*

In the previous chapter, we have introduced two routing and grooming strategies to set-up lightpaths for pure random demands (RLDs and REDs) and for semi-random demands (sRLDs and sREDs) with the global objective of optimizing network resources utilization. When a new connection request arrives, a proper path consisting of a sequence of links is selected. The approach is to transform the network into a weighted graph. The weight label on each edge in the graph depends on the current link status and represents the cost of routing a new circuit on that link. For instance, when routing an RLD, each fiber-link is assigned a weight inversely proportional to its free capacity. Thus, a fully used fiber-link is assigned an infinite cost. The routing algorithm tries to find the shortest path between the source node and the destination node of the incoming request using a conventional shortest path algorithm like Dijkstra's algorithm. If the algorithm fails to find a path with a finite cost, the new connection request is blocked.

Normally, once a lightpath has been set-up, its physical path and its wavelength are not changed in order to guarantee a smooth and continuous data communication. In spite of this, rerouting techniques may be addressed to improve the blocking performance and the network efficiency in the context of dynamic lightpath establishment. The basic idea of rerouting is to reroute some of the existing requests so that the new incoming request can be set-up in the network. A rerouting scheme can change the wavelength as well as the original path of the existing lightpaths to satisfy a new connection. However, when wavelength conversion is deployed in each node of the network, wavelength retuning is useless. Hence, the original path, also called basepath, of some existing lightpaths can be changed in order to accept a new request. The objective of any rerouting algorithm is to minimize the weighted number of rerouted lightpaths in the network while satisfying a new connection request. Rerouting schemes initiated in response to a change in traffic patterns or in response to a network failure are out of the scope of this thesis.

As a first approach, we propose a "Path Adjusting Rerouting" (PAR) algorithm. PAR extends a traditional rerouting approach to the case of multi-layer EXC/OXC nodes connected using unidirectional fiber-links. It is mainly based on the concept of crossover edges representing retunable lightpaths, and it aims to reduce the resource contention between random demands. As a second approach, we propose a "Time Limited Resource Reservation" (TLRR) algorithm. TLRR mainly deals with the concurrency between scheduled and pure random demands. It is quite different from traditional dynamic routing algorithms. Indeed, when a new connection request cannot be satisfied due to the current network state, traditional rerouting algorithms try to reroute some existing connections in order to accommodate the incoming connection request. However, the TLRR approach is based on a transient reservation window Δ. In its normal phase, TLRR tries to satisfy an incoming RxD request during a time period Δ. Beyond Δ, if some of the resources assigned to the RxD are reserved to be used by a higher priority request such as an SxD, this RxD is rerouted onto a new path for a new period Δ. By convention, an SxD must benefit from its guaranteed resources and cannot be rerouted.

7.1 Path Adjusting Rerouting (PAR) Algorithm

Wavelength rerouting is a viable and cost effective mechanism which can improve the blocking performance. However, Lee *et al.* in [J69] have shown that the rerouting problem is in general *NP*-complete. They also proposed a polynomial time rerouting scheme called "Move-To-Vacant Wavelength Retuning" (MTV-WR). MTV-WR moves a lightpath from one wavelength to another wavelength on the same path. This scheme has many desirable features such as short disruption period and simple switching control. The algorithm runs in polynomial time, but is not time optimal. The authors in [J143] presented a Dijkstra-like time optimal algorithm for wavelength rerouting with the parallel MTV-WR rerouting scheme.

The MTV-WR algorithm proved its efficiency to select the minimum weighted number of lightpaths to be rerouted when the lightpaths, the links, and the channels are all bidirectional. However, it fails to choose the minimum for the unidirectional case. As our auxiliary graph model for the network nodes is based on unidirectional edges, this motivated us to propose an extension of this algorithm to the unidirectional case. In addition, when wavelength conversion is deployed in each node of the network, wavelength retuning is useless. Another major modification is the extension of the MTV-WR algorithm to support path adjusting rerouting. In the following, we start by briefly describing the MTV-WR algorithm. Then, we introduce step by step the major extensions.

7.1.1 Traditional Rerouting Algorithm

In this section, we describe briefly the algorithm proposed in [J69, J143]. The objective of this algorithm is to minimize the weighted number of rerouted lightpaths in the network with the parallel MTV-WR rerouting scheme. The algorithm transforms the network into a graph. A network composed of N nodes, L links, and W wavelength channels per fiber-link is represented as an undirected graph with W subgraphs. Each subgraph (also called wavelength plane) corresponds to the network topology on a particular wavelength. The vertices in a subgraph correspond to the network nodes, and the edges correspond to the wavelength channels on the fiber-links of the network. These subgraphs are disjoint if the network does not allow for wavelength conversion. Let M denotes the cost of retuning a single WDM channel. Thus, the cost of retuning a lightpath is equal to M times the number of hops the lightpath passes through. Subsequently, a free edge in the network graph is assigned a tiny value ϵ. An edge used by a retunable lightpath is assigned a weight equal to the cost of retuning the whole lightpath. Finally, an edge used by a non-retunable lightpath is assigned an infinite weight. The cost M of rerouting a single WDM channel is chosen to be a large number such that the cost of satisfying an incoming request without any lightpath rerouting will always be smaller than the cost of satisfying a new lightpath while rerouting some of the existing lightpaths. For example, M may be any number which is larger than the sum of all the links' weights in the network.

A new connection request arrives randomly as a Poisson process with an exponentially distributed holding time. The algorithm works in two phases: the normal routing phase and the rerouting phase. In the first phase, the algorithm tries to satisfy the new connection without requiring any rerouting. This is accomplished by finding the shortest path in each of the W subgraphs using a conventional shortest path algorithm (like Dijkstra's algorithm). Then, the algorithm selects the path with the least cost. The rerouting phase is only executed when the normal routing phase fails to find a path with a finite cost for the incoming connection. The rerouting phase tries to reroute some lightpaths or connections so that the new connection can be accepted. It proceeds in three stages. In stage 1, it identifies all the retunable lightpaths. In stage 2, it constructs an auxiliary graph by creating crossover edges for every retunable lightpath. A crossover edge is created between the node x and the node y of a retunable lightpath p if there exists a path of length two or more between x and y comprising only the edges of p. In stage 3, shortest paths are found using a conventional shortest path algorithm in each of the subgraphs, and the shortest path with the minimum weighted number (if finite) of rerouted lightpaths is chosen. If no path with a finite cost can be found, the connection request is rejected.

Example 7.1.

As an illustration, let us consider a simple network scenario as described in Figure 7.1. There are 3 available wavelengths per fiber-link, and the network nodes do not provide any wavelength conversion capability. Hence, this network is represented by three disjoint wavelength planes. In addition, three requests δ_1, δ_2, and δ_3 are already satisfied. For instance, δ_1 is already routed on wavelength λ_1 between node 2 and node 5.

We assume that a new request δ_4 arrives and needs to be routed between node 1 and node 4. In the initial routing phase, an active wavelength is assigned an infinite weight, while a free wavelength is assigned a small weight ϵ. According to the current network state, no path with a finite weight can be found for this new connection request. Thus, the rerouting phase is initiated. This rerouting phase is decomposed into three stages/steps:

- First, we identify all the retunable lightpaths in the network. The cost of each link of a retunable lightpath is replaced by the cost of retuning the totality of this lightpath. This cost is equal to M times the number of hops the lightpath passes through, where

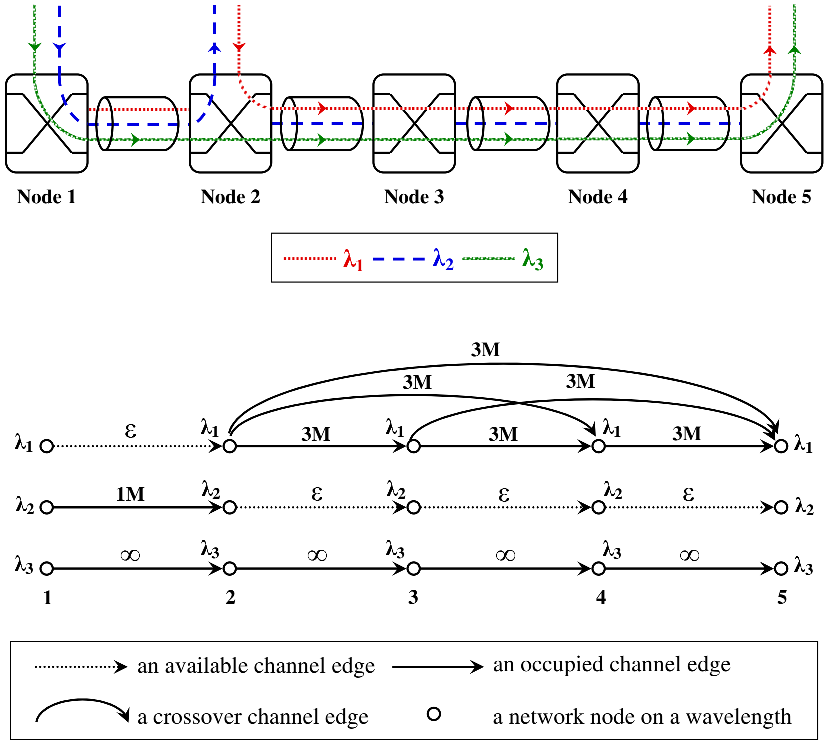


Fig. 7.1. An example of the MTV-WR rerouting algorithm

M denotes a large number chosen larger than the cost of the longest possible lightpath in the network. However, the cost of a non-retunable lightpath remains infinity. For instance, the edge between node 1 and node 2 on wavelength λ_1 is free and will be assigned a cost ϵ. Similarly, the edge between node 1 and node 2 on wavelength λ_2 is active and corresponds to a retunable lightpath. This lightpath is one hop long. Thus, the weight assigned to its corresponding edge between node 1 and node 2 on wavelength λ_2 is $1 \times M$. Finally, the edge between node 1 and node 2 on wavelength λ_3 is active, but its corresponding lightpath is not retunable. Hence, its weight remains equal to infinity.

- Second, the graph is slightly modified by introducing crossover edges for every retunable lightpath. For instance, let us consider the request δ_1 between node 2 and node 5. δ_1 is routed along the path $2-3-4-5$ on wavelength λ_1. As this request can be retuned on wavelength λ_2, crossover edges are established between any two non-adjacent nodes in the set $\{2, 3, 4, 5\}$. A common weight is assigned to all these crossover edges. This weight is equal to the cost of retuning the totality of the lightpath (or sub-lightpath if wavelength conversion is used) used by δ_1. In our example, this cost is equal to $3 \times M$ since this lightpath is three hops long.
- Finally, using conventional shortest path algorithm, the path with the minimum weight between node 1 and node 4 is selected. In our example, the path $1-2-3-4$ on wavelength λ_2 is selected since $(1 \times M + 2 \times \epsilon)$ is smaller than $(3 \times M + \epsilon)$.

In general, if no path with a finite cost is possible at the end of the last step, the incoming connection request is rejected.

7.1.2 Extension to the Unidirectional Case

Mohan *et al.* have introduced in [J143] an example wherein they show the failure of the algorithm to select the minimum weighted number of lightpaths to be rerouted when the lightpaths, the links, and the channels are all unidirectional. However, the origin of this failure remained unknown, and no solution was proposed. Before investigating this problem, let us review in Example 7.2 the scenario considered in [J143].

Example 7.2.

In this example, we consider the state of the network on a particular wavelength plane at a given instant of time. This network status is represented by the subgraph shown in Figure 7.2. On this wavelength plane, three lightpaths $\wp_1$, $\wp_2$, and $\wp_3$ are already routed using the unidirectional paths $7-3-6-9$, $8-4-1$, and $2-4-5$, respectively. We assume that all these lightpaths are retunable to another wavelength plane.

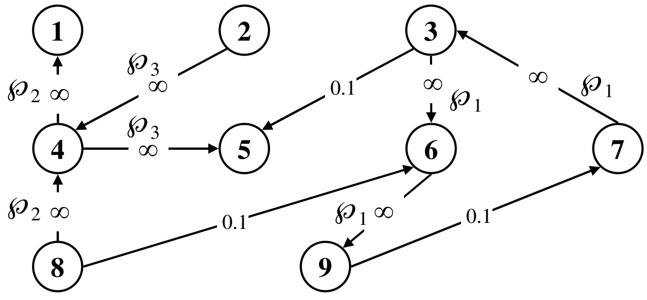


Fig. 7.2. A graph representing the status of the network on a particular wavelength plane with unidirectional paths

In addition, we assume that there is a request for a session from node 8 to node 5. In the normal routing phase, the weight of a free edge is fixed to $\epsilon = 0.1$, while the weight of the edges used by the lightpaths $\wp_1$, $\wp_2$, and $\wp_3$ are set to infinity. In this current state, no path with a finite weight can be found on this wavelength plane. We assume also that no route with a finite cost is available on any of the other wavelength planes to satisfy the incoming request. In such a scenario, the normal routing phase fails, and the rerouting phase is initiated. In the first stage of the rerouting phase, all the three lightpaths $\wp_1$, $\wp_2$, and $\wp_3$ are found to be retunable. Then, stage 2 builds an auxiliary graph by creating crossover edges as shown in Figure 7.3. In this auxiliary graph, the weight of a free edge is still equal to $\epsilon = 0.1$. However, the weight of the edges used by a retunable lightpath $\wp_i$ as well as the weight of its corresponding crossover edges are changed to $M \times k_i$; where k_i is the number of hops that the lightpath traverses, and M is the cost of retuning a single channel. For instance, crossover edges are created between the non-adjacent nodes $(7, 6)$, $(7, 9)$, and $(3, 9)$ representing the fact that they are connected by the same retunable lightpath $\wp_1$. These crossover edges as well as the original graph edges $7-3$, $3-6$, and $6-9$ are assigned a weight equal to $3 \times M$ corresponding to the cost of retuning the whole lightpath $\wp_1$.

Finally, in the third stage of the rerouting phase, a shortest path between node 8 and node 5 is computed on this wavelength plane, and the path $8-4-5$ of cost $4 \times M$ is returned.

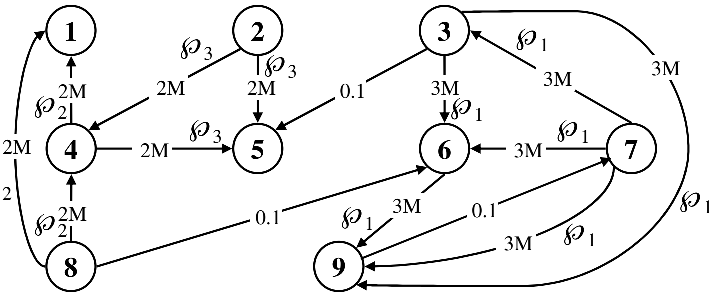


Fig. 7.3. The auxiliary graph with the crossover edges representing the retunable lightpaths

This value represents the cost of retuning the lightpaths $\wp_2$ and $\wp_3$. It is to be noted that, according to the current state of the auxiliary graph, the path $8-6-9-7-3-5$ has a weight of $6 \times M + 0.3$. This is incorrect since routing the incoming request on the path $8-6-9-7-3-5$ requires the rerouting of the lightpath $\wp_1$ of cost $3 \times M$ and the use of three free edges of cost 0.1 each. Thus, the weight of the path $8-6-9-7-3-5$ should be $3 \times M + 0.3$ instead of $6 \times M + 0.3$. Subsequently, the path with the minimum weighted number of existing lightpaths to be rerouted is $8-6-9-7-3-5$ and not $8-4-5$ as reported by the algorithm ($3 \times M + 0.3 < 4 \times M$).

In general, a crossover edge corresponds to two or more consecutive edges along the path of a retunable lightpath. This crossover edge indicates that its underlying edges will become simultaneously free by only retuning a single lightpath. The weight of this crossover edge is the cost of retuning the totality of the lightpath. However, a problem happens in the unidirectional case when we use a down-stream edge before an upper-stream edge. In this case, there is not any crossover edge that connects these two edges and points out their belonging to the same lightpath. Hence, using both edges results in counting the cost of retuning the same lightpath twice. Indeed, this happened in the Example 7.2 when we computed the weight of the path $8-6-9-7-3-5$. In this case, the down-stream edge $6-9$ of the lightpath $\wp_1$ is used before the upper-stream edge $7-3$ of the same lightpath $\wp_1$. Thus, when computing the weight of the path $8-6-9-7-3-5$, the shortest path algorithm added the cost of retuning the lightpath $\wp_1$ twice and returned a cost of $6 \times M + 0.3$.

This problem is overcome by using additional crossover edges. We refer to these new edges as **"Type-1 crossover edges"**. They are added between non-successive edges when the retunable/reroutable lightpath is three hops long or more. A Type-1 crossover edge is created to represent that the down-stream edge and the upper-stream edge will simultaneously become free by only rerouting a given single connection. In contrast with the initial type of crossover edges, the Type-1 crossover edge connects the ingress vertex of the down-stream edge to the egress vertex of the upper-stream edge using additional free edges of the network. For this reason, a shortest path is computed between the egress vertex of the down-stream edge and the ingress vertex of the upper-stream edge using only free edges in the network. Let $\wp$ be this path and $w_\wp$ be its weight. In the auxiliary graph, using this Type-1 crossover edge is equivalent to the use of its corresponding down-stream edge, the path $\wp$, and its upper-stream edge, consecutively. Thus, the weight assigned to this crossover edge is equal to the cost of retuning/rerouting the totality of the lightpath augmented by the weight $w_\wp$ of the path $\wp$. However, in the bidirectional scenario, there is no need to Type-1 crossover edges as the initial type of crossover edge connects the corresponding vertex in both directions. Let us go back to the previous example to illustrate how Type-1 crossover edges can solve the problem.

Example 7.3.

In this example, we consider the same wavelength plane of the previous Example 7.2 given in Figure 7.2. We assume that there is a request for a session from node 8 to node 5. As in the previous example, the normal routing phase fails to find a path with a finite cost between node 8 and node 5. Thus, the rerouting phase is initiated, and all the three lightpaths $\wp_1$, $\wp_2$, and $\wp_3$ are found to be retunable. Consequently, we build the auxiliary graph by creating the initial type of crossover edges. In addition, we need to add Type-1 crossover edges. Only the lightpath $\wp_1$ is three hops long. According to the direction of the data flow, the down-stream edge of lightpath $\wp_1$ is $6-9$ and its up-stream edge is $7-3$. A shortest path is computed between the egress vertex 9 of the edge $6-9$ and the ingress vertex 7 of the edge $7-3$. This path computation returns the path $9-7$ with a weight of 0.1. Consequently, a Type-1 crossover edge is created between the ingress vertex 6 of the edge $6-9$ and the egress vertex 3 of the edge $7-3$. This crossover edge is assigned a weight equal to the sum of the cost $3 \times M$ of retuning the lightpath $\wp_1$ and the weight 0.1 of the path $9-7$. The resulting auxiliary graph is represented in Figure 7.4.

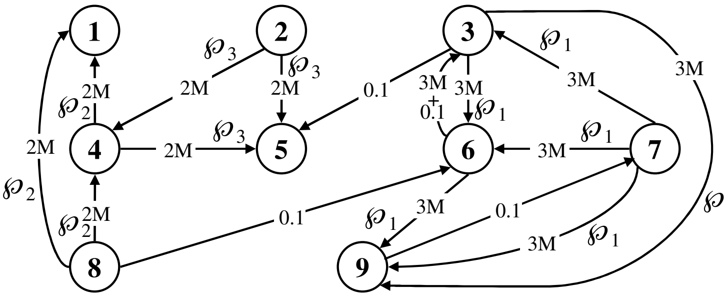


Fig. 7.4. The auxiliary graph with Type-1 crossover edges

Finally, in the third stage of the rerouting phase, a shortest path between node 8 and node 5 is computed on this wavelength plane, and the path $8-6-3-5$ of cost $3 \times M + 0.3$ is returned. One notices the use of the Type-1 crossover edge $6-3$ which solves the problem in the unidirectional case.

