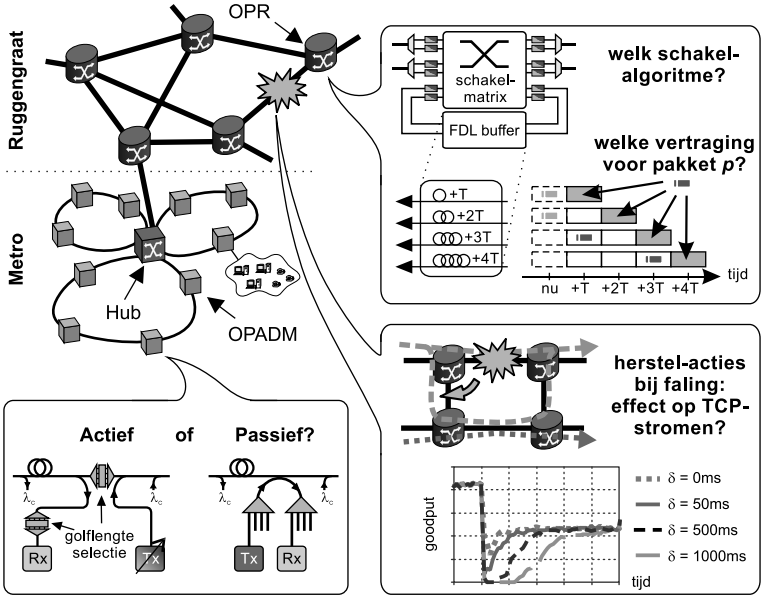
UNIVERSITEIT GENT

Universiteit Gent
Faculteit Toegepaste Wetenschappen
Vakgroep Informatietechnologie

Ontwerp en analyse van optische pakketschakelende netwerken

Design and analysis of optical packet switching networks

Chris Develder



INTEC

Proefschrift tot het bekomen van de graad van
Doctor in de Toegepaste Wetenschappen:
Elektrotechiek
Academiejaar 2003-2004

Promotor: Prof. Dr. Ir. Mario Pickavet

Universiteit Gent
Faculteit Toegepaste Wetenschappen

Vakgroep Informatietechnologie (INTEC)
Sint-Pietersnieuwstraat 41
BE-9000 Gent, België

tel. +32 9 264 99 70
fax. +32 9 264 99 60
web: http://www.intec.ugent.be

De auteur genoot voor dit werk een mandaat als Aspirant van het Fonds voor Wetenschappelijk Onderzoek – Vlaanderen (FWO–Vl.).

Voorwoord

Soms moeilijk uit te leggen, wat het nu precies betekent straks—hopelijk—een doctoraat te behalen. Iets waar ik amper een beeld van had toen ik er vier jaar geleden aan begon. Uitzonderlijk vroeg trouwens, die eerste vrijdag van oktober 1999, iets voor half negen 's ochtends (sic!)—een record dat ik de afgelopen vier jaar tot mijn scha en schande amper scherper heb kunnen stellen. 's Ochtends vroeg dus, zo goed als moederziel alleen op zoek naar m'n PC, een stukje netwerkkabel, software. En daarna, beginnen met het échte werk. Opzoekwerk vooral, veel lezen, me gaandeweg steeds meer afvragend wat ik nu precies zou gaan doen. Daar geleidelijk aan een vaag antwoord op vinden, en een ietwat concreter wanneer ik na een dik half jaar quasi stuurloos ronddobberen op een—gelukkig erg rustige—onderzoekszee in het Europese vaarwater terechtkom en er kennis maak met David, een project door intimi ook gekend als "Data And Voice Integration over DWDM". Daarin een leidraad en inspiratie vinden voor onderzoeksonderwerpen. Nu, dik drie jaar later, tegelijk met het vaarwel zeggen aan David, vaststellen dat ik gaandeweg voldoende materiaal bijeengesprokkeld heb om er dit proefschrift uit te distilleren. Waardoor mensen me straks wellicht "doctor" mogen noemen. Maar dan niet van 't soort dat helpen kan wanneer iemand onwel is.