7.1.3 Extension to the Path Adjusting Case

The basic idea of rerouting is to reroute some of the existing requests so that the new incoming request can be set-up in the network. A rerouting scheme can change the wavelength as well as the original path of the existing lightpaths or connections to satisfy a new request. Wavelength retuning retunes the wavelength of an existing lightpath or connection maintaining the same path. The original wavelength and the new alternate wavelength are distinct and are represented on two different wavelength planes. However, when wavelength conversion is deployed in each node of the network, wavelength retuning is useless. Hence, the original path, also called "basepath", of some existing lightpaths can be modified in order to accept the new request. A key difference between wavelength retuning and path adjusting is that the original basepath of a lightpath or connection and its alternate path may share some links. Thus, the basepath of a lightpath or connection and its alternate path are not totally link disjoint. Subsequently, creating a crossover edge connecting a vertex from one side of a common link to another vertex at the other side of this common link is not desirable since it contraries the definition of the crossover edges. Indeed, connecting such vertices by a crossover edge implies that all the underlying edges will become simultaneously free by rerouting a single lightpath from its original basepath to the new alternate path. This is not true specially for the common edge where both the original basepath and the alternate path pass through. To conclude, the crossover edges must be adapted to the path adjusting rerouting scenario.

For this purpose, we decompose the path of the original basepath of a lightpath or a connection into groups referred to as “Consecutive Uncommon Links Group” (CULG). As its name suggests, a CULG groups the consecutive links of a basepath that are not used by the alternate path. If a CULG has two consecutive links or more, **“Type-2 crossover edges”** are created to connect non-adjacent vertices in the CULG. Type-2 crossover edges are the simplest type of crossover edges and are equivalent to the original type of crossover edges applied inside the CULG. They reflect the fact that the two ends of the crossover edge are connected by a series of links that will become simultaneously free by rerouting the considered lightpath or connection from its basepath to the new alternate path. As for the original type of crossover edges, Type-2 crossover edges are assigned a weight equal to the cost of rerouting the totality of the lightpath or connection.

In order to point out that the links of different CULGs correspond to the same lightpath or connection, and consequently they will become simultaneously free when rerouting this lightpath/connection, additional crossover edges are added. We refer to these additional edges as **“Type-3 crossover edges”**. A Type-3 crossover edge connects the ingress vertex of an up-stream edge in a source CULG to the egress vertex of a down-stream edge in a destination CULG using some free edges of the network as well as some edges along the basepath of the lightpath or connection. For this purpose, all the edges along the original basepath between the up-stream edge and the down-stream edge are assigned a null weight if they belong to any CULG group. Then, a shortest path is computed between the egress vertex of the up-stream edge and the ingress vertex of the down-stream edge. Let $\wp$ be this path and $w_\wp$ be its weight. The path $\wp$ is made of free edges as well as of edges from the original basepath that will become free when the lightpath/connection is rerouted. A Type-3 crossover edge is created between the ingress vertex of the up-stream edge and the egress vertex of the down-stream edge. This edge indicates that the up-stream edge, the edges along the path $\wp$, and the down-stream edge will become/are free to accept any incoming connection after rerouting a single lightpath or connection. By analogy to Type-1 crossover edges, the weight assigned to such Type-3 crossover edges is equal to the cost of rerouting the totality of the lightpath/connection augmented by the weight $w_\wp$ of using the shortest path $\wp$ between the corresponding up-stream/down-stream edges. For an illustration on the use of the various types of crossover edges, let us consider the following example.

Example 7.4.

Let us consider the network given by Figure 7.5. On this network, a lightpath is already established along the path $A - B - C - D - E - F - G - H - I - J$. We assume that this basepath can be rerouted onto the new alternate path $A - K - L - E - F - M - N - J$.

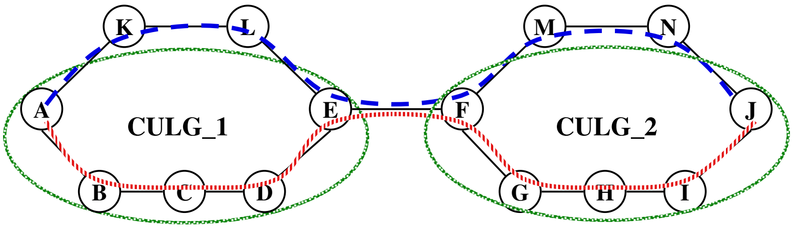


Fig. 7.5. Determining the CULG groups corresponding to a given basepath and its alternate path

Both the basepath and the alternate path use the same edge $E - F$. Thus, we mainly distinguish two CULG groups; namely CULG_1 and CULG_2. The CULG_1 contains the consecutive edges $A - B$, $B - C$, $C - D$, and $D - E$. Similarly, the CULG_2 contains the consecutive edges $F - G$, $G - H$, $H - I$, and $I - J$.

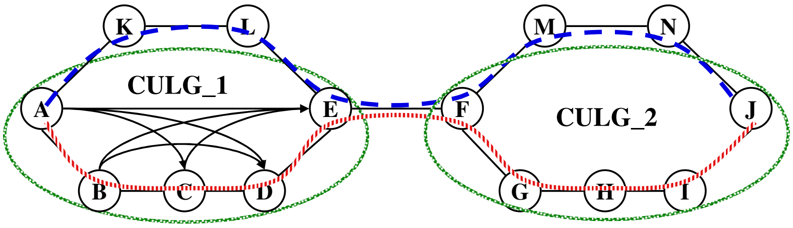


Fig. 7.6. Adding Type-2 crossover edges

Type-2 crossover edges are similar to the original type of crossover edges and are created inside a CULG. They connect the non-adjacent vertices of the CULG. For instance, the vertex A is connected to each of the vertices C, D, and E using Type-2 crossover edges. Figure 7.6 shows all the Type-2 crossover edges created inside the CULG_1.

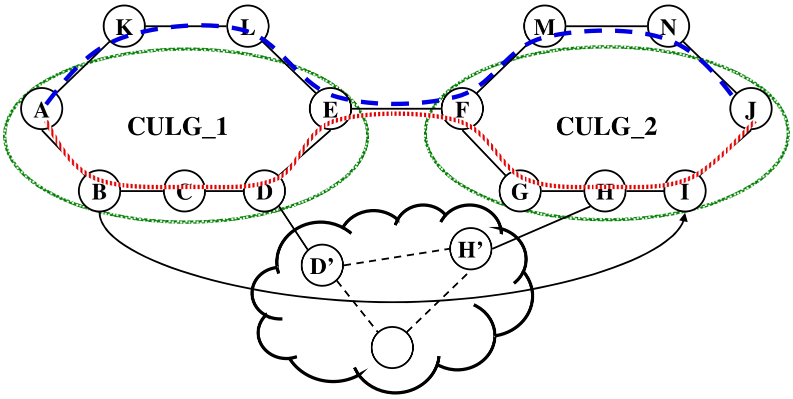


Fig. 7.7. Adding Type-3 crossover edges

Type-3 crossover edges connect vertices from different CULGs. For instance, let us consider the up-stream edge $B - C$ and the down stream edge $H - I$. In order to point out that these edges will become simultaneously free, we need to connect them by a Type-3 crossover edge. For this purpose, we assign a null weight to all the edges between the vertices C and H along the basepath if they belong to a CULG group. In our case, the edges $C - D$, $D - E$, $F - G$, and $G - H$ are assigned a null weight. Then, we compute a shortest path between the egress vertex C of the up-stream edge $B - C$ and the ingress vertex H of the down-stream edge $H - I$. Let $\wp = C - D - D' - \cdots - H' - H$ be this shortest path of weight $w_{\wp}$. Consequently, a Type-3 crossover edge is created between the ingress vertex B of the up-stream edge $B - C$ and the egress vertex I of the down-stream edge $H - I$. This edge indicates that the up-stream edge $B - C$, the edges along the path $\wp$, and the down-stream edge $H - I$ will become/are free to accept any incoming connection after rerouting the existing lightpath. Figure 7.7 illustrates this Type-3 crossover edge created in the auxiliary graph.

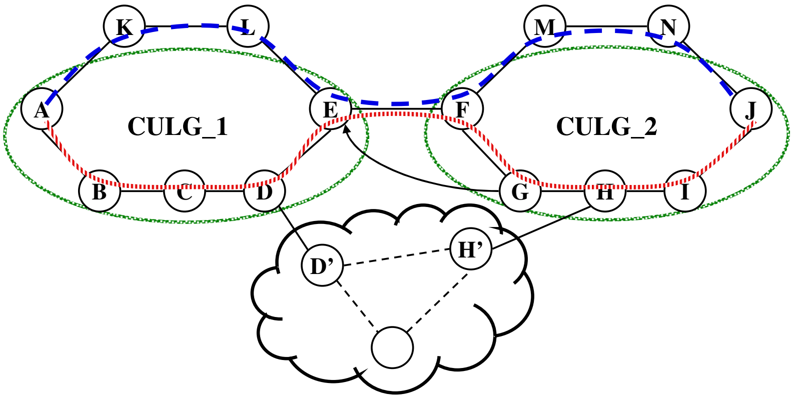


Fig. 7.8. Adding Type-1 crossover edges

In order to overcome the problem of the unidirectional case, Type-1 crossover edges are added. Such edges connect a down-stream edge to an upper-stream edge in order to indicate that they will become simultaneously free. For instance, let us consider the down-stream edge $G - H$ and the up-stream edge $D - E$. Likewise the previous case (*c.f.* Type-3 crossover edge between the vertices B and I), a shortest path is computed between the vertices H and D. Let $\wp = H - H' - \cdots - D' - D$ be this shortest path of weight $w_{\wp}$. Consequently, a Type-1 crossover edge is created between the ingress vertex G of the down-stream edge $G - H$ and the egress vertex E of the up-stream edge $D - E$. This edge indicates that the down-stream edge $G - H$, the edges along the path $\wp$, and the up-stream edge $D - E$ will become/are free to accept any incoming connection after rerouting the existing lightpath. Figure 7.8 illustrates this Type-1 crossover edge created in the auxiliary graph.

7.1.4 The Resulting Path Adjusting Rerouting (PAR) Algorithm

The proposed Path Adjusting Rerouting algorithm works in three phases and uses a graph representation of the network. Each node of the network is represented by its equivalent auxiliary graph model, and each fiber-link is represented by an equivalent edge. The first phase of the proposed PAR algorithm is the same as the normal routing phase already described in Chapter 6. Thus, when a new connection request arrives at the network, we try to satisfy it using the free resources of the network without requiring any rerouting. This is carried out by finding the shortest path between its source node and its destination node using a conventional shortest path algorithm (like Dijkstra's algorithm) combined with an appropriate dynamic weight assignment scheme. If the first phase of our algorithm fails to find a shortest path with a finite cost, the second phase of our algorithm is executed. The goal of this second phase is to relax the constraints on the network by trying to reduce and eliminate bottlenecks. The edges that constitute a bottleneck in the network are those with an infinite weight. This second phase of the algorithm proceeds in three stages.

In stage 1, we identify all the lightpaths/connections that contribute to form bottlenecks, and we check if new alternate paths can be provided to them. For this task, we first identify all the edges that constitute a bottleneck in the network. Then, we check if each lightpath/connection that passes through such edges can be rerouted onto another path. This is done as follows:

1. The considered lightpath/connection is removed from the network, and the resources that it uses are restituted to the network.
2. The considered edge identified as a bottleneck is assigned an infinite weight.
3. K-alternate shortest paths are computed between the source node and the destination node of the considered lightpath/connection. Such paths represent alternative solutions to route the considered lightpath/connection without passing through the edge identified as a bottleneck. The shortest paths that have finite weights/costs constitute the set of alternate paths. The cost of rerouting a lightpath/connection from the original basepath to a given alternate path is equal to M times the weight of this alternate path.

If the lightpath/connection passes through two or more edges identified as bottlenecks, additional sets of K-shortest paths must be computed. These K-shortest paths must not use any of the considered edges with infinite cost.

After identifying the lightpaths/connections that can be rerouted, we modify, in a second stage, the network auxiliary graph by creating crossover edges for every reroutable lightpath. Type-2 and Type-3 crossover edges are used to extend the wavelength retuning rerouting scheme to the path adjusting rerouting scheme (*c.f.* Section 7.1.3). In addition, as our auxiliary graph is made of unidirectional edges and as the connection requests are asymmetric, Type-1 crossover edges are also used (*c.f.* Section 7.1.2).

Once the auxiliary graph is constructed and the weights are assigned to all the edges, the shortest path is computed using a conventional shortest path algorithm (stage 3). If no path with a finite weight can be found, the connection request is rejected; otherwise the new connection can be established by rerouting some existing lightpaths. The lightpaths to be rerouted are determined by the crossover edges selected by the shortest path algorithm. Once the lightpaths are rerouted, the normal routing procedure is initiated again as described in phase 1. The aim of this third phase is to avoid any resource conflict between the resources needed for lightpath rerouting and those needed to route the new incoming request. The result of this last phase is a new route for the incoming request between its source node and its destination node. If this routing phase fails to find a route with a finite cost, the incoming request is rejected.

The PAR algorithm can be modified to run in two phases. In the first phase, we assign a weight to each edge in the network auxiliary graph. From these weights, we can identify all the edges that constitute a bottleneck in the network (edges with infinite weight). For each lightpath/connection passing through such overloaded edges, we try to find alternate paths as described previously. For each reroutable lightpath/connection and its corresponding set of feasible alternative paths, we modify the network auxiliary graph accordingly by creating crossover edges. Finally, we compute the shortest path between the source node of the incoming request and its destination node. If no path with a finite weight can be found, the incoming connection request is rejected. If the weight of the shortest path is smaller than M, the new connection can be established without rerouting any existing lightpaths. Otherwise, if the cost of the shortest path is larger than M, some lightpaths must be rerouted. Once the selected lightpaths are rerouted, the second phase is initiated to find a new route for the incoming request. This last phase consists in reassigning weights to the edges of the network auxiliary graph and then computing a new route using a shortest path algorithm. If this routing phase fails, the incoming request is rejected.

Because searching for alternate paths and creating crossover edges are difficult and time consuming tasks, we chose to use the first approach in order to keep the complexity at an acceptable level. Thus, we proceed to reroute some existing lightpaths/connections only when the incoming request cannot be satisfied using only free resources of the network. Additional details are provided by the pseudo-code given by Algorithm 7.1.

7.2 Time Limited Resource Reservation (TLRR) Algorithm

7.2.1 Motivation

When no preplanned demands are already routed on the network, there is not any network resources that are reserved for future use. In general, the decision on accepting or rejecting an incoming request is mainly driven by the maximum load, and consequently by the free capacity of the network evaluated during the known/estimated life duration of the incoming request. In our case, this maximum load is not other than the actual load of the network. Indeed, as the requests are treated sequentially, the network has only the knowledge of past and current requests. These requests, that are currently occupying the network, cannot ask for additional resources, but they will free their occupied resources once they end. Thus, the load carried by a network edge can only decrease with time. Hence, the maximum load of a network edge evaluated during the life duration of a request is independent of this life duration since it is equal to the load carried by this edge at the instant of arrival of the request. This explains the close performance behavior of sRxDs routing and of RxDs routing in the absence of pre-routed scheduled demands and for the same set of random demands.

However, the impact of the unpredictable life duration of the RxDs is best outlined when the network carries a mix of scheduled traffic and of random traffic. In this context, searching on the fly for free network resources is not trivial. Indeed, the maximum load of a network edge evaluated during the life duration of a random request does not depend only on the resources occupied by past and current requests, but also on the resources reserved to be used in the future by some scheduled demands. Thus, the maximum load of an edge, and consequently its free capacity, is tightly related to the life duration of the incoming request. The larger this life duration, the more probable the set-up of a new scheduled demand during this life duration, and the higher the expected network load. As the life duration of the RxDs is unknown, one assumes in the worst case scenario that this life duration is infinite. According to this assumption, the free capacity of an edge evaluated during the life duration of this RxD is smaller than the case when this request could be characterized by a predictable life-duration (as it is the case for sRxDs). This explains the higher rejection ratio encountered by the RxDs compared to the rejection ratio encountered by the sRxDs when the network carry a set of pre-routed scheduled demands.

Algorithm 7.1 Path Adjusting Rerouting Algorithm

```
Input: A network represented by its equivalent auxilliary graph $G'(V', E')$
       A set $\mathcal{D}_1$ of pre-routed $SxDs$
       A set $\mathcal{D}_2$ of $N$ random requests $sRXDs/RxDs$ to be routed
Output: The path assigned to each accepted random request and the rejection ratio $\varpi$

Begin
Step 1. Initializing
1      Route the $SxDs$ by reserving their required resources during their active period.
2      Update the state of the network auxiliary graph.
3      $Blocked := 0$

       for every incoming request $\delta_i\ (s_i, d_i, \alpha_i, \beta_i, n_i) \in \mathcal{D}_2$ do
Step 2. Normal Routing Phase
4      Check if some already routed $sRxDs/RxDs$ have ended.
       If so, free the resources assigned to them and update the network auxiliary graph.
5      Evaluate the free capacity of the network during the life duration of the incoming request $\delta_i$.
       (i.e., during $[\alpha_i, \beta_i]$ for $sRxDs$ and during $[\alpha_i, \infty]$ for $RxDs$)
6      Assign appropriate weights for the edges of the network auxiliary graph. (c.f. Sections 6.3 and 6.4 for
       $sRxDs$ and Sections 6.5 and 6.6 for $RxDs$)
7      Compute the shortest path between node $s_i$ and node $d_i$ on the subsequent graph.
       Let $\wp$ be this path and $w_\wp$ be its weight.
8      if $w_\wp < \infty$ then
8.1        Route the incoming request $\delta_i$ along the path $\wp$ by reserving the required resources during its life
           duration. (i.e., during $[\alpha_i, \beta_i]$ for $sRxDs$ and during $[\alpha_i, \infty]$ for $RxDs$)
8.2        Update the state of the network auxiliary graph.
8.3        Wait for a new random request.
       else
8.4        goto Step 3.
       endif
Step 3. Rerouting Phase
9      Save the current network state.
10     $\mathcal{R} := \emptyset$ (* set of reroutable requests *)
Step 3.1. Check for reroutable requests
11     $\mathcal{J} = \{e_i \in E' / w_{e_i} = \infty\}$ (* set of all the network edges with infinite weight *)
12     for each subset $\mathcal{S}$ $(\mathcal{S} \subset \mathcal{J}, \mathcal{S} \neq \emptyset)$ do
12.1       $\mathcal{C} = \{$already routed $\delta_j\ (s_j, d_j, \alpha_j, \beta_j, n_j) \in \mathcal{D}_2$ such that its associated path $\wp_j$ passes through all
           the edges of $\mathcal{S}\}$
12.2       for each request $\delta_j\ (s_j, d_j, \alpha_j, \beta_j, n_j) \in \mathcal{C}$ do
12.2.1         Remove the request $\delta_j$ from the network and free its reserved resources along $\wp_j$.
12.2.2         Evaluate the new free capacity of the network during the remaining active period of $\delta_j$. (i.e., during $[\alpha_i, \beta_j]$)
12.2.3         Assign appropriate weights for the edges of the network auxiliary graph.
12.2.4         Assign infinite weights for the edges of the subset $\mathcal{S}$.
12.2.5         Compute $K$-shortest paths $(\rho_1, \cdots, \rho_K)$ between node $s_j$ and node $d_j$.
12.2.6         The paths $\rho_l$ with finite weight $w_{\rho_l}$ represent alternative solutions to reroute the request $\delta_j$
               without passing through the edges identified as bottleneck.
               $\mathcal{R} = \mathcal{R} + \{(\delta_j, \wp_j, \rho_l, w_{\rho_l})\}$
12.2.7         Restore the stored current network state.
           endfor
       endfor
Step 3.2. Create crossover edges
13     for each $(\delta_j, \wp_j, \rho_l, w_{\rho_l}) \in \mathcal{R}$ do
13.1       Create Type-1, Type-2, and Type-3 crossover edges. The cost of rerouting $\delta_j$ from its basepath $\wp_j$
           to the alternate path $\rho_l$ is equal to $M \times w_{\rho_l}$.
       endfor
Step 3.3. Reroute some existing requests
14     Compute the shortest path between node $s_i$ and node $d_i$ on the subsequent graph.
       Let $\wp$ be this path and $w_\wp$ be its weight.
15     if $w_\wp < \infty$ then
15.1       Reroute the requests corresponding to the crossover edges selected by the path $\wp$.
15.2       Delete the crossover edges and update the state of the network auxiliary graph.
15.3       goto Step 4.
       else
15.4       Delete the crossover edges.
15.5       Block the incoming request $\delta_i$. $Blocked := Blocked + 1$
15.6       Wait for a new random request.
       endif
```

../..

../..

```
Step 4. Second Routing Phase
16   Reevaluate the free capacity of the network during the life duration of the incoming request $\delta_i$.
17   Assign appropriate weights for the edges of the network auxiliary graph.
18   Compute the shortest path between node $s_i$ and node $d_i$ on the subsequent graph.
     Let $\wp$ be this path and $w_\wp$ be its weight.
19   if $w_\wp < \infty$ then
19.1     Route the incoming request $\delta_i$ along the path $\wp$ by reserving the required resources during its life
         duration.
19.2     Update the state of the network auxiliary graph.
19.3     Wait for a new random request.
     else
19.4     Block the incoming request $\delta_i$. $Blocked := Blocked + 1$
19.5     Wait for a new random request.
     endif

     At the end of the simulation
20   return the rejection ratio $\varpi := Blocked/N$.
End.
```

Example 7.5.

Let us consider a network where we focus particularly on a given fiber-link with two free wavelengths. All the traffic requests that are considered in this example require a full optical channel capacity.

As a first approach, we do not consider any set of pre-established scheduled demands. Let us suppose that an RxD d_1 uses this fiber-link starting from instant t_0. A new RxD d_2 arrives at instant t_1 $(t_1 > t_0)$ and needs to be routed. Whether this request d_2 could be characterized by a finite or an infinite life duration, the free capacity of the fiber-link evaluated during the life duration of this request is equal to its free capacity evaluated at the instant of arrival t_1 of this request (*c.f.* Figure 7.9). According to the network state at t_1, the available wavelength of this fiber-link can be used to satisfy the new RxD d_2.

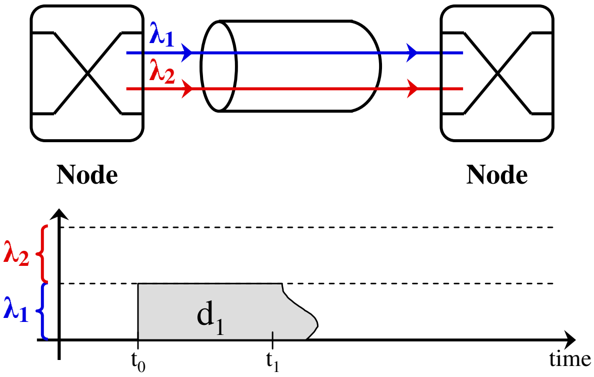


Fig. 7.9. Example of RxDs routing without pre-established scheduled demands

As a second approach, we consider a set of pre-established scheduled demands where an SxD D_1 is scheduled to use this fiber-link starting from instant t_2. We still suppose that the RxD d_1 uses this same fiber-link starting from instant t_0 $(t_0 < t_2)$. According to the network state at t_1 $(t_0 < t_1 < t_2)$, the new RxD d_2 arriving at t_1 can use the available wavelength of this fiber-link (*c.f.* Figure 7.10). This assumption is true if either d_1 or d_2

ends before the instant t_2, but it results in a conflict if both d_1 and d_2 remain active after t_2. By convention, an SxD such as D_1 must benefit from its guaranteed resources. In the worst case scenario where the RxDs d_1 and d_2 may have an infinite life duration, the free capacity of the fiber-link evaluated during the life duration of the incoming request d_2 is null. Thus, the request d_2 must be rejected at its instant of arrival t_1 because of the lack of available resources starting from t_2.

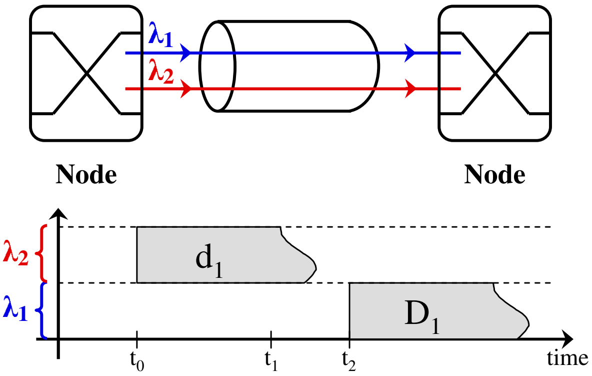


Fig. 7.10. Example of RxDs routing with pre-established scheduled demands

7.2.2 Principle

Since we know the future evolution of the network load due to the knowledge of pre-established connections, we propose to guarantee the availability of network resources for RxDs over a given period of time $\mathit{\Delta}$. The life duration of an RxD being unknown, its extinction date may occur during this period $\mathit{\Delta}$ or later. In this last case, a rerouting procedure may be necessary to ensure the RxDs' survivability. The originality of our approach consists in providing a time-limited guaranteed service to RxD requests instead of a pure best effort service, as is the case in general.

In fact, when a new RxD request $\delta_i(s_i, d_i, \alpha_i, \beta_i =?, n_i)$ arrives at the network, we assume that its life duration is less than or equal to a given period $\mathit{\Delta}$ ($\beta_i \leqslant \alpha_i + \mathit{\Delta}$). At this stage, we can route this request on the network as if it was an sRxD knowing its source node, its destination node, its data rate, and its predicted ending time. If the network lacks of free resources during its assumed life duration $\mathit{\Delta}$, the incoming RxD request δ_i is blocked at its instant of arrival α_i; otherwise, it is routed using the resources that remain available during the period $[\alpha_i, \alpha_i + \mathit{\Delta}]$. If the incoming RxD is still active at the end of this period $\mathit{\Delta}$, the connection request will continue to occupy its specified resources until it ends or until it is preempted by an SxD. Let us recall that SxDs are QoS-guaranteed. In this latter case (*i.e.,* preempted by an SxD), we try to satisfy the RxD δ_i using a different path during a new time period $\mathit{\Delta}$. It may happen that this rerouting scheme is not possible due to the lack of network resources during this new time period $\mathit{\Delta}$. Consequently, the RxD is interrupted. On the other hand, if this rerouting is possible, the RxD δ_i is satisfied using a new path for a new period $\mathit{\Delta}$. This rerouting scheme based on the transient reservation window $\mathit{\Delta}$ is repeated until the end of the demand or until its interruption.

During the simulation, several parameters are to be evaluated:

1. the cumulative number ϖ_1 of RxDs that are blocked at their instant of arrival.

2. the cumulative number ϖ_2 of RxDs that are initially accepted for a period $\varDelta$ but are blocked beyond $\varDelta$ due to the lack of network resources. ϖ_2 represents also the number of times the rerouting procedure is initiated without succeeding to find a new route for the RxD.
3. the number ϱ of successful rerouting.

Based on these parameters, two metrics are defined in order to evaluate the performance of our rerouting scheme: the rejection ratio ϖ and the number of rerouting procedures ϑ. If N is the total number of RxDs, ϖ and ϑ are computed as follows:

$$\varpi = \frac{\varpi_1 + \varpi_2}{N} \tag{7.1a}$$

$$\vartheta = \varpi_2 + \varrho \tag{7.1b}$$

For additional details on the TLRR algorithm, please refer to the Example 7.6 and to the pseudo-code given by Algorithm 7.2.

Example 7.6.

Let us consider a network where we focus particularly on a given fiber-link with two free wavelengths $\{\lambda_1, \lambda_2\}$. All the traffic requests that are considered in this example require a full optical channel capacity. We consider also a set of pre-established scheduled demands where two SxDs D_1 and D_2 are scheduled to use this fiber-link; D_1 is scheduled to use this fiber-link from t_1 to t_3 on wavelength λ_1, while D_2 is scheduled to use this fiber-link from t_2 to t_4 on wavelength λ_2 ($t_1 < t_2 < t_3 < t_4$) (*c.f.* Figure 7.11). A new RxD request d_1 arrives at the network at t_0 ($t_0 < t_1$) and needs to be routed. We assume that the tear-down date t_5 of d_1 is unknown at its instant of arrival t_0.

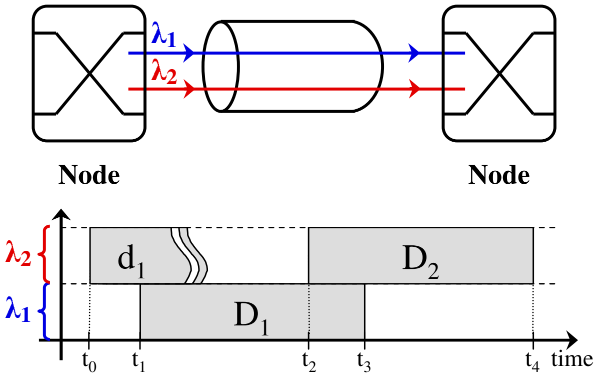


Fig. 7.11. Example of TLRR rerouting algorithm

Depending on the relative value of t_0, t_2, t_5, and the transient reservation window $\varDelta$, we can distinguish different scenarios:

- If $t_0 + \varDelta > t_2$: the fiber-link cannot carry the incoming RxD d_1 even when $t_5 < t_2$. If there isn't any other free resources in the network, the request d_1 is blocked at its instant of arrival.
- If $t_0 + \varDelta < t_2$: the fiber-link can carry the incoming RxD d_1 for a period $\varDelta$. For this task, we reserve this fiber-link during the time interval $[t_0, t_0 + \varDelta]$.
    - If $t_5 < t_0 + \varDelta$: the request d_1 ends during the period $\varDelta$, and it frees its occupied resources.

- ○ If $t_0 + \mathit{\Delta} < t_5 < t_2$: the request d_1 is still active at $t_0 + \mathit{\Delta}$, and it continues to occupy its specified resources and specifically its resources on this fiber-link. Once it ends at instant t_5, it frees its occupied resources.
- ○ If $t_0 + \mathit{\Delta} < t_2 < t_5$: the request d_1 is still active at $t_0 + \mathit{\Delta}$, and it continues to occupy its specified resources and specifically its resources on this fiber-link. At instant t_2, the RxD d_1 is preempted by the SxD request D_2. This can happen since SxDs have a higher priority than RxDs. In this case, the TLRR algorithm tries to reroute the request d_1 on a new path for a new period $\mathit{\Delta}$. If such a new path exists in the network, the request is satisfied by reserving the resources along this new path during the period $[t_2, t_2 + \mathit{\Delta}]$. Conversely, if there is not enough free resources in the network, the request d_1 is definitively interrupted at instant t_2.

It is to be noted that every time the RxD is interrupted by an SxD, the TLRR algorithm tries to satisfy the RxD using a new path for a new period $\mathit{\Delta}$ until the RxD ends or until it is definitively interrupted.

7.2.3 Additional Issues

• In practice and since we know the SxDs' activity period, one can initiate the rerouting process by anticipating any RxD preemption. In this way, we reduce the disruption period due to the rerouting process. The impact of this anticipation period is out of the scope of our study.

• When an RxD is interrupted by an SxD, the basic TLRR algorithm searches the network for a new path that remains available during a period $\mathit{\Delta}$. One can choose for the rerouting phase a transient reservation window $\mathit{\Delta}'$ that is smaller than the initial transient reservation window $\mathit{\Delta}$ of the normal routing phase. The impact of the parameters $\mathit{\Delta}$ and $\mathit{\Delta}'$ on network performance can be evaluated.

• Since the RxDs are characterized by an unpredictable life duration, the main aim of the transient reservation window is to estimate a value for this life duration. Instead of choosing a fixed value for $\mathit{\Delta}$, one may use an adaptive one. The initial value of $\mathit{\Delta}$ is set to a small value. As the simulation goes on, more and more RxDs are first routed on the network, then torn down when they ended. By computing the average life duration of the RxDs already terminated, one can have an accurate idea on the life duration of the RxDs in the network. Hence, each time a new RxD is terminated, we recompute the average life duration of the RxDs already terminated, and we update the value of the adaptive transient reservation window $\mathit{\Delta}$. In the following, we refer to this approach as "Adaptive Time Limited Resource Reservation" (ATLRR).

7.3 Simulation Results and Analysis

For the following simulations, we suppose that the network is already dimensioned using the SA algorithm to satisfy the set of 2190 SLD requests inherent to the traffic set SxD_7 (*c.f.* Section 3.6). The cost of such a network is given in Table 7.1. Once the network is dimensioned, the SLDs are routed by reserving their required resources during their active period. Subsequently, the network can be over-dimensioned by providing additional resources in order to route random traffic demands. In our case study, the parameters $\ddot{\alpha}$, $\ddot{\beta}$, and $\ddot{\gamma}$ of the over-dimensioning procedure are set to 15, 15, and 25, respectively. These values are chosen in order to prevent any sRxD/RxD blocking due to the lack

Algorithm 7.2 Time Limited Resource Reservation Algorithm

```
Input: A network represented by its equivalent auxilliary graph $G'(V', E')$
       A set $\mathcal{D}_1$ of pre-routed $SxDs$
       A set $\mathcal{D}_2$ of $N$ random requests $RxDs$ to be routed
       The transient reservation window $\varDelta$
Output: The path assigned to each accepted random request, the rejection ratio $\varpi$, and the number of
        rerouting procedures $\vartheta$

Begin
Step 1. Initializing
1        Route the $SxDs$ by reserving their required resources during their active period.
2        Update the state of the network auxiliary graph.
3        for each $RxD$ $\delta_i$ $(s_i, d_i, \alpha_i, \beta_i, n_i) \in \mathcal{D}_2$ do
3.1          Schedule a "RxD_Arrival_Event($\delta_i$)" event at instant $\alpha_i$.
3.2          Schedule a "RxD_Departure_Event($\delta_i$)" event at instant $\beta_i$.
         endfor
4        $Blocked := 0$
5        $\vartheta := 0$

Step 2. Events Handling
6        for every occurrence of a new event "New_Event" do
6.1          case "New_Event" of
               "RxD_Arrival_Event" :
                  (* Arrival of a new $RxD$ $\delta_i$ $(s_i, d_i, \alpha_i, \beta_i, n_i)$ *)
6.1.1             Evaluate the free capacity of the network during the period $[\alpha_i, \alpha_i + \varDelta]$.
6.1.2             Assign appropriate weights for the edges of the network auxiliary graph. (c.f. Sections 6.3 and
                  6.4) (* The tear-down date of the $RxD$ $\delta_i$ is assumed to be $\alpha_i + \varDelta$ *)
6.1.3             Compute the shortest path between node $s_i$ and node $d_i$ on the subsequent graph.
                  Let $\wp$ be this path and $w_\wp$ be its weight.
6.1.4             if $w_\wp < \infty$ then
6.1.4.1               Route the incoming request $\delta_i$ along the path $\wp$ by reserving the required resources during
                      $[\alpha_i, \alpha_i + \varDelta]$.
6.1.4.2               Update the state of the network auxiliary graph.
6.1.4.3               Schedule a "RxD_FalseDeparture_Event($\delta_i$)" event at instant $\alpha_i + \varDelta$.
6.1.4.4               Wait for a new event.
                  else
6.1.4.5               Block the incoming request $\delta_i$. $Blocked := Blocked + 1$
6.1.4.6               Delete the "RxD_Departure_Event($\delta_i$)" event.
6.1.4.7               Wait for a new event.
                  endif
               "RxD_FalseDeparture_Event" :
                  (* A period $\varDelta$ has passed since the arrival or the rerouting of the $RxD$ $\delta_i$, and the request is
                  still active. *)
6.1.5             Check the status of the resources currently occupied by the $RxD$ $\delta_i$.
6.1.6             if they are FREE then
                      (* Extend the reservation of the resources *)
6.1.6.1               Reserve the same resources currently occupied by the $RxD$ $\delta_i$ till the arrival of the next
                      $SxD$ $\delta_j$ $(s_j, d_j, \alpha_j, \beta_j, n_j)$. (* Only $SxDs$ can preempt $RxD$ requests. *)
6.1.6.2               Update the state of the network auxiliary graph.
6.1.6.3               Schedule a new "RxD_FalseDeparture_Event($\delta_i$)" event at instant $\alpha_j$.
6.1.6.4               Wait for a new event.
                  else
                      (* Reroute the request *)
6.1.6.5               $\vartheta := \vartheta + 1$
6.1.6.6               Evaluate the free capacity of the network during a new period of length $\varDelta$.
6.1.6.7               Assign appropriate weights for the edges of the network auxiliary graph.
6.1.6.8               Compute the shortest path between node $s_i$ and node $d_i$ on the subsequent graph.
                      Let $\wp$ be this path and $w_\wp$ be its weight.
6.1.6.9               if $w_\wp < \infty$ then
6.1.6.9.1                 Route the request $\delta_i$ along the path $\wp$ by reserving the required resources during the next
                          period of length $\varDelta$.
6.1.6.9.2                 Update the state of the network auxiliary graph.
6.1.6.9.3                 Schedule a new "RxD_FalseDeparture_Event($\delta_i$)" event after a period $\varDelta$.
6.1.6.9.4                 Wait for a new event.
                      else
6.1.6.9.5                 Block the request $\delta_i$. $Blocked := Blocked + 1$
6.1.6.9.6                 Delete the "RxD_Departure_Event($\delta_i$)" event.
6.1.6.9.7                 Wait for a new event.
                      endif
                  endif
                                                                                                   ../..
```

../..

```
            "RxD_Departure_Event" :
              (* Departure of the $RxD$ $\delta_i$ $(s_i, d_i, \alpha_i, \beta_i, n_i)$ *)
6.1.7         Free the network resources still reserved for the request $\delta_i$.
6.1.8         Update the network auxiliary graph.
6.1.9         Delete the "RxD_FalseDeparture_Event($\delta_i$)" event.
6.1.10        Wait for a new event.
          endcase
      endfor

7   return  the rejection ratio $\varpi := Blocked/N$.
8   return  the number of rerouting procedures $\vartheta$.
End.
```

of electrical ports. Thus, connection blocking will only be due to the lack of o_1 optical ports. The number of o_1-ports is over-dimensioned according to a parameter $\ddot{\eta}$ chosen in the range $0\% - 20\%$.

Table 7.1. Routing cost of the 2190 SLDs inherent to the set SxD_7

SLDs' Routing Cost								
SA Algorithm K=4	o_1-ports	8404	e_1-ports	-	r_1-ports	-	Overall Cost	19654
	o_2-ports	1875	e_2-ports	944	r_2-ports	931	Network Congestion	136
	o_3-ports	-	e_3-ports	-	r_3-ports	-	Computation Time	5600 sec

We try to satisfy on the fly the requests of the following traffic sets (*c.f.* Section 3.6):

- The set $sRxD_1/RxD_1$ of 2659 sRLD/RLD requests.
- Three set of sRED/RED requests; namely $sRxD_2/RxD_2$, $sRxD_4/RxD_4$, and $sRxD_5/RxD_5$. These sets are characterized by the same average traffic load but with different burstiness levels. The set $sRxD_2/RxD_2$ is characterized by the highest burstiness level, while the set $sRxD_4/RxD_4$ is characterized by the lowest one.
- The set $sRxD_3/RxD_3$ of 10817 traffic requests. This set is the mix of the set $sRxD_1/RxD_1$ of 2659 sRLD/RLD requests and the set $sRxD_2/RxD_2$ of 8158 sRED/RED requests with the highest burstiness level.

At this stage, the sRxDs/RxDs can be routed sequentially using the Dijkstra algorithm combined with the appropriate weight assignment schemes introduced in Chapter 6. As a first step, we consider the sRxDs with known life duration, and we route them using the weight assignment schemes described in Section 6.3 and Section 6.4. As a second step, we consider the same set of demands but we neglect the information referring to their life duration (RxDs). In other terms, these same sets of demands may be routed using the weight assignment schemes described in Section 6.5 and Section 6.6. In addition, we implement both the PAR and the TLRR rerouting strategies. Under rerouting strategies, two metrics are considered: the demands rejection ratio ϖ and the amount of required rerouting operations ϑ.

It is to be noted that the CPU time needed to determine a routing/rerouting path for a random request exceeds rarely 20 ms depending on the network load. This time was evaluated on a 3 GHz Intel Pentium IV processor with 1 Giga Bytes of RAM memory. Assuming that the time needed to establish a connection is of several hundreds of milliseconds including the signalling stage as well as the path determining stage, we assume that enough time remains available for the signalling process.

7.3.1 Performance of the PAR Algorithm

Performance of the PAR Rerouting Scheme under sRLD Requests

When a new sRLD enters the network, it asks for network resources for a given period of time. Knowing its life duration, the incoming sRLD can be routed using the Dijkstra algorithm combined with the sRLDs dynamic weight assignment scheme (*c.f.* Section 6.3). However, when the network lacks of free resources, some existing connections may be rerouted in order to satisfy the incoming request. Assuming pre-routed SLDs, Figure 7.12 illustrates the rejection ratio of the sRLD requests when no rerouting is performed and when the PAR scheme is enabled. In average, 2% additional sRLDs can be satisfied using the PAR scheme.