Soms moeilijk uit te leggen dus. Maar vooraleer daartoe een (hopelijk lovenswaardige) poging te ondernemen, mag in deze prelude de traditionele pleiade van bedankjes niet ontbreken. Er zijn immers een heleboel mensen die het tot stand komen van het bundeltje papier dat u nu in handen houdt mogelijk gemaakt hebben. Allereerst de mensen van het FWO-Vlaanderen voor het verlenen van een doctoraatsbeurs. Voor de aanzet ertoe ben ik dank verschuldigd aan prof. Hans Vangheluwe, toen nog doctorandus in de Biomath onderzoeksgroep, die me het idee influisterde om die beurs aan te vragen. Helaas heeft hij daar zelf weinig aan gehad, want zoals u op de kaft al kon lezen, kwam dit proefschrift tot stand in de vakgroep Informatietechnologie, kortweg Intec. En dat heb ik mede te danken aan m'n toenmalige thesisbegeleider Mike Vogeleer. Verrassend zal het niet zijn dat ik nu plaats maak om nogal wat andere Intec-mensen te danken.

Op de eerste plaats wil ik onze gedreven voorzitter prof. Paul Lagasse bedanken om me van de faciliteiten van de vakgroep gebruik te laten maken. Uiteraard dank ook ik prof. Piet Demeester om me de kans te geven onder zijn hoede het doctoraatswerk te verrichten. Hem, maar ook en vooral prof. Mario Pickavet ben ik dankbaar voor de begeleiding en inhoudelijke discussies. Bij die

laatste apprecieerde ik ook de bijdragen van m'n (ex-)collega's Didier Colle, Jan Cheyns, Erik van Breusegem, Ruth Van Caenegem, Sophie Demaesschalck, Peter Arijs, Filip De Turck. Voor de onontbeerlijke technische ondersteuning gaat mijn welgemeende danku uit naar Bert De Vuyst, Bart De Vreese, Brecht Vermeulen en Pieter Thysebaert. Deze laatste vier maken trouwens ook deel uit van het lijstje collega's waaraan ik vooral dank verschuldigd ben voor de minstens evenzeer gesmaakte bijstand in recreatieve momenten. Een lijstje dat in analfabetische volgorde vervolledigd wordt met Koert Vlaeminck, Bruno Volckaert, Thijs Lambrecht, Bart Dhoedt, Filip De Turck, Mario Pickavet, Bart Duysburgh, Sofie Van Hoecke, et al. Het rijtje Intec-mensen wil ik graag afsluiten met zij die op het administratieve vlak die tot dit doctoraat hebben bijgedragen, waarbij ik onwillekeurig denk en dank aan Martine Buysse, Ilse Meersman en Karien Hemelsoen.

Onder meer dankzij hun financieel-technische hoogstandjes kon ik steeds present zijn op de vergaderingen van het eerder vermelde Europese project David. En zo ook buiten de eigen vakgroep al es een stimulans tot onderzoek meemaken. Het spreekt voor zich dat ik hiervoor dank verschuldigd ben aan de David-partners, waarvan ik speciaal Franco Callegati, Walter Cerroni, Nicolas Le Sauze, Tina Fjelde, Dominique Chiaroni, Brian Mortensen, Michael Berger en uiteraard project-leider Lars Dittmann wil vermelden. Ook de partners van andere Europese projecten, met name Lion en Cost 266, mogen in dit bedankje niet ontbreken. Op nationale schaal ben ik een van de gelukkigen die mag meewerken aan het GBOU-project rond optische knoop- en netwerkarchitecturen. Een GBOU-project waaruit ik de collega's van andere vakgroepen en universiteiten hier graag vernoemen wil, zoals bijvoorbeeld Benny Van Houdt en Koen Laevens.

Tot bijna slot wil ik ook de mensen bedanken die me toelieten in hun midden m'n zinnen te verzetten en eventjes letterlijk en/of figuurlijk ver weg te zijn van optische pakketschakelende knopen: m'n reisgezelletjes (danku Karen, Gudrun, Sabine, Eva, Luc, Daan, Hilke, Sarah, Roel, Joris, et al.), m'n oud-medestudentes reclamekunst (danku Caroline, Ilke, Heidi, An), m'n tegen- en medespelertjes dinsdag- of woensdagavondlijk minivoetbal, en uiteraard alle (oud-)leden van het Gents Academie-Orkest (in het bijzonder An, Eleonora, Valerie, Wouter, Evelyne, Nele, Katrien, Katrien, Wim). Van deze laatsten verdienen Nele en Evelyne een ongelooflijk speciale vermelding, vermits zij de moed hebben opgebracht het naleescomité voor dit proefschrift te vormen en zich doorheen de volgende honderd en nog wat bladzijden te worstelen—iets wat u dankzij hun tussenkomst hopelijk iets makkelijker zal vallen. Wat mij er trouwens doet aan denken dat ik bijna vergeten was Mario Pickavet te bedanken voor het leveren van ook inhoudelijke feedback. Bij deze: merci Mario!