Performance of the PAR Rerouting Scheme under RLD Requests

The RLDs are characterized by an unknown life duration at their instant of arrival. Thus, they can be routed using the Dijkstra algorithm combined with the dynamic weight assignment scheme developed for RLDs (*c.f.* Section 6.5). However, when the network lacks of free resources, some existing connections may be rerouted in order to satisfy the incoming request. Assuming pre-routed SLDs, Figure 7.13 illustrates the rejection ratio of the RLD requests when no rerouting is performed and when the PAR scheme is enabled. The PAR algorithm proofs its performance to alleviate the constraint due to the unknown life duration of the RLDs. In average, 5% additional RLDs can be satisfied using the PAR scheme.

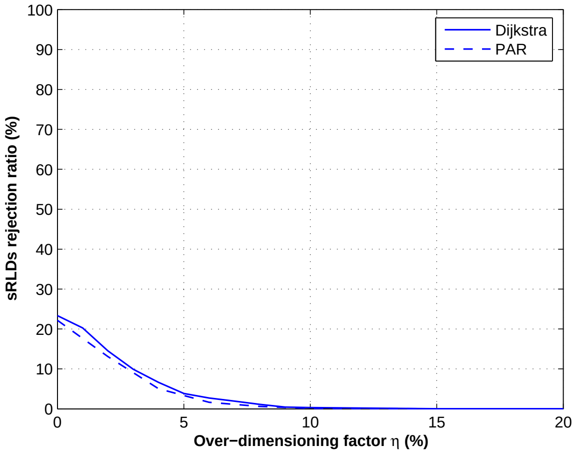


Fig. 7.12. sRLDs rejection ratio using PAR rerouting scheme

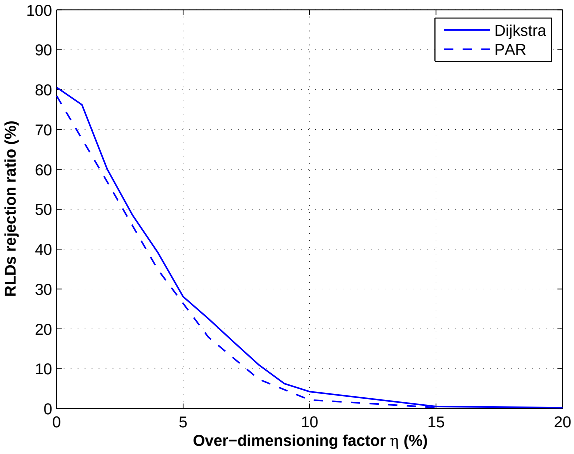


Fig. 7.13. RLDs rejection ratio using PAR rerouting scheme

Reliability of the Plots

In order to check the reliability of the plots of the previous simulations, we introduce the concept of the Confidence Interval (CI). In statistics, a CI is an interval estimate of a population parameter. In other words, instead of estimating the parameter by a single value, a range of plausible values for the unknown parameter is provided. The width of the confidence interval gives us an idea on the incertitude/precision of the unknown parameter. A very wide interval may indicate that more data should be collected before stating anything regarding the estimated parameter.

In a narrower sense, a CI for a population parameter is an interval with an associated confidence level p. This CI is computed from a set of random data samples taken from the population. If the sampling is repeated several times and the CI is recalculated from each set, p % of the calculated CIs would contain the population parameter in question.

For instance, we consider a set of n random samples $\{X_1, X_2, \cdots, X_n\}$ from a normally distributed population with unknown mean μ and known variance σ^2. Based on these observations, we can estimate a value $\bar{X}$ for the mean μ as follows:

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{7.2}$$

We note that the sample mean $\bar{X}$ of a normally distributed sample is also normally distributed with the same expectation μ but with a standard error equal to $\sigma/\sqrt{n}$. Consequently, the random variable $Z = \frac{\bar{X}-\mu}{\sigma/\sqrt{n}}$ is also normally distributed independent of μ with mean equal to 0 and variance equal to 1. Hence it is possible to find numbers $-z$ and z, independent of μ, where Z lies in between with probability p. p is a measure of how confident we want to be. If we take $p = 0.95$, we have:

$$P(-z \leqslant Z \leqslant z) = p = 0.95 \tag{7.3}$$

It follows from the normal distribution function that the value of z is equal to:

$$z = 1.96 \tag{7.4}$$

Thus, the endpoints of the confidence interval are given by:

$$\begin{aligned} 0.95 &= P(-z \leqslant Z \leqslant z) \\ &= P(-1.96 \leqslant \frac{\bar{X}-\mu}{\sigma/\sqrt{n}} \leqslant 1.96) \\ &= P(\bar{X} - 1.96 \times \frac{\sigma}{\sqrt{n}} \leqslant \mu \leqslant \bar{X} + 1.96 \times \frac{\sigma}{\sqrt{n}}) \end{aligned} \tag{7.5}$$

However, when the variance σ^2 is unknown and has to be estimated from the data samples X_i, we have to use the Student's t-distribution (or also t-distribution) rather than the normal distribution. In probability and statistics, the Student's t-distribution is a probability distribution that arises in the problem of estimating the mean of a normally distributed population when the sample size is small. When the sample size is large, say 100 or above, the Student's t-distribution is very similar to the standard normal distribution. However, with smaller sample sizes, the Student's t-distribution is leptokurtic, which means it has relatively more scores in its tails than does the normal distribution. As a result, we have to extend the interval endpoints farther from the mean in order to contain a given percentage of the area. We recall that with a normal distribution, 95% of the distribution is within 1.96 standard deviations of the mean. Using the Student's t-distribution and a set of 5 samples, 95% of the area is within 2.78 standard deviations of the mean.

For instance, we reconsider the same set of n random samples $\{X_1, X_2, \cdots, X_n\}$ from a normally distributed population. In this case, we assume that both the mean μ and the variance σ^2 are unknown. Thus, we can estimate a value $\bar{X}$ for the mean μ and a value $\bar{S}^2$ for the variance σ^2 as follows:

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{7.6a}$$

$$\bar{S}^2 = \frac{1}{n-1}\sum_{i=1}^{n} (X_i - \bar{X})^2 \tag{7.6b}$$

In [J144], Gosset studied the random variable $T = \frac{\bar{X}-\mu}{\bar{S}/\sqrt{n}}$. While similar to Z, the variance $\bar{S}^2$ of T is estimated. Gosset's work showed that T is a Student's t-distribution with $\nu = n-1$ degrees of freedom. The Student's t-distribution depends only on ν and not on μ nor on σ. The endpoints of the confidence interval are obtained as follows:

$$\begin{aligned} p &= P(-t \leqslant T \leqslant t) \\ &= P(-t \leqslant \frac{\bar{X}-\mu}{\bar{S}/\sqrt{n}} \leqslant t) \\ &= P(\bar{X} - t \times \frac{\bar{S}}{\sqrt{n}} \leqslant \mu \leqslant \bar{X} + t \times \frac{\bar{S}}{\sqrt{n}}) \end{aligned} \tag{7.7}$$

The Table 7.2 lists a few values of t as a function of the degree of freedom ν and the confidence level p. We note that a Student's t-distribution with a large degree of freedom ($\nu = \infty$) reduces to a standard normal distribution.

Table 7.2. Values of the inverse cumulative distribution function of the Student's t-distribution

		p				
		90%	95%	99%	99.5	99.9%
ν	5	2.015	2.571	4.032	4.773	6.869
	10	1.812	2.228	3.169	3.581	4.587
	25	1.708	2.060	2.787	3.078	3.725
	50	1.676	2.009	2.678	2.937	3.496
	100	1.660	1.984	2.626	2.871	3.390
	∞	1.645	1.960	2.576	2.807	3.291

As stated before, using a normal distribution, the confidence interval with confidence level of 95% is equal to $[\bar{X} - 1.96\ \sigma/\sqrt{n}, \bar{X} + 1.96\ \sigma/\sqrt{n}]$. Using a Student's t-distribution and a set of 51 samples ($n = 51,\ \nu = 50$), the confidence interval with confidence level of 95% is equal to $[\bar{X} - 2.009\ \bar{S}/\sqrt{n}, \bar{X} + 2.009\ \bar{S}/\sqrt{n}] = [\bar{X} - 0.2813\ \bar{S}, \bar{X} + 0.2813\ \bar{S}]$.

All the simulation carried out in this chapter are computed using 50 different sets of random traffic requests *sRxDs*/*RxDs*. For each simulation, we plot the average value of the results as well as the confidence interval with confidence level of 95% computed using the Student's t-distribution. For instance, assuming pre-routed SLDs, Figure 7.14 replots the rejection ratio of the sRLD requests and the RLD requests for different values of the over-dimensioning parameter $\ddot{\eta}$ when no rerouting is performed and when the PAR scheme is enabled. In this plot, the bars represent the average rejection ratio for a given type of requests and a given routing scheme, while the vertical lines represent the confidence intervals surrounding these rejection ratios. We are pretty sure (95% confidence level) that the confidence interval contains the real rejection ratio of the random requests.

7.3.2 Performance of the TLRR Algorithm

Impact of the Value of Δ

In this section, we investigate the impact of the value of the transient reservation window Δ on the blocking performance of the network. For this purpose, the over-dimensioning parameter $\ddot{\eta}$ has been fixed to '0', thus no additional o_1-port has been added to the network. We evaluate the rejection ratio

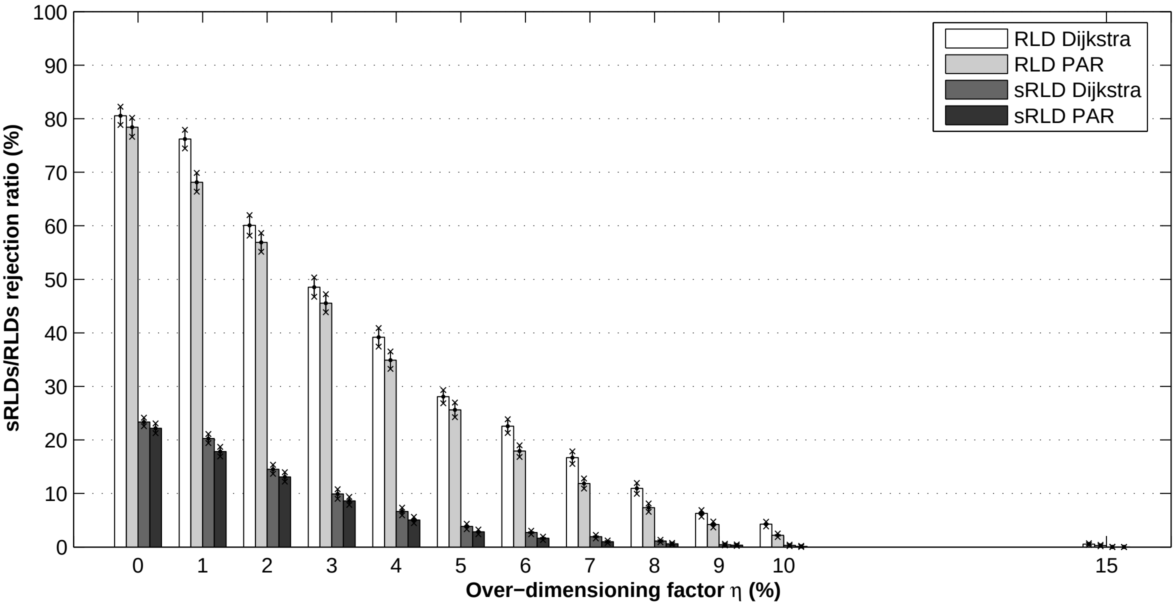
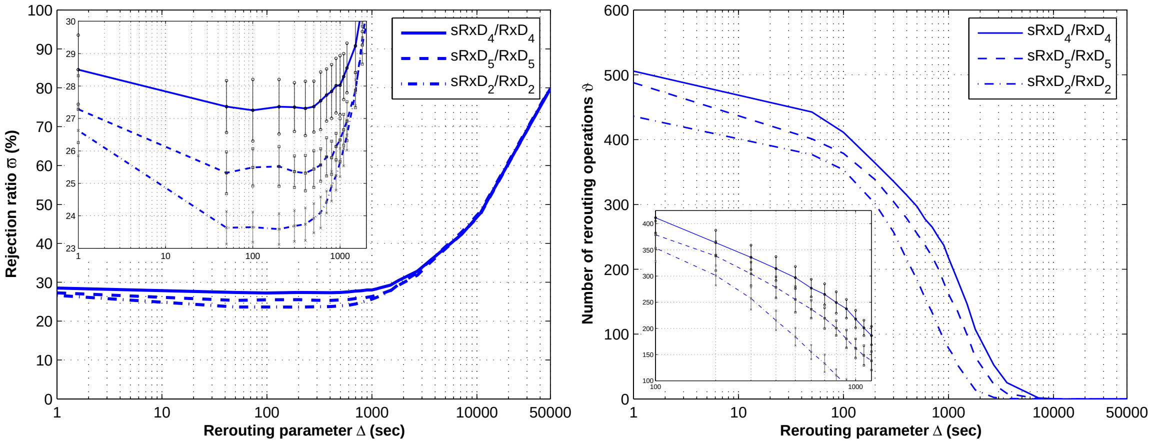


Fig. 7.14. sRLDs and RLDs rejection ratio using PAR rerouting scheme



(a) Rejection ratio ϖ

(b) Rerouting operations ϑ

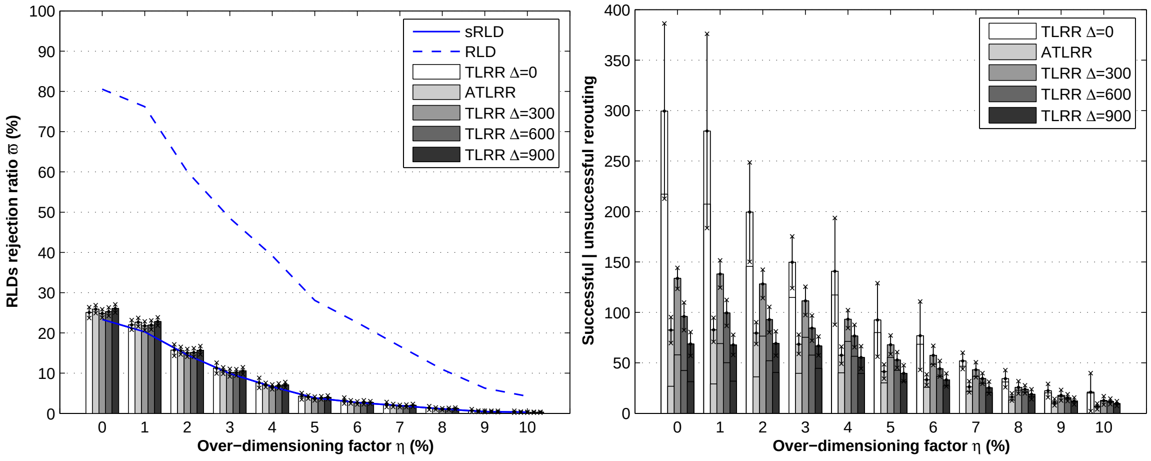
Fig. 7.15. Impact of $\varDelta$ on the network performance

ϖ (Figure 7.15(a)) as well as the number of required rerouting operations ϑ (Figure 7.15(b)) for the different sets of sREDs/REDs ($sRxD_2/RxD_2$, $sRxD_4/RxD_4$, and $sRxD_5/RxD_5$).

We conclude that traffic burstiness has a negligible impact on the rejection ratio. For large $\varDelta$, the performance of the rerouting scheme approaches that of RxD routing with infinite life duration. Conversely, for small $\varDelta$, this performance approaches that of RxD routing based only on the current network state ($\varDelta = 0$). These two results are quite intuitive. It appears that for $\varDelta$ ranging between 100 and 500 seconds, the rejection ratio reaches a minimum. The more bursty the connections, the lower this minimum. However, it appears that the least bursty connections require more frequent rerouting. It is to be noted that the higher the values of $\varDelta$, the lower the number of rerouting operations. Thus, the dimensioning of the parameter $\varDelta$ is a trade-off between the rejection ratio and the signaling complexity. For instance, for $\varDelta = 1000$ seconds, the decrease in the number of rerouting operations is noticeable compared to the quasi-stability of the rejection ratio.

Impact of $\ddot{\eta}$ on sRLDs/RLDs Network Performance

In this section, we investigate the impact of the over-dimensioning factor $\ddot{\eta}$ on sRLDs/RLDs network performance for some given values of Δ (mainly Δ= 0, 300, 600, and 900). For this purpose, we consider the set $sRxD_1/RxD_1$ composed of 2659 full-wavelength requests. For positive values of Δ, the rejection ratio is slightly lower than for $\Delta = 0$ (Figure 7.16(a)). Figure 7.16(b) shows that the number of rerouting operations decreases significantly for large values of $\ddot{\eta}$. In addition, the probability of the rerouting operation to successfully find a new path for the request to be rerouted increases as the over-dimensioning factor $\ddot{\eta}$ increases. Finally, it is to be noted that the ATLRR succeeds to have a good estimation of the value of Δ and achieves rejection ratios for RLDs comparable to those obtained for sRLDs while requiring the lowest number of rerouting operations.



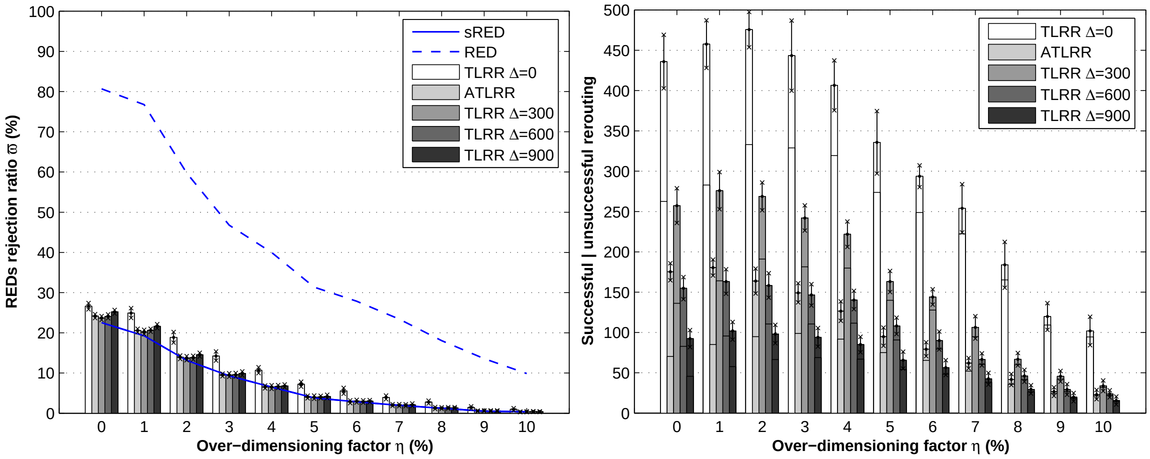
(a) Rejection ratio ϖ

(b) Rerouting operations ϑ

Fig. 7.16. Impact of η on network performance under RLD traffic flow

Impact of $\ddot{\eta}$ on sREDs/REDs Network Performance

In this section, we investigate the impact of the over-dimensioning factor $\ddot{\eta}$ on sREDs/REDs network performance for some given values of Δ (mainly Δ= 0, 300, 600, and 900). In order to focus our interest on the rejection ratio, we consider the set of random demands $sRxD_2/RxD_2$ with the highest burstiness. The results plotted in Figure 7.17 confirm what we have already noticed in Figure 7.15 and in Figure 7.16. Indeed, for positive values of Δ, the rejection ratio is slightly lower than for $\Delta = 0$ with the best rejection ratio being around $\Delta = 300$ seconds (Figure 7.17(a)). Moreover, for larger values of Δ, the number of rerouting operations is significantly reduced (Figure 7.17(b)). It is to be noted that as the over-dimensioning factor $\ddot{\eta}$ increases, the ratio of the number of unsuccessful rerouting to the number of successful rerouting decreases. In other terms, the higher $\ddot{\eta}$, the higher the percentage of successful rerouting, which is also quite an intuitive result. Finally, it is to be noted that the ATLRR succeeds to have a good estimation of the value of Δ and achieves rejection ratios for REDs comparable to those obtained for sREDs while requiring the lowest number of rerouting operations.