En dan nu het echte slot van dit voor—of bij nader inzien eigenlijk hoofdzakelijk dank—woord, dat ik afrond met een dankuwel voor de zorgen en steun van m'n familie, niet alleen tijdens m'n vier jaar doctoraat, maar ook de voorafgaande studentenjaren.

Gent, 3 oktober 2003.
Chris Develder.

Inhoud

Acroniemen

2R	Reshaping and Reamplification
3R	Reshaping, Reamplification and Retiming
ADSL	Asynchronous Digital Subscriber Line
ATM	Asynchrone Transfer Mode
AWG	Arrayed Waveguide Grating
B&S	Broadcast-and-Select
BCP	Burst Controle-Pakket
BHP	Burst Hoofding-Pakket
EDFA	Erbium-Doped Fibre Amplifier (met Erbium gedopeerde vezelsignaal-versterker)
FBG	Fibre Bragg Grating
FDL	Fiber Delay Line (vezelvertragingslijn)
FIFO	First In First Out
FSK	Frequency Shift Keying
FTTH	Fiber To The Home (optische vezel tot thuis)
GMPLS	Generalized Multi-Protocol Label Switching (veralgemeend MPLS)
ILP	Integer Linear Programming
IP	Internet Protocol
LAUC	Latest Available Unused Channel (laatst beschikbare ongebruikte kanaal)
LAUC-VF	LAUC with Void Filling (LAUC met leemtevulling)
MAC	Medium Access Control (toegangscontrole tot een medium)
MAN	Metropolitan Area Network (metro-netwerk)
MEMS	Micro-Electro-Mechanical System(s)

MMAP	Markov Arrival Process with Marked transitions
MPLS	Multi-Protocol Label Schakelen
OBS	Optisch Burst Schakelen
OLT	Optical Line Terminal
ONU	Optical Network Unit
OPS	Optisch PakketSchakelen
OXC	Optical Cross Connect (optische kruisschakelaar)
PLR	Packet Loss Rate (pakketverlies-verhouding)
PON	Passief Optisch Netwerk
PSR	Photonic Slot Routing
QoS	Quality of Service
RAM	Random Access Memory (geheugen met willekeurige toegang)
RTO	Retransmission TimeOut (time-out voor heruitzenden)
RTT	Round Trip Time
RWA	Routing and Wavelength Assignment
SAR	Segmentation And Reassembly (segmentatie en wedersamenstelling)
SCM	Sub-Carrier Multiplexing
SFQ	Stochastic Fair Queuing
SOA	Semi-conductor Optical Amplifier (halfgeleider-optische versterker)
TCP	Transport Controle Protocol
WAN	Wide Area Network (ruggengraatnetwerk)
WDM	Wavelength Division Multiplexing (golflengtemultiplexering)

Hoofdstuk 1
Inleiding

Ondanks de recente economische terugval na het uiteenspatten van de technologie-zeepbel, stelt men ook vandaag nog steeds een aanhoudende groei vast van het telecommunicatieverkeer [40]. Het is al lang geen geheim meer dat het oorspronkelijk dominerende telefoonverkeer in volume overtroffen wordt door Internet-verkeer [41]. Geïnspireerd op de technologie van dat Internet, wat gekarakteriseerd wordt door een pakketgebaseerde overdracht van heterogene informatie, is het idee gegroeid om ook in de ondersteunende transportnetwerken pakketschakelen toe te passen, en dit om de aanwezige bandbreedte zo efficiënt mogelijk te benutten.