(a) Rejection ratio ϖ (b) Rerouting operations ϑ

Fig. 7.17. Impact of η on network performance under RED traffic flow

Impact of $\ddot{\eta}$ on sRxDs/RxDs Network Performance

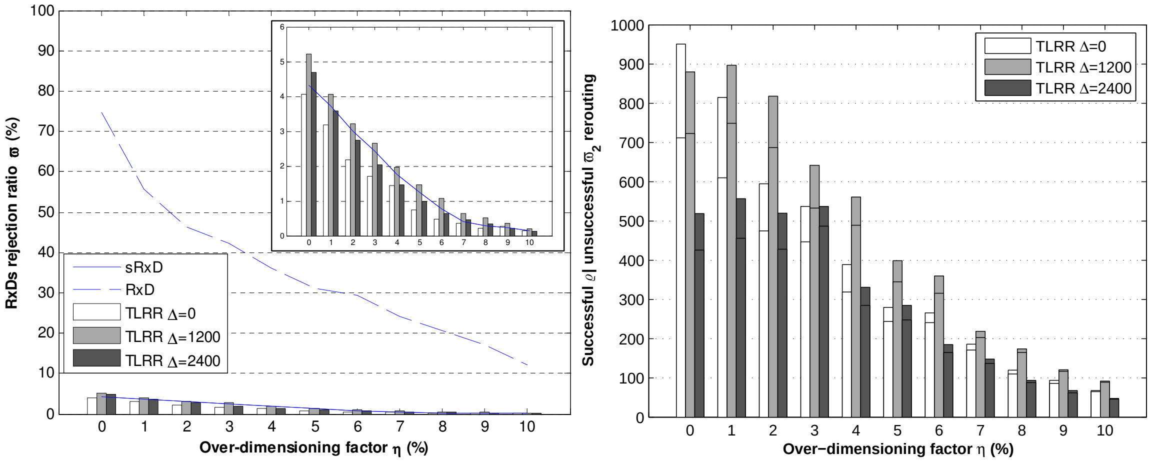
For this last scenario, one assumes that the network is dimensioned for both SLDs and SEDs of the traffic set SxD_7. The cost of such a network is given in Table 7.3. In this case, the parameters $\ddot{\alpha}$, $\ddot{\beta}$, and $\ddot{\gamma}$ of the over-dimensioning procedure are still fixed to 15, 15, and 25, respectively.

Table 7.3. Routing cost of the 2190 SLDs and SEDs of the SxD_7 traffic set

SLDs' and SEDs' Routing Cost								
SA Routing	o_1-ports	21150	e_1-ports	3519	r_1-ports	3519	Overall Cost	92742
+	o_2-ports	1875	e_2-ports	944	r_2-ports	931	Network Congestion	333
IG Grooming	o_3-ports	4192	e_3-ports	2094	r_3-ports	2098	Computation Time	8219 sec

We consider the set $sRxD_3/RxD_3$ of 10817 traffic requests. This set is the mix of the set $sRxD_1/RxD_1$ of 2659 sRLD/RLD requests and the set $sRxD_2/RxD_2$ of 8158 sRED/RED requests with the highest burstiness level. Under pre-routed SLD and SED traffic demands, the choice of the parameter $\varDelta$ is more complex. As illustrated in Figure 7.18(a), for a given value of $\ddot{\eta}$, the impact of the value of $\varDelta$ on network rejection ratio is not monotonic. For instance, the network performance in terms of rejection ratio is higher when $\varDelta = 1200$ seconds than when $\varDelta = 2400$ seconds, and it is optimum when $\varDelta = 0$. One notices that when $\varDelta$ is supposed to be infinite, the rejection ratio is always much higher than for finite values of $\varDelta$. However, it is to be noted that positive values of $\varDelta$ yield a lower number of rerouting operations (Figure 7.18(b)). In addition, for an equal number of rerouting operations, positive values of $\varDelta$ yield more successful rerouting than when $\varDelta = 0$ (for instance for $\ddot{\eta} = 3\%$). We note also that for $\varDelta = 2400$ seconds, an over-dimensioning factor $\ddot{\eta} = 7\%$ is recommended since beyond this percentage, the slope of the rejection ratio is flat.

To conclude, by introducing a rerouting strategy based on a transient reservation period $\varDelta$, it is possible to achieve rejection ratios for RxDs comparable to those obtained for sRxDs.



(a) Rejection ratio ϖ

(b) Rerouting operations ϑ

Fig. 7.18. Impact of η on network performance under a mix of RLD and RED traffic flows

8

Conclusions

In this thesis, we have investigated various problems varying from network planning to traffic engineering in multi-layer WDM networks. A key element of such a network topology is the nodal architecture, and the ability of this architecture to insert traffic into the network and to drop locally oriented traffic to the client layer. In addition, a node might perform other functions such as traffic pass-through and traffic grooming. In our study, we have adopted a rather simple and flexible node architecture where a non-blocking Electrical Cross-Connect (EXC) is coupled to a non-blocking Optical Cross-Connect (OXC). This double stage node architecture allows dynamic configuration of the add, drop, and bypass ports as well as it implements hierarchical switching and grooming functions. The optical layer, at the OXC level, is used to add, drop, and switch full wavelength data requests, while the electrical layer, at the EXC level, is used to add, drop, switch, and groom lower rate traffic requests and can perform arbitrary wavelength conversion. This multi-layer network node can be represented by means of an equivalent auxiliary graph that enables easy handling and managing traffic grooming in multi-layer WDM mesh networks. The originality of this model is that, by only manipulating the edges of the auxiliary graph constructed by the model and the weights of these edges, one can achieve various objectives using different grooming policies.

Depending on the problem being addressed, various models can be used to represent the traffic on a network. In network planning problems, deterministic static representations of the expected spatial traffic distribution are considered. Conversely, in traffic engineering problems, the traffic is usually characterized by stochastic processes that capture its dynamics and randomness. We have proposed in Chapter 3 a backbone traffic characterization well suited to dynamic circuit-switched optical WDM networks. The proposed traffic characterization captures the daily and weekly periodic distributions of the traffic flow observed on point-to-point WDM pipes in a real core network, as well as the shorter time-scale stochastic variations. This proposed characterization mainly decomposes the traffic flow into six basic traffic types. These traffic types cover the time predictability scale from the most deterministic (SxDs) to the most random (RxDs) in terms of arrival date and life duration of the requests. They also cover the different granularities of the traffic demands (xLDs and xEDs).

In Chapter 5, we have investigated the problem of efficiently designing a WDM mesh network with grooming capability for a given set of scheduled traffic requests (SxDs). In this context, cost optimization consists in considering the impact of time and space correlation between the demands as well as the impact of traffic grooming on the number of required electrical ports and optical ports. Several approaches have been proposed. First, the routing and grooming problem has been formulated as an Integer Linear Program (ILP). For small network topologies and/or limited number of requests, this mathematical formulation can be solved using some commercial softwares. The limitation of the ILP approach is that the number of variables and the number of equations increase explosively as the size of the network increases and/or the number of requests increases. The computation complexity makes it hard to be useful for networks of practical size. However, by means of efficient heuristic algorithms, it may be possible to get some results for large size networks. In our proposed heuristic approach, the network design problem is divided into two sub-problems which are solved separately. The first sub-problem routes the SLDs and the SEDs over the physical topology. Due to their deterministic aspect, the requests (SLDs and SEDs) can be routed by means of global optimization tools such as the Simulated Annealing heuristic (SA). For an instance of the SEDs routing solution, the second sub-problem deals with grooming several SEDs onto Grooming Lighpaths (GLs). The adopted strategy is an Iterative process based on a Greedy algorithm (IG). By means of numerical results, we show that the proposed heuristic approach can achieve a near-optimal solution in a significantly reduced time compared to the time needed to solve the problem by the exact approach.

Once the network is dimensioned, the SxDs (SLDs and SEDs) are routed by reserving their required resources during their active period. This is carried out by the management plane that informs each node of the availability of the various equipment (electrical ports and optical ports) within the network. At this stage, the network can be over-dimensioned by providing additional resources in order to route some random traffic demands. The control plane is in charge of routing the random demands that arrive one at a time at the network and need to be routed on the fly. The control plane operates in real time and is distributed in all the nodes of the network. We assume that GMPLS signaling, still under specification, is used to share information between the two planes.

In Chapter 6, we try to route a set of RxDs/sRxDs over the dimensioned/over-dimensioned network. Due to their random nature, RxDs/sRxDs must be routed online. More specifically, REDs/sREDs must also be aggregated on the fly. In our proposed approach, we make use of the auxiliary graph model of the network node. We associate to each edge in this auxiliary graph a property tuple reflecting the current network state. Based on this property tuple, a weight is appropriately computed and assigned to each edge. By means of the well-known Dijkstra algorithm combined with an appropriate dynamic cost assignment to each edge of the equivalent auxiliary graph, we can simultaneously and in real time determine the logical topology that consists of lightpaths, route the lightpaths over the physical topology, and finally route the traffic on the virtual topology. Simulation results outline the impact of the unpredictable life duration of RxDs. Indeed, when the network carries a set of pre-routed scheduled demands, a higher rejection ratio is encountered by the RxDs compared to the rejection ratio of the sRxDs.

In order to improve the RxDs blocking performance and the network efficiency, rerouting techniques are addressed. Two approaches are proposed in this matter: Path Adjusting Rerouting (PAR) and Time Limited Resource Reservation (TLRR). The former extends the concept of wavelength retuning scheme implementing crossover edges. The resulting PAR approach is capable of rerouting a lightpath or connection by modifying its physical route. The latter (TLRR) is quite different from traditional dynamic routing algorithms. TLRR consists in providing a time-limited guaranteed service to random demands instead of a pure best-effort service.

In this dissertation, we did not investigate the problem of the feasibility of all-optical connections (*lightpaths*) in real core networks. In such networks, as no electrical 3R regeneration (Re-amplification, Re-shaping, and Re-timing) is performed at intermediate nodes, transmission impairments induced by long-haul optical components complicate the process of path selection and wavelength assignment. Transmission impairments such as attenuation, noise (essentially due to amplifier spontaneous emission), dispersion (*e.g.,* chromatic dispersion, polarization mode dispersion), crosstalk, and nonlinear effects accumulate along the lightpath and may result in unacceptable Bit Error Rate (BER) values at the receiver's side. Consequently, the feasibility of a lightpath is no longer simply a function of the topology and the resource availability, but it also depends on the accumulation of impairments along the path. If the impairments accumulation is excessive, the BER value at the destination node may be unacceptable (exceeds a certain tolerable threshold) making the lightpath unusable for data transmission. A solution to this problem consists in placing sparse electrical regenerators along an unacceptable lightpath. The problem of the feasibility of all-optical connections is currently the subject of other studies in our research team.

Some of the original aspects developed in this thesis open to further investigations. The extension of all the techniques provided for routing and grooming of scheduled and random demands deserve to be validated in the context of Quality of Transmission (QoT) dependant network design. Undoubtedly, such an extension outlines multiple challenges. Another extension could consist in proposing a more rational dimensioning of the parameter Δ associated to the TLRR mechanism. Certainly, a relationship exists between the characteristics of the random traffic demands and the optimum value of Δ. At last, all along this thesis, we have assumed potential wavelength conversion at each node. In other terms, the wavelength assignment problem was out of the scope of our investigations. An extension of our work could consist in trying to minimize the amount of required wavelength converters in the network.

Part III

End Matter

A

Grids and Grid Networks

The Grid concept was formally developed in the mid-1990s as a shared computing approach that provides high-performance services on parallel computers and storage devices connected locally by a high-capacity Local Area Network (LAN). Nowadays, the Grid concept is enlarged to cover national and intercontinental scale networks enabling the virtualization of network computing resources.

Therefore, Grid can be defined as a flexible, distributed, and secured information technology environment that enables multiple services to be relatively independent from the specific infrastructure properties. The Grid architecture allows to present any form of information on any device at any location. Such an architecture strongly complements the era of ubiquitous digital information and services.

As the Grid architecture continues to evolve, network services are also being constantly improved. Grids are often associated with high-performance applications because of the community wherein they were originally developed. However, because Grid architecture is highly flexible, Grids have also been adopted for use by many other application areas that are less computationally intensive.

A.1 General Attributes of Grid Networks

In the following, we list the key attributes of Grid network features such as capability abstraction, resource sharing, programmability, and scalability. All these features are discussed within the Open Grid Forum (OGF) [W171], where a large community of users, developers, and vendors are leading the global standardization effort for grid computing.

A.1.1 Abstraction and Virtualization

A key feature of the Grid is its potential for abstracting capabilities. The Grid architectural model presupposes an environment wherein available modular resources can be detected, gathered, and utilized without any restriction imposed by specific low-level infrastructure implementations. The virtualization of resources is a powerful tool for creating advanced data network services. A major advantage of the virtualization of Grid network functionality through abstraction techniques is the increased flexibility in service creation, provisioning, and differentiation. It allows specific application requirements to be more directly matched with network resources. Virtualization also enables networks with different characteristics to be implemented within a common infrastructure and enables network processes and resources to be integrated directly with other types of Grid resources.

A.1.2 Resource Sharing

A primary motivation for the design and development of Grid architectures is to enhance the capability for resource sharing. Similarly, a major advantage of Grid networks is the provision of various options for resource sharing which is difficult (if not impossible) in traditional data networks. The virtualization of network resources allows for the creation of new types of data networks based on resource sharing techniques that have been recently implemented.

A.1.3 Programmability and Flexibility

An important characteristic of the Grid is that it is a programmable environment. However, until recently, Grid networks have not been programmable. This programmability provides a flexibility that is not achievable in the existing infrastructures. As noted, traditional network infrastructure has been designed to support fairly static services with fixed parameters. In the last few years, several initiatives have been established to create a Grid architecture that enables the communication services to be substantially more flexible. Using these new methods, Grid services can be provisioned as *programmable* allowing highly flexible and dynamic services and resources allocation including dynamic reconfigurations.

A.1.4 Scalability

Scalability for information technology has many dimensions. It can refer to a geographical expansion, an enhanced performance, an increase in the number of offered services, etc. Grid environments are by definition highly distributed and are, therefore, highly scalable geographically. As a result, Grid networks can extend not only across metro areas, regions, and nations but also across world-wide areas.

A.2 Types of Grids

Nowadays, many types of Grids exist, and new Grids are continually being designed to address new information technology challenges. A grid can be classified in various ways; for example, by quality of physical configuration, by topology, and by locality. Grids within an enterprise are called intra-grids, inter-linked Grids within multiple organizations are called inter-grids, and Grids external to an organization are called extra-grids. Grids can have a small or large geographical distribution, *i.e.,* distributed locally, nationally, or world-wide. Grids can also be classified by their primary resources and functions; for example, computational Grids provided for high-performance or specialized distributed computing. Grids can provide modest-scale computation by integrating computing resources across an enterprise campus or large-scale computation by integrating computers across a nation such as the TeraGrid in the United States of America (USA).

A.3 A Large-scale Grid Solution for Biological and Physical Cross-Site Simulations

In many areas of science, such as biotechnology, the simulation of true-life experiments is only achievable on high-performance computers with large scale data storage devices. Most new companies involved in this promising domain cannot afford in general such equipment. Today, a wide range of applications can benefit from Grids such as astrophysics, climate science, condensed matter, and biomolecular modeling. In such cases, the only possibility of achieving insight is to undertake *in silico* experiments using massive simulations running upon expensive high-performance computing facilities.

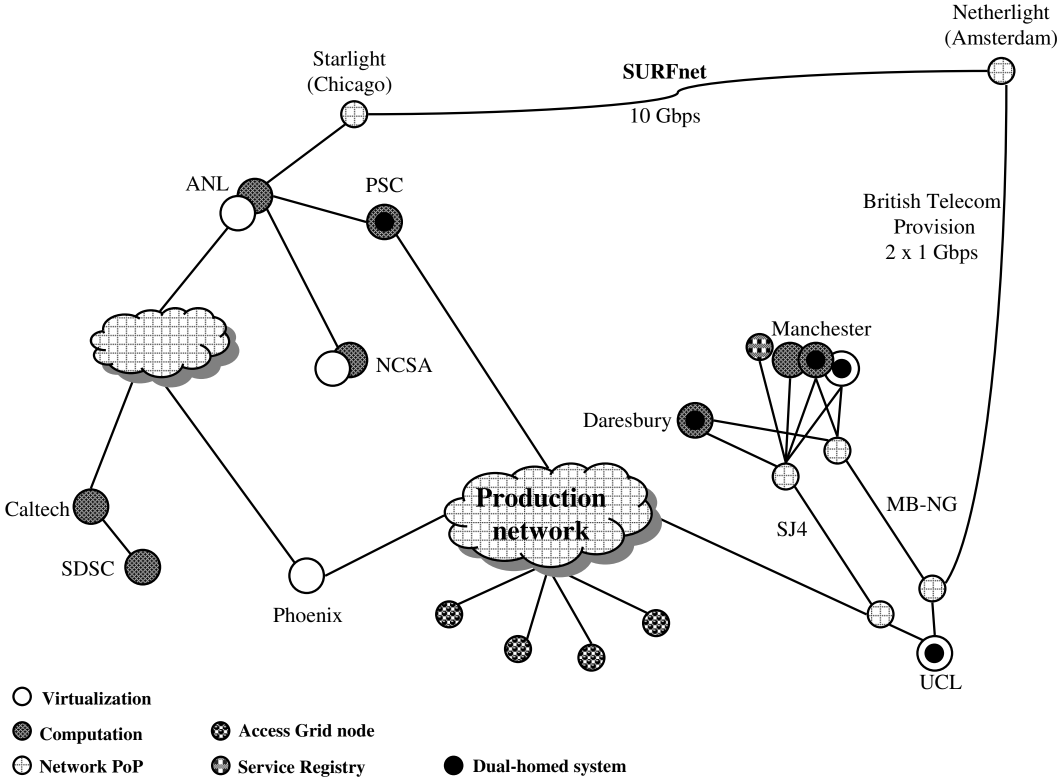


Fig. A.1. Experimental topology considered in TeraGyroid project

As an example of such experiments, we can cite the TeraGyroid [W172] project. This project coupled cutting-edge grid technologies, high-performance computing, visualization, and computational steering capabilities to produce a major leap forward in soft condensed matter simulation. This ambitious project is the result of an international collaboration linking the USA's TeraGrid [W173] and the UK's eScience Grid [W174], jointly funded by the National Science Foundation (NSF) and the Engineering and Physical Sciences Research Council (EPSRC). The testbed and networks are depicted schematically (in much simplified form) in Figure A.1. The scientific objective of the TeraGyroid project is to study defect pathways and dynamics in Gyroid[1] self-assembly via the largest set of lattice-Boltzmann simulations ever performed. TeraGyroid addressed a large-scale problem of genuine scientific interest at the same time as it showed how intercontinental Grids can be interconnected to reduce the time to insight. The application was based upon RealityGrid [W172] and used computational steering techniques developed to accelerate the exploration of parameter spaces.

In terms of Grid technology, the project demonstrated the migration of simulations across the Atlantic. Calculations exploring the parameter space were steered from University College London (UCL) and Boston University. Steering required reliable near-real-time data transport across the Grid to visualization engines. The output datasets were visualized at a number of sites including Manchester

[1] Amphiphiles are chemicals with hydrophobic (water-avoiding) tails and hydrophilic (water-attracting) heads. When dispersed in solvents or oil/water mixtures, they self-assemble into complex shapes called Gyroids

University, University College London (UCL), and Argonne National Laboratory (ANL) using both commodity clusters and Silicon Graphics Incorporation (SGI) Onyx systems. Migrations required the transfer of large checkpoint files (0.5 TB) across the Atlantic. The computational resources used are listed in Table A.1 where processing capacity and memory space are expressed in Tera Flops (TF) and in Tera Bytes (TB), respectively.

Table A.1. Experimental computational resources

Site	Architecture	Processors	Peak (TF)	Memory (TB)
HPCx (Daresbury)	IBM Power 4 Regatta	1024	6.6	1.0
CSAR (Manchester)	SGI Origin 3800	512	0.8	0.5 (shared)
CSAR (Manchester)	SGI Altrix	256	1.3	0.38
PSC (Pittsburgh)	HP-Compaq	3000	6	3.0
NCSA (Urbana-Champaign)	Itanium 2	512	2.0	4.0
SDSC (San Diego)	Itanium 2	256	1.3	1.0

The network used was essentially a pre-UKLight configuration (Figure A.1). Within the UK, the traffic was carried on the SuperJANET development network (MB-NG), which provided a dedicated 2.5 Gbps IP path. British Telecommunications donated two 1 Gbps links from London to Amsterdam, which in conjunction with high-bandwidth SURFnet [W175] provision and the TeraGrid backbone completed the circuit. The project was a major success as it resulted in the first transatlantic federation of major high performance computing facilities through the use of Grid technology.