1.1 Optisch PakketSchakelen

De media die benut worden voor de transmissie van informatie zijn erg gediversifiëerd: men kan koper gebruiken (zoals in de klassieke telefoonlijnen, nu ook uitgebuit door xDSL technologieën, of coaxiale kabel) of zelfs lucht voor draadloze verbindingen, maar in de kern van het netwerk is optische vezel ongetwijfeld het sterkst vertegenwoordigd. Transmissie van licht in de vrije ruimte was al in 1880 door Bell gebruikt voor z'n zogenaamde fotofoon, maar het gebruik van tastbare optische dragers werd pas in de jaren vijftig van de volgende eeuw gepostuleerd. Praktisch gebruik van optische vezels bleef evenwel nog geruime tijd beperkt tot het vervaardigen van gebruiksvoorwerpen en kledij. Zowat de enige communicatie-gerelateerde toepassing was de vezelscoop (Eng. fibrescope), onder meer populair in medische kringen voor bv. gastroscopie. Medio jaren zestig, na de komst van de laser, begon men het ware potentieel van optische vezel als transmissiemedium te ontsluiten, maar was geremd door de te hoge attenuatie in toenmalige vezels: Kao geloofde in optische vezelcommunnicatie en schatte in 1966 dat men erin moest slagen de attenuatie onder de 20dB/km te krijgen [42]. Het was wachten tot 1970, toen onderzoekers bij Corning erin slaagden uit met titanium gedopeerd kwartsglas een vezel te fabriceren met voldoende lage attenuatie, zodat die geschikt was voor communicatie. Die vinding gaf een nieuwe impuls aan het onderzoekswerk, wat resulteerde in het ontdekken van enerzijds geschikte golflengtes waarbij de attenuatie minimaal was (eerst nog rond 800nm, maar nadien de superieure 1310nm en later 1550nm) en anderzijds vezels met verbeterde karakteristieken (in het

bijzonder de zogenaamde “single mode fibre”, SMF [43]). In de vroege jaren tachtig waren de eerste lange afstandsnetwerken gebaseerd op optische vezelcommunicatie dan ook een feit (cf. de netwerken van MCI en Sprint in de V.S.).

Een ontwikkeling die het gebruik van optische transmissietechnologie een extra ruggensteun gaf, was golflengtemultiplexering (Eng. Wavelength Division Multiplexing, WDM). Door de komst van versterkers met een relatief wijd golflengtebereik (de zogenaamde EDFA’s), die simultaan verschillende golflengten kunnen versterken, was het in de jaren negentig mogelijk geworden kost-efficiënt meerdere golflengten gelijktijdig over een zelfde vezel te versturen. Intussen is men erin geslaagd het aantal beschikbare golflengten nog op te drijven door nauwere kanaalspatiëring toe te passen en zijn bv. al bronnen met een bereik van meer dan 150 golflengten gedemonstreerd [44], terwijl multiplexers gekend zijn die zelfs tot 400 golflengten kunnen onderscheiden [45].

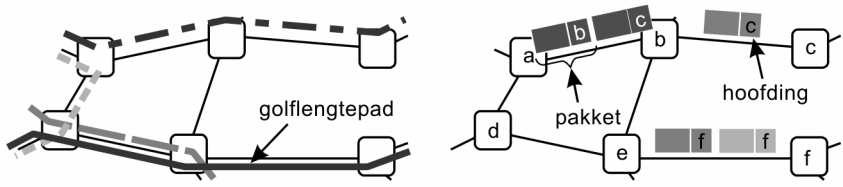
De eerste netwerken en zelfs het gros van hun hedendaagse opvolgers, gebruik(t)en optische communicatie voor punt-tot-punt verbindingen. Aan de uiteinden van de vezels wordt het licht telkens terug naar een elektrisch signaal omgezet en elektronisch verwerkt. Het correct routeren van de informatie doorheen het netwerk gebeurt dan puur elektronisch: van netwerk-functionaliteit in het optische domein is nog geen sprake. Recentere ontwikkelingen maken het evenwel mogelijk ook in de schakelpunten waar verschillende vezels samenkomen, de signalen optisch te behandelen. Een beroemd voorbeeld zijn de Micro-Elektro-Mechanische Systemen (MEMS, [46]), waarmee minuscule spiegeltjes gestuurd kunnen worden zodat ze licht van een ingangs- naar een uitgangsvezel overbrengen.

De functionaliteit van optische schakelaars die vandaag verkrijgbaar zijn laat toe om optische netwerken uit te bouwen die het concept van circuitschakelen toepassen. Dit betekent dat, om informatie uit te wisselen tussen twee eindpunten van het optische netwerk, er tussen deze punten een connectie opgezet wordt gedurende de duur van die uitwisseling, net zoals in klassieke telefonie. Er wordt dus bandbreedte gereserveerd, met name deze overeenstemmend met een golflengte, voor gegevensoverdracht tussen die eindpunten: men realiseert een zogenaamd lichtpad. Niet onbelangrijke verschilpunten met het voorbeeld van klassieke telefonie zijn evenwel dat (i) de bandbreedte veel hoger (2.5, 10 of 40 Gbit/s) en (ii) de duur van de verbinding veel langer is (uren, dagen, weken of zelfs maanden). Vermits een connectie overeenstemt met een golflengte, hanteert men ook wel de term golflengteschakelen (Eng. lambda switching). Verschillende nationale en internationale projecten hebben de werkbaarheid ervan aangetoond [47,48]. Het onderzoek met betrekking tot golflengteschakelen omvat tegenwoordig hoofdzakelijk het controleren en beheren van het resulterend optische netwerk [2,48]. Door de specifieke eigenschappen van de optische technologie dienen sommige van de bestaande werkwijzen uit het elektrische domein aangepast en/of uitgebreid te worden. In deze context wordt in onderzoeksmiddens vooral aandacht besteed aan het veralgemeend multi-protocol labelschakelen (Eng. Generalized Multiprotocol Label Switching, GMPLS [49–51]).

In de circuitgeschakelde aanpak van golflengteschakelen is het een heikel punt de aanwezige bandbreedte efficiënt te benutten, gelet op de hoge bandbreedte geassociëerd met een golflengte (en dus een circuit). Om aan de bandbreedte-

behoeften van de gebruikers van het optische netwerk zo kost-efficiënt mogelijk te voldoen, zal men aan de schakelaars in het optische netwerk ook elektrische schakelaars toevoegen die met een fijnere bandbreedte-granulariteit werken. Zo ontstaat het probleem van "grooming" (zie verder in Hoofdstuk 4): hoe en waar worden de (elektrische) verkeersstromen van fijnere granulariteit samengenomen in optische lichtpaden?

Om de bandbreedte aanwezig in het optische domein doeltreffend te benutten, is optisch pakketschakelen de aangewezen uitweg die het "grooming"-probleem omzeilt. In plaats van lichtpaden op te zetten, waarbij tussen de eindpunten een continue lichtstroom voor de gegevensoverdracht zorgt, zal men de gegevens over optische pakketten verdelen. Deze worden dan elk afzonderlijk tussen de eindpunten verstuurd. Zo kan één enkele golflengte gebruikt worden voor pakketten die behoren tot verschillende connecties, zolang die samen niet méér bandbreedte vereisen dan de golflengte voorziet. Om de verschillende verkeersstromen te onderscheiden, maakt men dan gebruik van de zogenaamde hoofding (Eng. header): met elke hoeveelheid gegevens die in een pakket vervat zit, correspondeert een hoofding die informatie bevat over die over te dragen gegevens, zoals begin- en eindpunt waartussen ze in het optische netwerk worden uitgewisseld. Het onderscheid met elektrisch pakketschakelen ligt in het feit dat de eigenlijke pakket-inhoud (Eng. payload) optisch behandeld wordt en dus in de schakelknopen geen conversie van en naar het elektrische domein ondergaat. Men spreekt in dit verband ook van transparante optische netwerken wanneer de eigenlijke bitrate van die inhoud willekeurig kan zijn. Merk op dat de hoofding wel elektro-optische conversie kan ondergaan, vermits de optische signaalverwerking, nodig om de hoofding te interpreteren, nog in haar kinderschoenen staat (zie verder).



Figuur 1.1: Circuit- versus pakketschakelen.