B

Network Coding

In today's communication networks, end-to-end information delivery is achieved by routing operations performed at intermediate nodes. However, routing does not encompass all operations that can be performed at a node. Recently, the notion of *network coding* emerges as a promising generalization of routing.

The network coding mechanism is proposed to improve the throughput utilization of a given network topology. Actually, the principle behind network coding is to allow an intermediate node to generate data by encoding (*i.e.,* computing a function of) its received data. Compared to other traditional approaches (*e.g.,* building multicast trees), network coding makes optimal use of the available network resources. Potential advantages of this generality of network coding over routing are many folds: resource efficiency, computational efficiency, and robustness to network dynamics [C51].

B.1 Origin of Network Coding

Network coding has its origins in the work of Ahlswede *et al.* [J145] and Li *et al.* [J146]. Ahlswede *et al.* [J145] show that coding within a network allows a source to multicast information at a rate approaching the smallest minimum cut between the source and any receiver as the coding symbol size approaches infinity. In this same work, the first example that highlights the utility of network coding was also given. Figure B.1 shows this famous example that reinvigorated the field of network information theory.

In this example, the source node a has two information bits x and y that need to be forwarded to the nodes f and g. Each directed edge represents a channel that can transmit one information bit at a time. Suppose that node a sends the bit x on the edge toward node b and the bit y on the edge toward node c. The node b will forward the bit x to both nodes d and f, while the node c will forward the bit y to both nodes d and g. In traditional transport network, node d can transmit either the bit x or the bit y. Finally, node e will forward the information bit received from node d to both nodes f and g. Whether the node d transmitted the information bit x or y, one of the two sink nodes f and g will receive one bit twice and will never receive the other bit.

Figure B.1 shows a way around this difficulty. The label on each edge specifies the function of x and y transmitted over that edge. Node a sends the bit x on one out-edge and the bit y on the other one. Nodes b and c copy the bit they receive to both out-edges. The key element of this approach is that the node d can transmit the function $x \oplus y$ rather than either the bit x or the bit y alone. Finally, node e copies the bit it receives to both out-edges. Consequently, node f receives both x and $x \oplus y$

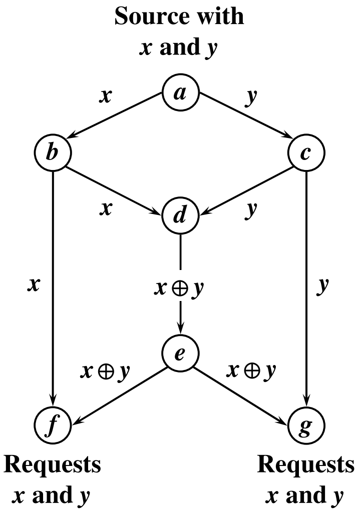


Fig. B.1. A network coding solution that achieves multicast from node a to nodes f and g.

and thus, it can compute the value of y as well. Similarly, node g can compute x from y and $x \oplus y$. Hence, both nodes f and g obtain both bits x and y.

B.2 Two Restricted Problems

There are two restricted versions of the network coding problem: the multicast problem and the k-pairs communication problem [D158]. Both problems are considered as fundamental problems in network communication.

B.2.1 Multicast Problem

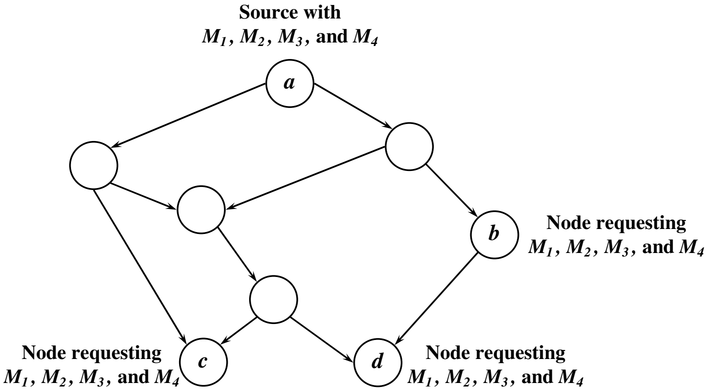


Fig. B.2. An example of the multicast problem.

In this version of the network coding problem, there is a single source for all the messages and a set of sinks which request these messages. Figure B.2 represents an example of the multicast problem wherein node a is the source of four messages, and nodes b, c, and d are the sinks which request all the messages. Since all messages have the same source and the same set of sinks, one can rephrase the feasibility of a multicast problem as an optimization problem in which the source is trying to send as many messages as possible to all the sinks.

In small generalization of the multicast problem, each sink still requests all the messages, but there is no restriction on the sources. This generalization is interesting because essentially all the results that hold for the multicast problem also hold for this generalization.

B.2.2 *k*-Pairs Communication Problem

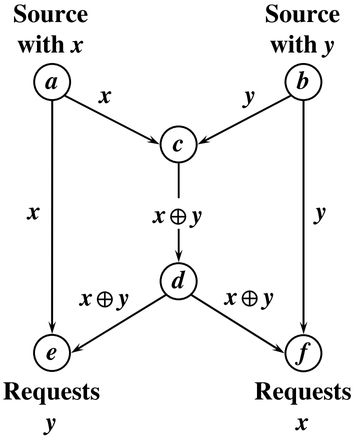


Fig. B.3. An example of the k-pairs communication problem.

Another important problem in network communication is the provisioning k different point-to-point connections. In an instance of the k-pairs communication problem, there are k commodities where each commodity has a single source and a single sink. In the example in Figure B.3, there are two commodities. For the one commodity, node a is the source and node f is the sink. For the other, node b is the source and node e is the sink. The k-pairs communication problem is interesting because it is closely related to the classical multicommodity flow problem.

On an abstract level, instances of both problems are defined in exactly the same way. However, commodities in the multicommodity flow problem are only traffic “flows” whereas commodities in the k-pairs communication problem are “information” (*i.e.,* they can be copied and encoded).

B.3 From Information Theory to Network Coding

Network coding is a relatively new field that has roots in information theory as well as connections to two classical problems in graph theory; namely the *Steiner tree packing* problem and the *multicommodity flow* problem.

B.3.1 Information Theory

Information theorists have concentrated on communication over a single channel. In the closest approach to network coding, a set of senders wishes to transmit information across a channel to a set of receivers [B154]. However, a network cannot be accurately modeled as a single channel because the physical structure of a network introduces complexities that are not present in the single channel case.

B.3.2 Steiner Tree Packing

In the traditional view of communication in a network, data can be replicated at nodes but not encoded together with other data. The problem of packing fractional Steiner trees in a graph can be used to model this type of communication.

Given a directed graph $\mathcal{G}$, a specified node r, and a subset $\mathcal{S}$ of vertices, a Steiner tree is a tree rooted at r and containing $\mathcal{S}$. A fractional Steiner tree is a Steiner tree along with a weight between 0 and 1. In the Steiner tree packing problem, the objective is to find the maximum number of edge-disjoint fractional Steiner trees in $\mathcal{G}$.

The problem of multicasting from a source to a set of sinks has traditionally been studied as a fractional Steiner tree packing problem. Assuming that a node r needs to transmit data to a set $\mathcal{S}$ of nodes, each tree in a fractional Steiner tree packing can be used to send a fraction of the data given by its weight.

B.3.3 Multicommodity Flow

The communication problem in which there is a single source and a single sink for each commodity has traditionally been viewed as a multicommodity flow problem. In this perspective, data is modeled as fluid: the amount of data leaving a node must be exactly equal to the amount of data entering this node except at the source and the sink.

In an instance of the multicommodity flow problem, there is a directed capacitated graph $\mathcal{G}$ and k commodities. For each commodity i, there is a single source s_i, a single sink t_i, and a demand d_i. In a multicommodity flow, the total flow of all the commodities across an edge in $\mathcal{G}$ must be no more than the capacity of that edge. The objective is to find the largest fraction r such that for every commodity i, at least rd_i units of commodity i flow from source s_i to sink t_i. The multicommodity flow problem is most closely related to the k-pairs communication problem.

The fundamental concept of network coding was introduced for satellite communication networks in [J147]. The concept was fully developed in a subsequent work by Ahlswede *et al.* in [J145] where the term "network coding" was introduced. The advantage of network coding over routing (*i.e.,* the traditional way of operating a network) was pointed out for the first time by means of a very simple example known as the butterfly network. Today, the application of network coding extends well beyond maximizing network capacity to areas such as error-correction, wire-tapping protection and detection, and distributed network management.

C

Radio Over Fiber Technology

Wireless communication is entering a new phase where the focus is shifting from voice to multimedia services. Present costumers are no longer interested in the underlying technology, but they simply need reliable and cost effective communication systems that can support, any time and anywhere, any media they want. Furthermore, new wireless subscribers are signing up at an increasing rate demanding more capacity while the radio spectrum is limited. Consequently, wide-band radio links become more prevalent in today's communication systems.

One emerging technology in wireless access systems is Radio Over Fiber (ROF) also called fiber to the air. Nowadays, radio over fiber systems are considered as a promising candidate to support real time wireless multimedia services by combining the high capacity of optical fibers and the flexibility of wireless networks. A radio over fiber system consists of a Central Station (CS) and a number of Remote Stations (RSs). In such a system, in order to reduce the cost of the remote stations, most of the signal processing (including coding, multiplexing, radio frequency generation and modulation, etc.) is made in the central station rather than in the base stations.

The basic concept of ROF consists in modulating a laser diode by means of pre-modulated radio frequency instead of traditional baseband information [W176]. Several techniques for generating and distributing microwave signals via optical fiber exist. These techniques may be classified into two main categories; namely Intensity Modulation-Direct Detection (IM-DD) and Remote Heterodyne Detection (RHD) [J148]. In the central station, the electrical signal may be a baseband signal, a modulated intermediate frequency signal, or a modulated radio frequency signal. In all cases, the aim is to produce appropriate radio frequency signals at the remote station that meet the specifications of the wireless application, *i.e.,* radio frequency signal must contain data in appropriate modulation format.

Today's ROF systems are designed to perform additional functionalities besides transportation and mobility functions. These functions include data modulation, signal processing, and frequency conversion [C52, J148].

C.1 Advantages of ROF Systems

In the following, we discuss the main advantages of ROF technology.

C.1.1 Low Attenuation Loss

The distribution of high frequency signals on electrical copper pairs has been widely investigated for Digital Subscriber Line (xDSL) access systems. Attenuation distortion, phase distortion, and crosstalk are some of the major disruptive effects observed on copper lines in access networks. Today, the most performing xDSL technology, Very high bit rate DSL (VDSL2+), enables downstream data rates up to 100 Mbps and upstream data rates up to a few tens of Mbps but on very short range (under 300 meters). Similarly, the attenuation, shadowing, and Rayleigh effects increase with frequency in free space transmission. However, the most commonly used Single Mode Fiber (SMF) has attenuation losses below 0.2 dB/km and 0.5 dB/km in the 1.5 μm and the 1.3 μm transmission windows, respectively. Polymer Optical Fiber (POF), a recent type of optical fibers, exhibits higher attenuations ranging from 10 to 40 dB/km in the 500 nm and the 1300 nm regions [C53]. These losses are much lower than those encountered by high frequency microwaves in free space propagation and copper wire transmission. Therefore, by transmitting microwaves in optical form, transmission distances are increased several folds and the required transmission powers are significantly reduced.

C.1.2 Large Bandwidth

Today's SMF optical fibers can provide a huge transmission bandwidth up to 50 THz. In addition to the high transmission capacity, such a bandwidth enables high speed signal processing such as filtering, mixing, frequency conversion, etc. Furthermore, processing in the optical domain allows to use cost-effective optical components such as laser diodes and modulators [J149]. For instance, millimeter-wave filtering can be achieved by converting the electrical signal to be filtered into an optical signal. Filtering is performed in the optical domain using optical components such as the Mach-Zehnder interferometer or Bragg gratings. Finally, the filtered signal is converted back into an electrical signal.

C.1.3 Easy Installation and Maintenance

In ROF systems, complex and expensive equipment is kept at the CSs, thereby making the RSs relatively simpler. For instance, most ROF techniques eliminate the need for a local oscillator and related equipment at the RS. In such cases, a photodetector, a Radio Frequency (RF) amplifier, and an antenna make up the RS equipment, whereas modulation equipment and switching equipment are kept at the CS and are shared by several RSs. Such a scheme reduces significantly the RSs installation and maintenance costs. Easy installation and low maintenance cost of RSs are key requirements in millimeter-wave systems because of the large number of required antenna sites.

C.1.4 Reduced Power Consumption

Reduced power consumption is favored by the reduced range of the wireless propagation section between a CS and a RS. In some applications, the antenna sites are operated in passive mode. Considering that RSs are sometimes placed in areas not covered by the power grid, reduced power consumption could be a significant feature of ROF systems.

C.2 Application of Radio Over Fiber Technology

ROF technology may be introduced in mobile communications, mobile broadband systems, wireless local areas networks, etc. The main application areas are briefly discussed below.

C.2.1 Mobile Communications

The filed of mobile communications is an important application area of ROF technology. The ever-increasing demands for bandwidth services as well as the rising number of mobile subscribers have heightened the need for increasing bandwidth capacity on the underlying telecommunication infrastructure. Therefore, mobile traffic (in Global System for Mobile communications (GSM) and Universal Mobile Telecommunications System (UMTS) networks) can be relayed cost-effectively between the CSs and the RSs by exploiting the benefits of fiber technology. Other ROF functionalities such as dynamic capacity allocation can offer significant operational benefits to cellular networks.

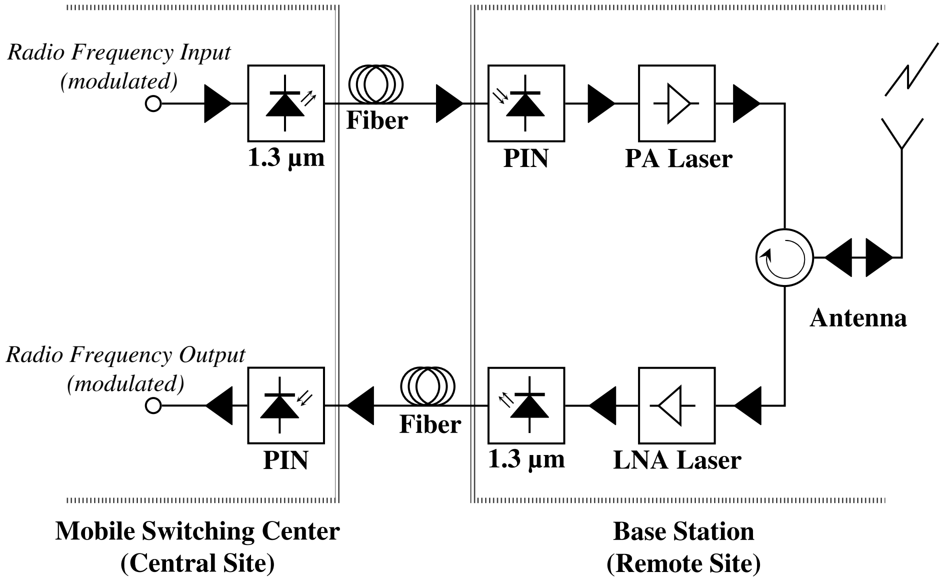


Fig. C.1. 900 MHz radio over fiber system

Figure C.1 illustartes an example of ROF systems in the framework of a GSM network. In the area of GSM networks, the CS could be the Mobile Switching Center (MSC) and the RS could be the Base Station (BS). The benefit of using such a system lays in the capability to dynamically allocate capacity based on traffic demands.

C.2.2 Mobile Broadband Systems

A Mobile Broadband System (MBS) is a mobile network that supports broadband services. Future MBSs such as mobile Wimax (IEEE 802.16e), Beyond-3^{rd} generation mobile networks (Beyond-3G), and 4^{th} generation mobile networks (4G) under specification within the 3^{rd} Generation Partnership Project (3GPP) consortium [W177] target data rates at the air interface up to 155 Mbps per user. Such data rates in the wireless domain are only achievable on very short range. Hence, the 60 GHz band is allocated as follows: the $62-63$ GHz band is allocated for the downlink ,and the $65-66$ band is allocated for the uplink. Since the cell size is of the order of hundreds of meters (micro-cells), a high density of radio cells is required in order to achieve the desired coverage. If ROF technology is used to generate the millimeter-waves, the base stations would be simpler and cheaper, thereby making full scale deployment of MBS networks economically feasible.

C.2.3 Wireless Local Area Networks

As portable terminals become increasingly powerful and widespread, the demand for mobile broadband access to Local Area Networks (LANs) is also increasing. This leads to higher carrier frequencies in the bid to meet the demand for capacity. For instance, current wireless LANs operate at 2.4 GHz and offer a maximum capacity up to 11 Mbps per carrier (IEEE 802.11b). Next generation broadband wireless LANs are primed to offer up to 54 Mbps per carrier and require higher carrier frequencies in the 5 GHz band (IEEE 802.11a/D7.0). Higher carrier frequencies in turn lead to micro- and pico-cells which involve difficulties associated with the coverage problem. A cost effective way around this problem is to deploy ROF technology. A wireless LAN at 60 GHz has been realized by first transmitting from the CS an Intermediate Frequency (IF) generated by a stable oscillator together with the data over the fiber. The oscillator frequency is then used to up-convert the data to millimeter-waves at the RSs which greatly simplifies the RSs and leads to efficient base station design.

References

Conference Papers

[C1] M. Brunato, and R. Battiti. A multistart randomized greedy algorithm for traffic grooming on mesh logical topologies. In *Procs. ONDM'02*, pages 417–430, February 2002.

[C2] S. Thiagarajan, and A. K. Somani. A capacity correlation model for WDM network with constrained grooming capabilities. In *Procs. ICC'01*, pages 1592–1596, June 2001.

[C3] B. Ramamurthy, and A. Ramakrishnan. Virtual topology reconfiguration of wavelength-routed optical WDM networks. In *Procs. GLOBECOM'00*, pages 1269–1275, November 2000.

[C4] A. Gençata, and B. Mukherjee. Virtual-topology adaptation for WDM mesh networks under dynamic traffic. In *Procs. INFOCOM'02*, pages 48–56, June 2002.

[C5] I. Alfouzan, and A. Jayasumana. Dynamic reconfiguration of wavelength-routed WDM networks. In *Procs. LCN'01*, pages 477–485, November 2001.

[C6] M. Brunato, R. Battiti, and E. Salvadori. Load balancing in WDM networks through adaptive routing table changes. In *Procs. NETWORKING'02*, pages 289–300, May 2002.

[C7] L. Zhang, K.-H. Lee, C.-H. Youn, and H.-G.Yeo. Adaptive virtual topology reconfiguration policy employing multi-stage traffic prediction in optical internet. In *Procs. HPSR'02*, pages 127–131, May 2002.

[C8] W. Yao, and B. Ramamurthy. Rerouting schemes for dynamic traffic grooming in optical WDM mesh networks. In *Procs. GLOBECOM'04*, pages 1793–1797, November 2004.

[C9] K.-M. Chan, and T. P. Yum. Analysis of least congested path routing in WDM lightwave networks. In *Procs. INFOCOM'94*, pages 962–969, June 1994.

[C10] A. Gencata, and B. Mukherjee. CATZ (Capacity Allocation with Time Zones): A methodology for cost-efficient bandwidth allocation and reconfiguration in a world-wide WDM network. In *Procs. ECOC'02*, pages 1–2, September 2002.

[C11] H. Zhu, H. Zang, K. Zhu, and B. Mukherjee. Dynamic traffic grooming in WDM mesh networks using a novel graph model. In *Procs. GLOBECOM'02*, pages 2681–2685, November 2002.

[C12] S. Ramamurthy, and B. Mukherjee. Survivable WDM mesh networks. Part I-Protection. In *Procs. INFOCOM'99*, pages 744–751, March 1999.

[C13] S. Ramamurthy, and B. Mukherjee. Survivable WDM mesh networks. Part II-Restoration. In *Procs. ICC'99*, pages 2023–2030, June 1999.

[C14] I. Tomkos. Transport performance of WDM metropolitan area transparent optical networks. In *Procs. OFC'02*, pages 350–352, March 2002.

[C15] R. Cardillo, V. Curri, and M. Mellia. Considering transmission impairments in wavelength routed networks. In *Procs. ONDM'05*, pages 421–429, February 2005.

[C16] N. Barakat, and A. Leon-Garcia. An analytical model for predicting the locations and frequencies of 3R regenerations in all-optical wavelength-routed WDM networks. In *Procs. ICC'02*, pages 2812–2816, April 2002.

[C17] M.A. Ezzahdi, S. Al Zahr, M. Koubàa, N. Puech, and M. Gagnaire. LERP: A quality of transmission dependent heuristic for routing and wavelength assignment in hybrid WDM networks. In *Procs. ICCCN'06*, pages 125–130, October 2006.

[C18] S. Al Zahr, M. Gagnaire, N. Puech, and M. Koubàa. Physical layer impairments in WDM core networks: A comparison between a north-American backbone and a pan-European backbone. In *Procs. BROADNETS'05*, pages 335–340, October 2005.

[C19] I. Ouveysi, and Y. K. Tham. Network design for multi-hour traffic profile. In *Procs. ATNAC'95*, December 1995.

[C20] I. Ouveysi, F. Safael, and A. Wirth. Dimensioning and dynamic reconfiguration of hierarchical multi-service crossconnect platforms for multihour traffic profiles. In *Procs. ICC'98*, pages 249–252, June 1998.

[C21] J. Kuri, N. Puech, M. Gagnaire, and E. Dotaro. Routing and wavelength assignment of scheduled lightpath demands in a WDM optical transport network. In *Procs. ICOCN'02*, pages 270–273, November 2002.

[C22] J. Kuri, N. Puech, M. Gagnaire, and E. Dotaro. Routing foreseeable lightpath demands using a tabu search meta-heuristic. In *Procs. GLOBECOM'02*, pages 2803–2807, November 2002.

[C23] M. Roughan, and J. Gottlieb. Large scale measurement and modeling of backbone internet traffic. In *Procs. ITCom'02*, pages 190–201, July/August 2002.

[C24] M. Roughan, A. Greenberg, C. Kalmanek, M. Rumsewicz, J. Yates, and Y. Zhang. Experience in measuring backbone traffic variability: Models, metrics, measurements and meaning. In *Procs. ITC'03*, pages 379–388, September 2003.

[C25] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg. Fast accurate computation of large-scale IP traffic matrices from link loads. In *Procs. SIGMETRICS'03*, pages 206–217, June 2003.

[C26] K. Papagiannaki, N. Taft, and C. Diot. Impact of flow dynamics on traffic engineering design principles. In *Procs. INFOCOM'04*, pages 2295–2306, March 2004.

[C27] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot. Traffic matrix estimation: Existing techniques and new directions. In *Procs. SIGCOMM'02*, pages 161–174, August 2002.

[C28] Y. Zhang, M. Roughan, C. Lund, and D. Donoho. An information-theoretic approach to traffic matrix estimation. In *Procs. SIGCOMM'03*, pages 301–312, August 2003.

[C29] J. Brutlag. Aberrant behavior detection in time series for network monitoring. In *Procs. LISA'00*, pages 139–146, December 2000.

[C30] V. Yegneswaran, P. Barford, and J. Ullrich. Internet intrusions: Global characteristics and prevalence. In *Procs. SIGMETRICS'03*, pages 138–147, June 2003.

[C31] K. Papagiannaki, N. Taft, Z. Zhang, and C. Diot. Long-term forecasting of internet backbone traffic: Observations and initial models. In *Procs. INFOCOM'03*, pages 1178–1188, April 2003.

[C32] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True. Deriving traffic demands for operational IP networks: Methodology and experience. In *Procs. SIGCOMM'00*, pages 257–270, August/September 2000.

[C33] S. Sarvotham, R. Riedi, and R. Baraniuk. Connection-level analysis and modeling of network traffic. In *Procs. SIGCOMM'01*, pages 99–103, August 2001.

[C34] P. Tomlinson, G. Hill, and A. Tzanakaki. Comparison of transparent and opaque optical transport network designs. In *Procs. ECOC'02*, pages 1–2, September 2002.

[C35] A. Tzanakaki, I. Zacharopoulos, and I. Tomkos. Near and longer term architectural designs for OXCs/OADMs/Network topologies. In *Procs. PS'03*, pages 271–273, October 2003.

[C36] H. Lausen, T. Klein, L. Bersiner, M. M. Klein Koerkamp, M. C. Donckers, B. H. M. Hams, and W. H. G. Horsthuis. Pigtailed thermo-optic 1×2-switch in polymer: Design and experimental evaluation. In *Procs. EFOC&N'94*, pages 99–101, June 1994.

[C37] I. Chlamtac, A. Fumagalli, and C.-J. Suh. Switching multi-buffer delay lines for contention resolution in all-optical deflection networks. In *Procs. GLOBECOM'96*, pages 1624–1628, November 1996.

[C38] S. Yamano, F. Xue, and S. J. B. Yoo. Load-sensitive deflection routing for contention resolution in optical packet switched networks. In *Procs. ICCCN'03*, pages 243–248, October 2003.

[C39] J. P. Jue. An algorithm for loopless deflection in photonic packet-switched networks. In *Procs. ICC'02*, pages 2776–2780, April/May 2002.

[C40] G. C. Hudek, and D. J. Muder. Signaling analysis for a multi-switch all-optical network. In *Procs. ICC'95*, pages 1206–1210, June 1995.

[C41] M. Yoo, and C. Qiao. Just-Enough-Time (JET): A high speed protocol for bursty traffic in optical networks. In *Procs. IEEE/LEOS Annual Meeting'97*, pages 26–27, August 1997.

[C42] K. E. Stukjaer, C. Joergensen, S. L. Danielsen, B. Mikkelsen, M. Vaa, R. J. Pedersen, H. Povlsen, M. Schilling, K. Daub, K. Dutting, W. Idler, M. Klenk, E. Lach, G. Laube, K. Wunstel, P. Doussiere, A. Jourdan, F. Pommerau, G. Soulage, L. Goldstein, J. Y. Emery, N. Vodjdani, F. Ratovelomanana, A. Enard, G. Glastre, D. Rondi, and R. Blondeau. Wavelength conversion devices and techniques. In *Procs. ECOC'96*, pages 33–40, September 1996.

[C43] S. J. B. Yoo, C. Caneau, R. Bhat, and M. A. Koza. Wavelength conversion by quasi-phase-matched difference frequency generation in AlGaAs waveguides. In *Procs. OFC'95*, pages 377–380, February/March 1995.

[C44] J. Yates, J. Lacey, D. Everitt, and M. Summerfield. Limited-range wavelength translation in all-optical networks. In *Procs. INFOCOM'96*, pages 954–961, March 1996.

[C45] G. Jeong, and E. Ayanoglu. Comparison of wavelength-interchanging and wavelength-selective cross-connects in multiwavelength all-optical networks. In *Procs. INFOCOM'96*, pages 156–163, March 1996.

[C46] K. Zhu, and B. Mukherjee. On-line approaches for provisioning connections of different bandwidth granularities in WDM mesh networks. In *Procs. OFC'02*, pages 549–551, March 2002.

[C47] J. Kuri, N. Puech, and M. Gagnaire. Diverse routing of scheduled lightpath demands in an optical transport network. In *Procs. DRCN'03*, pages 69–76, October 2003.

[C48] D. Eppstein. Finding the k shortest paths. In *Procs. FOCS'94*, pages 154–165, November 1994.

[C49] I. Chlamtac, A. Ganz, and G. Karmi. Lightnet: Lightpath based solutions for wide bandwidth WANs. In *Procs. INFOCOM'90*, pages 1014–1021, June 1990.

[C50] H. Harai, M. Murata, and H. Miyahara. Performance of alternate routing methods in all-optical switching networks. In *Procs. INFOCOM'97*, pages 516–524, April 1997.

[C51] C. Gkantsidis, and P. R. Rodriguez. Network coding for large scale content distribution. In *Procs. INFOCOM'05*, pages 2235–2245, March 2005.

[C52] J. M. Fuster, J. Marti, P. Candelas, F. J. Martinez, and L. Sempere. Optical generation of electrical modulation formats. In *Procs. ECOC'01*, pages 536–537, September 2001.

[C53] Y. Koike. POF technology for the 21st century. In *Procs. POF'01*, pages 5–8, September 2001.

Journals & Magazines

[J54] E. W. M. Wong, A. K. M. Chan, and T.-S. P. Yum. A taxonomy of rerouting in circuit switched networks. *IEEE Communications Magazine*, 37(11):116–122, November 1999.

[J55] P. Aukia, M. Kodialam, P. V. N. Koppol, T. V. Lakshman, H. Sarin, and B. Suter. RATES: A server for MPLS traffic engineering. *IEEE Network*, 14(2):34–41, March/April 2000.

[J56] L. Chiu, and H. Modiano. Traffic grooming algorithms for reducing electronic multiplexing costs in WDM ring networks. *IEEE/OSA Journal of Lightwave Technology*, 18(1):2–12, January 2000.

[J57] J. Wang, W. Cho, V. R. Vemuri, and B. Mukherjee. Improved approaches for cost-effective traffic grooming in WDM ring networks: ILP formulations and single-hop and multihop connections. *IEEE/OSA Journal of Lightwave Technology*, 19(11):1645–1653, November 2001.

[J58] P. J. Wan, G. Calinescu, L. Liu, and O. Frieder. Grooming of arbitrary traffic in SONET/WDM BLSRs. *IEEE Journal on Selected Area in Communications*, 18(10):1995–2003, October 2000.

[J59] X. Zhang, and C. Qiao. An effective and comprehensive approach for traffic grooming and wavelength assignment in SONET/WDM rings. *IEEE/ACM Transaction on Networking*, 8(5):608–617, October 2000.

[J60] K. Zhu, and B. Mukherjee. Traffic grooming in an optical WDM mesh network. *IEEE Journal on Selected Area in Communications*, 20(1):122–133, January 2002.

[J61] S. Thiagarajan, and A. K. Somani. Capacity fairness of WDM networks with grooming capabilities. *SPIE Optical Network Magazine*, 2(3):24–31, May/June 2001.

[J62] H. Zhu, H. Zang, K. Zhu, and B. Mukherjee. A novel generic graph model for traffic grooming in heterogeneous WDM mesh networks. *IEEE/ACM Transaction on Networking*, 11(2):285–299, April 2003.

[J63] D. Banerjee, and B. Mukherjee. Wavelength-routed optical networks: Linear formulation, resource budgeting tradeoffs, and a reconfiguration study. *IEEE/ACM Transaction on Networking*, 8(5):598–607, October 2000.

[J64] A. Gencata, and B. Mukherjee. Virtual-topology adaptation for WDM mesh networks under dynamic traffic. *IEEE/ACM Transactions on Networking*, 11(2):236–247, April 2003.

[J65] W. Golab, and R. Boutaba. Policy driven automated reconfiguration for performance management in WDM optical networks. *IEEE Communication Magazine*, 42(1):44–51, January 2004.

[J66] A. Narula-Tam, and E. Modiano. Dynamic load balancing in WDM packet networks with and without wavelength constraints. *IEEE Journal on Selected Areas in Communications*, 18(10):1972–1979, October 2000.

[J67] N. Sreenath, B. H. Gurucharan, G. Mohan, and C. Siva Ram Murthy. A two-stage approach for virtual topology reconfiguration of WDM optical networks. *SPIE Optical Networks Magazine*, 2(3):58–71, May/June 2001.

[J68] J.-F. P. Labourdette, G. W. Hart, and A. S. Acampora. Branch-Exchange sequences for reconfiguration of lightwave networks. *IEEE Transactions on Communications*, 42(10):2822–2832, October 1994.

[J69] K.-C. Lee, and V. O. K. Li. A wavelength rerouting algorithm in wide-area all-optical networks. *IEEE/OSA Journal of Lightwave Technology*, 14(6):1218–1229, June 1996.

[J70] H. Zang, J. P. Jue, and B. Mukherjee. A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks. *SPIE Optical Network Magazine*, 1(1):47–60, January 2000.

[J71] E. Karasan, and E. Ayanoglu. Performance of WDM transport networks. *IEEE Journal on Selected Areas in Communications*, 16(7):1081–1096, September 1998.

[J72] M. Pickavet, P. Demeester, D. Colle, D. Staessens, B. Puype, L. Depre, and I. Lievens. Recovery in multilayer optical networks. *IEEE/OSA Journal of Lightwave Technology*, 24(1):122–134, January 2006.

[J73] P. Demeester, M. Gryseels, A. Autenrieth, C. Brianza, L. Castagna, G. Signorelli, R. Clemenfe, M. Ravera, A. Jajszczyk, D. Janukowicz, K. Van Doorselaere, and Y. Harada. Resilience in multilayer networks. *IEEE Communications Magazine*, 37(8):70–76, August 1999.

[J74] S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee. Survivable WDM mesh networks. *IEEE/OSA Journal of Lightwave Technology*, 21(4):870–883, April 2003.

[J75] S. De Maesschalck, D. Colle, A. Groebbens, C. Develder, A. Lievens, P. Lagasse, M. Pickavet, P. Demeester, F. Saluta, and M. Quagliatti. Intelligent optical networking for multilayer survivability. *IEEE Communications Magazine*, 40(1):42–49, January 2002.

[J76] G. Mohan, C. Siva Ram Murthy, and A. K. Somani. Efficient algorithms for routing dependable connections in WDM optical networks. *IEEE/ACM Transactions on Networking*, 9(5):553–566, October 2001.

[J77] J. L. Marzo, E. Calle, C. Scoglio, and T. Anjali. QoS online routing and MPLS multilevel protection: A survey. *IEEE Communication Magazine*, 41(10):126–132, October 2003.

[J78] R. Sabella, E. Iannone, M. Listani, M. Berdusco, and S. Binetti. Impact of transmission performance on path routing in all-optical transport networks. *IEEE/OSA Journal on Lightwave Technology*, 16(11):1965–1972, November 1998.

[J79] J. Strand, A. L. Chiu, and R. Tkach. Issues for routing in the optical layer. *IEEE Communications Magazine*, 39(2):81–87, February 2001.

[J80] Y. Huang, J. P. Heritage, and B. Mukherjee. Connection provisioning with transmission impairment consideration in optical WDM networks with high-speed channels. *IEEE/OSA Journal of Lightwave Technology*, 23(3):982–993, March 2005.

[J81] I. Cerutti, A. Fumagalli, and M. J. Potasek. Effect of chromatic dispersion and self-phase modulation in multihop multirate WDM rings. *IEEE Photonics Technology Letters*, 14(3):411–413, March 2002.

[J82] G. Shen, W. Grover, T. Cheng, and S. Bose. Sparse placement of electronic switching nodes for low-blocking in translucent optical networks. *OSA Journal of Optical Networking*, 1(12):424–441, November 2002.

[J83] X. Yang, and B. Ramamurthy. Sparse regeneration in translucent wavelength-routed optical networks: Architecture, network design and wavelength routing. *Photonic Network Communications*, 10(1):39–53, July 2005.

[J84] E. Rosenberg. A nonlinear programming heuristic for computing optimal link capacities in a multi-hour alternate routing communications network. *Journal of Operations Research*, 35(3):354–364, May-June 1987.

[J85] J. Kruithof. Telefoonverkeersrekening (calculation of telephone traffic). *De Ingenieur*, 52(8):E15–E25, February 1937.

[J86] V. Paxson, and S. Floyd. Wide-area traffic: The failure of poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, June 1995.

[J87] Y. Vardi. Network tomography: Estimating source-destination traffic intensities from link data. *Journal of the Americal Statistical Association*, 91(433):365–377, March 1996.

[J88] C. Tebaldi, and M. West. Bayesian inference on network traffic using link count data. *Journal of the American Statistical Association*, 93(442):557–576, June 1998.

[J89] M. E. Crovella, and A. Bestavros. Self-similarity in world wide web traffic: Evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6):835–846, December 1997.

[J90] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2(1):1–15, February 1994.

[J91] T. Tuan, and K. Park. Multiple time scale congestion control for self-similar network traffic. *Performance Evaluation*, 36(1):359–386, August 1999.

[J92] K. Park, and T. Tua. Performance evaluation of multiple time scale TCP under self-similar traffic conditions. *ACM Transactions on Modeling and Computer Simulation*, 10(2):152–177, April 2000.

[J93] J. Cao, D. Davis, S. Vander Wiel, and B. Yu. Time-varying network tomography: Router link data. *Journal of the American Statistical Association*, 95(452):1063–1075, December 2000.

[J94] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True. Deriving traffic demands for operational IP networks: Methodology and experience. *IEEE/ACM Transactions on Networking*, 9(3):265–280, June 2001.

[J95] J. Kuri, N. Puech, M. Gagnaire, E. Dotaro, and R. Douville. Routing and wavelength assignments of scheduled lightpath demands. *IEEE Journal on Selected Areas in Communications*, 21(8):1231–1240, October 2003.

[J96] A. L. Chiu, and E. H. Modiano. Traffic grooming algorithms for reducing electronic multiplexing costs in WDM ring networks. *IEEE/OSA Journal of Lightwave Technology*, 18(1):2–12, January 2000.

[J97] C. R. Giles, and M. Spector. The wavelength add/drop multiplexer for lightwave communication networks. *Bell Labs Technical Journal*, 4(1):207–229, January-March 1999.

[J98] B. Mukherjee. WDM optical communication networks: Progress and challenges. *IEEE Journal on Selected Areas in Communications*, 18(10):1810–1824, October 2000.

[J99] L. A. Cox, and J. Sanchez. Cost savings from optimized packing and grooming of optical circuits: Mesh versus ring comparisons. *SPIE Optical Networks Magazine*, 2(3):72–90, May/June 2001.

[J100] G. Shen, T. H. Cheng, S. K. Bose, C. Lu, and T. Y. Chai. Architectural design for multistage 2-D MEMS optical switches. *IEEE/OSA Journal of Lightwave Technology*, 20(2):178–187, February 2002.

[J101] J. T. W. Yeow, and S. S. Abdallah. Novel MEMS L-switching matrix optical cross-connect architecture: Design and analysis-optimal and staircase-switching algorithms. *IEEE/OSA Journal of Lightwave Technology*, 23(10):2877–2892, October 2005.

[J102] A. Ware. New photonic-switching technology for all-optical networks. *Lightwave Magazine*, 17(3):92–98, Mar 2000.

[J103] M. Makihara, M. Sato, F. Shimokawa, and Y. Nishida. Micromechanical optical switches based on thermocapillary integrated in waveguide substrate. *IEEE/OSA Journal of Lightwave Technology*, 17(1):14–18, January 1999.

[J104] S. Hardy. Liquid-crystal technology vies for switching applications. *Lightwave Magazine*, 16(13):44–46, December 1999.

[J105] N. K. Shankar, J. A. Morris, C. P. Yakymyshyn, and C. R. Pollock. A 2×2 fiber optic switch using chiral liquid crystals. *IEEE Photonics Technology Letters*, 2(2):147–149, February 1990.

[J106] D. K. Cheng, Y. Liu, and G. J. Sonek. Optical switch based on thermally activated dye-doped biomolecular thin films. *IEEE Photonics Technology Letters*, 7(4):366–369, April 1995.

[J107] A. Kar-Roy, and C. S. Tsai. 8×8 symmetric nonblocking integrated acousto-optic space switch module on linbo$_3$. *IEEE Photonics Technology Letters*, 4(7):731–734, July 1992.

[J108] D. A. Smith, A. d'Alessandro, J. E. Baran, D. J. Fritz, J. L. Jackel, and R. S. Chakravarthy. Multiwavelength performance of an apodized acousto-optic switch. *IEEE/OSA Journal of Lightwave Technology*, 14(9):2044–2051, September 1996.

[J109] A. Agrawal, T. S. El-Bawab, and L. B. Sofman. Comparative account of bandwidth efficiency in optical burst switching and optical circuit switching networks. *Photonic Network Communications*, 9(3):297–309, May 2005.

[J110] T. S. El-Bawab, A. Agrawal, F. Poppe, L. B. Sofman, D. Papdimitriou, and B. Rousseau. The evolution to optical-switching-based core networks. *SPIE Optical Networks Magazine*, 4(2):7–19, March/April 2003.

[J111] T. S. El-Bawab, and J.-D. Shin. Optical packet switching in core networks: Between vision and reality. *IEEE Communications Magazine*, 40(9):60–65, September 2002.

[J112] L. Dittmann, C. Develder, D. Chiaroni, F. Neri, F. Callegati, W. Koerber, A. Stavdas, M. Renaud, A. Rafel, J. Sole-Pareta, W. Cerroni, N. Leligou, L. Dembeck, B. Mortensen, M. Pickavet, N. Le Sauze, M. Mahony, B. Berde, and G. Eilenberger. The european ist project david: A viable approach toward optical packet switching. *IEEE Journal on Selected Areas in Communications*, 21(7):1026–1040, September 2003.

[J113] S. Yao, B. Mukherjee, S. J. B. Yoo, and S. Dixit. A unified study of contention-resolution schemes in optical packet-switched network. *IEEE/OSA Journal of Lightwave Technology*, 21(3):672–683, March 2003.

[J114] I. Chlamtac, A. Fumagalli, L. G. Kazovsky, P. Melman, W. H. Nelson, P. Poggiolini, M. Cerisola, A. N. M. M. Choudhury, T. K. Fong, R. T. Hofmeister, C.-L. Lu, A. Mekkittikul,

D. J. M. Sabido IX, C.-J. Suh, and E. W. M. Wong. CORD: Contention resolution by delay lines. *IEEE Journal on Selected Areas in Communications*, 14(5):1014–1029, June 1996.

[J115] Y. Li, G. Xiao, and H. Ghafouri-Shiraz. Fixed-wavelength conversion for contention resolution in optical packet switches. *Microwave and Optical Technology Letters*, 41(3):185–187, March 2004.

[J116] L. Yi, and H. Ghafouri-Shiraz. Contention resolution by shared wavelength converters and fiber delay lines in an optical packet switch. *Microwave and Optical Technology Letters*, 38(5):395–398, July 2003.

[J117] C. Qiao, and M. Yoo. Optical Burst Switching (OBS): A new paradigm for an optical Internet. *Journal of High Speed Networks*, 8(1):69–84, March 1999.

[J118] T. Battestilli, and H. Perros. An introduction to optical burst switching. *IEEE Communications Magazine*, 41(8):S10–S15, August 2003.

[J119] R. S. Barr, and R. A. Patterson. Grooming telecommunication networks. *SPIE Optical Networks Magazine*, 2(3):20–23, May/June 2001.

[J120] D. Campi, and C. Coriasso. Wavelength conversion technologies. *Photonic Network Communications*, 2(1):85–95, March 2000.

[J121] S. J. B. Yoo. Wavelength conversion technologies for WDM network applications. *IEEE/OSA Journal of Lightwave Technology*, 14(6):955–966, June 1996.

[J122] R. W. Tkach, A. R. Chraplyvy, F. Forghieri, A. H. Gnauck, and R. M. Derosier. Four-photon mixing and high-speed WDM systems. *IEEE/OSA Journal of Lightwave Technology*, 13(5):841–849, May 1995.

[J123] M. C. Tatham, G. Sherlock, and L. D. Westbrook. $20 - nm$ optical wavelength conversion using nondegenerate four-wave mixing. *IEEE Photonics Technology Letters*, 5(11):1303–1306, November 1993.

[J124] M. C. Cardakli, A. B. Sahin, O. H. Adamczyk, A. E. Willner, K. R. Parameswaran, and M. M. Fejer. Wavelength conversion of subcarrier channels using difference frequency generation in a PPLN waveguide. *IEEE Photonics Technology Letters*, 14(9):1327–1329, September 2002.

[J125] H. Kawaguchi, K. Oe, H. Yasaka, K. Magari, M. Fukuda, and Y. Itaya. Tunable optical wavelength conversion using a multielectrode distributed feedback diode with a saturable absorber. *IEEE Electronics Letters*, 23(20):1088–1090, September 1987.

[J126] T. Durhuus, B. Mikkelsen, C. Joergensen, S. Lykke Danielsen, and K. E. Stubkjaer. All-optical wavelength conversion by semiconductor optical amplifiers. *IEEE/OSA Journal of Lightwave Technology*, 14(6):942–954, June 1996.

[J127] T. Dnrhuus, C. Joergensen, B. Mikkelsen, R. J. S. Pedersen, and K. E. Stubkjaer. All optical wavelength conversion by SOA's in a Mach-Zehnder configuration. *IEEE Photonics Technology Letters*, 6(1):53–55, January 1994.

[J128] H. J. Lee, M. Sohn, K. Kim, and H. G. Kim. Wavelength dependent performance of a wavelength converter based on cross-gain modulation and birefringence of a semiconductor optical amplifier. *IEEE Photonics Technology Letters*, 11(2):185–187, February 1999.

[J129] R. A. Barry, and P. A. Humblet. Models of blocking probability in all-optical networks with and without wavelength changers. *IEEE Journal on Selected Areas in Communications*, 14(5):858–867, June 1996.

[J130] S. Subramaniam, M. Azizoglu, and A. K. Somani. All-optical networks with sparse wavelength conversion. *IEEE/ACM Transaction on Networking*, 4(4):544–557, August 1996.

[J131] K.-C. Lee, and V. O. K. Li. A wavelength-convertible optical network. *IEEE/OSA Journal of Lightwave Technology*, 11(5):962–970, May/June 1993.

[J132] A. Banerjee, J. Drake, J. P. Lang, B. Turner, K. Kompella, and Y. Rekhter. Generalized multiprotocol label switching: An overview of routing and management enhancements. *IEEE Communications Magazine*, 39(1):144–150, January 2001.

[J133] A. Banerjee, J. Drake, J.P. Lang, B. Turner, B. Awduche, L. Berger, K. Kompella, and Y. Rekhter. Generalized multiprotocol label switching: an overview of signaling enhancements and recovery techniques. *IEEE Communications Magazine*, 39(7):144–151, July 2001.

[J134] R. Ramaswami, and K.N. Sivarajan. Routing and wavelength assignment in all-optical networks. *IEEE/ACM Transaction on Networking*, 3(5):489–500, October 1995.

[J135] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science Magazine*, 220(4598):671–680, May 1983.

[J136] H. Zang, J. P. Jue, and B. Mukherjee. A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks. *SPIE Optical Networks Magazine*, 1(1):47–60, January 2000.

[J137] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, December 1959.

[J138] R. Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 16(1):87–90, January 1958.

[J139] A. Birman. Computing approximate blocking probabilities for a class of all-optical networks. *IEEE Journal on Selected Areas in Communications*, 14(5):852–857, June 1996.

[J140] J. P. Lang, V. Sharma, and E. A. Varvarigos. An analysis of oblivious and adaptive routing in optical networks with wavelength translation. *IEEE/ACM Transactions on Networking*, 9(4):503–517, August 2001.

[J141] R. Ramaswami, and A. Segall. Distributed network control for optical networks. *IEEE/ACM Transactions on Networking*, 5(6):936–943, December 1997.

[J142] K. Zhu, H. Zhu, and B. Mukherjee. Traffic engineering in multigranularity heterogeneous optical WDM mesh networks through dynamic traffic grooming. *IEEE Network*, 17(2):8–15, March/April 2003.

[J143] G. Mohan, and C. Siva Ram Murthyy. A time optimal wavelength rerouting algorithm for dynamic traffic in WDM networks. *IEEE/OSA Journal of Lightwave Technology*, 17(3):406–417, March 1999.

[J144] Student [William Sealy Gosset]. The probable error of a mean. *Biometrika*, 6(1):1–25, March 1908.

[J145] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, July 2000.

[J146] S.-Y. R. Li, R. W. Yeung, and N. Cai. Linear network coding. *IEEE Transactions on Information Theory*, 49(2):371–381, February 2003.

[J147] R. W. Yeung, and Z. Zhang. Distributed source coding for satellite communications. *IEEE Transactions on Information Theory*, 45(4):1111–1120, May 1999.

[J148] U. Gliese, T. N. Nielsen, S. Norskov, and K. E. Stubkjaer. Multifunctional fiber-optic microwave links based on remote heterodyne detection. *IEEE Transactions on Microwave Theory and Techniques*, 46(5):458–468, May 1998.

[J149] G. Maury, A. Hilt, T. Berceli, B. Cabon, and A. Vilcot. Microwave-frequency conversion methods by optical interferometer and photodiode. *IEEE Transactions on Microwave Theory and Techniques*, 45(8):1481–1485, August 1997.

Books

[B150] K. Zhu, H. Zhu, and B. Mukherjee. *Traffic Grooming in Optical WDM Mesh Networks.* Springer Science + Business Media, Inc., New York, NY 10013, USA, 2005.

[B151] J.-P. Vasseur, M. Pickavet, and P. Demeester. *Network Recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS.* Morgan Kaufmann Publishers, San Francisco, CA 94111, USA, 2004.

[B152] B. Mukherjee. *Optical WDM Networks.* Springer Science + Business Media, Inc., New York, NY 10013, USA, 2006.

[B153] R. Ramaswami, and K. N. Sivarajan. *Optical Networks: A Practical Perspective.* Morgan Kaufmann Publishers, San Francisco, CA 94104, USA, 2002.

[B154] T. M. Cover, and J. A. Thomas. *Elements of Information Theory.* John Wiley & Sons, Inc., Hoboken, NJ 07030, USA, 2006.

Dissertations

[D155] A. Gençata. *Topology and Bandwidth Adaptation in Optical WDM Backbone Networks with Dynamic Traffic.* PhD thesis, Istanbul Technical University, 2003.

[D156] J. Kuri. *Optimization Problems in WDM Optical Transport Networks with Scheduled Lightpath Demands.* PhD thesis, Ecole Nationale Supérieure des Télécommunications, 2003.

[D157] S. Ramamurthy. *Optical Design of WDM Network Architectures.* PhD thesis, University of California, Davis, 1998.

[D158] A. R. Lehman. *Network Coding.* PhD thesis, Massachusetts Institute of Technology, 2005.

Websites

[W159] The Internet Engineering Task Force Website.
http://www3.ietf.org/home.html.

[W160] Optical Internetworking Forum Website.
http://www.oiforum.com/.

[W161] International Telecommunication Union Website.
http://www.itu.int/net/home/index.aspx.

[W162] National Science Foundation Website: A brief history of NSF and the Internet.
http://www.nsf.gov/news/news_summ.jsp?cntn_id=103050.

[W163] National Laboratory for Applied Network Research, Measurement and Network Analysis Group.
http://www.ntu.edu.sg/home5/PG01080112/.

[W164] Pew Internet & American Life Project Website.
http://www.pewinternet.org.

[W165] Abilene Website: Advanced networking for leading-edge research and education.
http://abilene.internet2.edu/.

[W166] Infoplease Website: U.S. statistics, cities, states, military affairs, postal information, societies, and associations.
http://www.infoplease.com/states.html.

[W167] IETF Working Group Website: Path Computation Element (PCE).
http://www.ietf.org/html.charters/pce-charter.html.

[W168] ILOG CPLEX Website.
http://www.ilog.com/products/cplex/.

[W169] Sandia National Laboratories Website: A survey of global optimization methods.
http://www.cs.sandia.gov/opt/survey/main.html.

[W170] U.S. National Institute of Standards and Technology Website: Dictionary of algorithms and data structures.
http://www.nist.gov/dads/.

[W171] Open Grid Forum Website.
http://www.ogf.org/index.php.

[W172] TeraGyroid Project Website: Grid-based lattice-Boltzmann simulations of defect dynamics in amphiphilic liquid crystals.
http://www.realitygrid.org/TeraGyroid/.

[W173] TeraGrid Project Website.
http://www.teragrid.org/.

[W174] Research Councils UK Website: About the UK e-Science programme.
http://www.rcuk.ac.uk/escience/.

[W175] SURFnet Website: High-quality Internet for higher education and research.
http://www.surfnet.nl/info/en/home.jsp.

[W176] Advanced Radio-Optics Integrated Technology (ADROIT) Group Website.
http://www.rnet.ryerson.ca/adroit/.

[W177] The 3rd Generation Partnership Project (3GPP) Website.
http://www.3gpp.org/.

List of Publications

[1] M. Gagnaire, and E. A. Doumith are co-authors of a chapter to appear in *Traffic Grooming for Optical Networks: Foundations and Techniques* co-ordinated by Rudra Duta, Ahmed Kamal, and George Rouskas and published by Springer Science.

[2] E. A. Doumith, and M. Gagnaire. Impact of traffic predictability on WDM EXC/OXC network's performance. To appear in *Procs. of the IEEE Journal on Selected Areas in Communications (JSAC), Issue on traffic engineering for multi-layer networks*, pages 895–904, June 2007.

[3] E. A. Doumith, and M. Gagnaire. Traffic routing in a multi-layer optical network considering rerouting and grooming strategies. In *Procs. of the* 49th *IEEE Global Communication Conference (GLOBECOM'06)*, San Francisco California USA, December 2006.

[4] E. A. Doumith, and M. Gagnaire. Traffic grooming in multi-layer WDM networks: Meta-heuristics versus sequential algorithms. In *Procs. of the* 12th *IEEE EUNICE Open European Summer School (EUNICE'06)*, pages 129–136, Stuttgart Germany, September 2006.

[5] E. A. Doumith, M. Gagnaire, O. Audouin, and R. Douville. From network planning to traffic engineering for optical VPN and multi-granular random demands. In *Procs. of the* 25th *IEEE International Performance Computing and Communication Conference (IPCCC'06)*, pages 127–134, Phoenix Arizona USA, April 2006.

[6] E. A. Doumith, M. Gagnaire, O. Audouin, and R. Douville. Network nodes dimensioning assuming electrical traffic grooming in an hybrid OXC/EXC WDM network. In *Procs. of the* 2nd *IEEE/Create-Net International Conference on Broadband Networks (BROADNETS'05)*, pages 286–294, Boston Massachusetts USA, October 2005.

[7] E. A. Doumith, M. Gagnaire, O. Audouin, and N. Puech. Traffic engineering for virtual private networks and random traffic demands in WDM optical core networks. In *Procs. of the* 14th *IEEE International Conference on Computer Communications and Networks (ICCCN'05)*, pages 317–322, San Diego California USA, October 2005.

[8] E. A. Doumith, M. Koubaa, N. Puech, and M. Gagnaire. Gain and cost brought in by wavelength conversion for the routing and wavelength assignment of two traffic classes in WDM networks. In *Procs. of the* 10th *European Conference on Networks & Optical Communications (NOC'05)*, pages 147–154, London UK, July 2005.

Glossary

SYMBOLS 1-9

2D	2-Dimensional.
3D	3-Dimensional.
3G	3rd Generation.
3GPP	3rd Generation Partnership Project.
3R	Re-amplification, Re-shaping, and Re-timing.
4G	4th Generation.

A

ADM	Add Drop Multiplexer.
ANL	Argonne National Laboratory.
ATLRR	Adaptive Time Limited Resource Reservation.
ATM	Asynchronous Transfer Mode.

B

BER	Bit Error Rate.
BS	Base Station.

C

CAC	Constraint based Routing Label Distribution Protocol.
CI	Confidence Interval.
CPL	Common Path Length.
CPU	Central Processing Unit.
CR-LDP	Central Station.
CS	Central Station.
CULG	Consecutive Uncommon Links Group.

D

DFG	Difference Frequency Generation.
DLE	Dynamic Lightpath Establishment.
DoS	Denial-of-Service.

E

ECS	Electrical Circuit Switch(-ing).
E/O	Electro-Optical.
EPS	Electrical Packet Switch(-ing).
EPSRC	Engineering and Physical Sciences Research Council.
EXC	Electrical Cross-Connect.

F

FDM	Frequency Division Multiplexing.
FF	First-Fit.
FWM	Four Wave Mixing.

G

GL	Grooming Lightpath.
GMPLS	Generalized Multi-Protocol Label Switching.
GSM	Global System for Mobile communications.

H

HDTV	High-Definition Television.

I

IETF	Internet Engineering Task Force.
IF	Intermediate Frequency.
IG	Iterative Greedy.
ILP	Integer Linear Program.
IM-DD	Intensity Modulation - Direct Detection.
IP	Internet Protocol.
IS-IS	Intermediate System to Intermediate System.
ISP	Internet Service Provider.
ITU-T	International Telecommunication Union.

L

LAN	Local Area Network.
LP	Lightpath.
LSP	Label Switched Path.
LUW	Least Used Wavelength.
LVHF	Least Virtual Hop First.

M

MBS	Mobile Broadband System.
MEMS	Micro-Electro-Mechanical System.
MHTM	Multi-Hour Traffic Matrices.
MPLS	Multi-Protocol Label Switching.
MSC	Mobile Switching Center.
MTV	Move-To-Vacant.
MTV-WR	Move-To-Vacant Wavelength Retuning.
MUW	Most Used Wavelength.

N

NSF	National Science Foundation.
NSFNET	National Science Foundation NETwork.

O

OADM	Optical Add/Drop Multiplexer.
OBS	Optical Burst Switch(-ing).
OCS	Optical Circuit Switch(-ing).
O/E	Opto-Electrical.
O/E/O	Optical-Electrical-Optical.
OGF	Open Grid Forum.
OIF	Optical Internetworking Forum.
OPS	Optical Packet Switch(-ing).
OSPF	Open Shortest Path First.
OVPN	Optical Virtual Private Network.
OWS	Optical Wavelength Switch(-ing).
OXC	Optical Cross-connect.

P

PAR	Path Adjusting Rerouting.
PDL	Polarization-Dependent Loss.
POF	Polymer Optical Fiber.

Q

QoS	Quality of Service.
QoT	Quality of Transmission.

R

RAM	Random Access Memory.
RED	Random Electrical Demand.
RF	Radio Frequency.
RHD	Remote Heterodyne Detection.
RLD	Random Lightpath Demand.
RLEL	Rerouting at the Logical Electrical Level.
ROF	Radio Over Fiber.
RPOL	Rerouting at the Physical Optical Level.
RS	Remote Station.
RSVP	Resource Reservation Protocol.
RWA	Routing and Wavelength Assignment.
RxD	Random Demand.

S

SA	Simulated Annealing.
SAN	Storage Area Network.
SDH	Synchronous Digital Hierarchy.
SED	Scheduled Electrical Demand.
SGI	Silicon Graphics Incorporation.
SGP	Successful Grooming Pair.
SLD	Scheduled Lightpath Demand.
SLE	Static Lightpath Establishment.
SMF	Single Mode Fiber.
SOA	Semiconductor Optical Amplifiers.
SONET	Synchronous Optical Network.
sRED	semi-Random Electrical Demand.
sRxD	semi-Random Demand.
sRxLD	semi-Random Lightpath Demand.
SxD	Scheduled Demand.

T

TB	Tera Bytes.
TCP	Transfer Control Protocol.
TDM	Time Division Multiplexing.
TE	Traffic Engineering.
TF	Tera Flops.
TLRR	Time Limited Resource Reservation.

U

UCL	University College London.
UGP	Unsuccessful Grooming Pair.
UK	United Kingdom.
UMTS	Universal Mobile Telecommunications System.
USA	United States of America.

V

VC	Virtual Circuit.
VDSL	Very high bitrate Digital Subscriber Line.
VP	Virtual Path.

W

WDM	Wavelength Division Multiplexing.
WR	Wavelength Retuning.

X

xDSL	Digital Subscriber Line.
xED	Electrical Demand.
XFM	Cross-Phase Modulation.
XGM	Cross-Gain Modulation.
xLD	Lightpath Demand.

Vita

Elias A. DOUMITH was born on July 1980, in Bhersaf, Lebanon. He completed his undergraduate studies in electronic engineering at the Lebanese University of Computer Science and Telecommunications, Roumieh, Lebanon. In 2003, he received a Master of Science from the École Nationale Supérieure des Télécommunications (ENST) in Paris, France. In 2004, he joined the Networks and Computer Science Department of the ENST to pursue his PhD's studies in Optical Networks. His main field of interest is network planning and traffic engineering in multi-granular WDM optical networks. His thesis work can be summarized by:
- Design a traffic model inspired from traffic measurement on a real core network. The proposed traffic classes cover the time predictability scale from the most deterministic to the most random allowing for multi-granularity traffic demands.
- Detail a double stage node architecture that handles full-wavelength circuits as well as sub-wavelength electrical connections.
- Investigate the problem of efficiently designing a WDM mesh network with grooming capability for a given set of traffic requests via theoretical formulation (ILP formulation) as well as simulation experiments (Simulated Annealing and Iterative Greedy algorithms).
- Develop two traffic engineering optimization algorithms for online dynamic routing and grooming purposes. These algorithms extend the basic shortest-path algorithm reflecting various grooming policies.
- Conceive two rerouting algorithm to enhance network performance under dynamic random traffic scenarios.