Wanneer men de term Optisch PakketSchakelen (OPS, Eng. Optical Packet Switching) hanteert [52–53], doelt men doorgaans op het geval waarbij de lengte van de pakketten constant is en aangeduid wordt met de term "slot". De schakelaars die deze pakketten behandelen zijn in dit geval voorzien van synchronisatie-eenheden aan hun ingangen zodat pakketten, afkomstig van verschillende vezels, en/of verschillende golflengtes op een zelfde vezel, gealigneerd worden. De schakelaar kan dan repetitief werken: slot per slot de hoofdingen van de nieuw aankomende pakketten inspecteren en al deze pakketten gelijktijdig naar de correcte uitgangen doorschakelen. Omdat dergelijke synchronisatie niet voor de hand ligt en om de vereiste schakelsnelheden te beperken, heeft men het Optisch Burst Schakelen (Eng. Optical Burst Switching, OBS) bedacht [54]. Hierbij is de pakketlengte variabel en typisch langer dan de in OPS gehanteerde slotlengte. De

schakelaar heeft nu geen synchronisatie-eenheden die pakketten over verschillende ingangen heen aligneert. De voornaamste motivatie voor OBS is het afzwakken van de technologische vereisten die gesteld worden door OPS. Daarbij is men bereid sommige gekende voordelen van slot-gebaseerde aanpakken (efficiëntere benutting van bandbreedte, eenvoudiger realiseren van gedifferentiëerde dienstkwaliteit, etc.) in te leveren.

In het kader van OPS en OBS is het niet oninteressant hier ook melding te maken van het zogenaamde ORION concept, wat staat voor "Overspill Routing In Optical Networks" en aan deze vakgroep bedacht werd (o.a. [55]). Het achterliggende idee is het gerbuik van een hybride aanpak die het beste van de pakket- en circuitgeschakelde concepten tracht te verenigen. Het concept stelt voor een golflengteschakelende kruisschakelaar (optisch) te combineren met een pakketschakelaar (bv. een elektronische IP-router). De binnenkomende golflengten worden via 1×2 schakelaars verbonden met zowel de pakketschakelaar als de golflengteschakelaar. De bedoeling is het gros van de gegevens volgens het principe van golflengteschakelen te versturen en dus operaties per pakket te vermijden. Om evenwel de aanwezige bandbreedte zo volledig mogelijk te benutten, kan door het meesturen van een aangepast controlesignaal de binnenkomende golflengte omgeleid worden naar de pakketschakelaar. Het gevolg is dat de pakketschakelaar een veel lagere belasting moet verwerken dan in een puur pakketgeschakeld concept, terwijl de bandbreedte nagenoeg even efficiënt benut wordt: het aantal golflengten die nodig zijn om een gegeven hoeveelheid verkeer te dragen is beduidend lager dan in een puur golflengteschakelend concept.

De commerciële realisatie van OPS wordt nog in de weg gestaan door technologische moeilijkheden, die voornamelijk bestaan uit het ontbreken of slechts in een beperkte vorm aanwezig zijn van (i) optisch geheugen, (ii) geavanceerde, volledig optische signaalverwerking voor de hoofding van de pakketten, (iii) optische synchronisatie en (iv) optische signaalregeneratie. Momenteel wijden verschillende groepen en projecten onderzoek aan het oplossen of verhelpen van deze tekortkomingen. Volledig optische 3R-regeneratie—herstellen van vorm, versterking en timing van het signaal—is intussen meermaals gedemonstreerd [56–58]. Ook synchronisatie via optische technologie werd onder meer in [59–61] gerealiseerd, maar blijft door het gebruik van vezelvertragingslijnen (Eng. Fibre Delay Lines, FDL) eerder omslachtig. Verschillende volledig optische implementaties van het verwerken van de hoofdingen in optische pakketschakelaars werden verwezenlijkt [62–65], ook al blijft de functionaliteit eerder beperkt. Echt optisch geheugen, in de zin van een tegenhanger van het elektrisch geheugen met in de tijd willekeurige toegang (Eng. Random Access Memory, RAM), is vandaag nog niet beschikbaar: het gebruik van optische flip-flops [66] om volledige pakketten te stockeren lijkt momenteel nog geen werkbare oplossing. Hoewel het stoppen van licht—of het tot zeer lage snelheden (tientallen m/s) vertragen ervan—geen science fiction meer is [67,69], lijkt het in praktijk nog niet bruikbaar, ondanks het feit dat ook bij kamertemperatuur al vertraagd licht waargenomen werd [70]. Het vooralsnog ontbreken van een optische tegenhanger van RAM kan gelukkig wel verholpen worden door het gebruik van FDL's, zoals onder meer verder in dit werk aangetoond zal worden. Een alternatief is het gebruik van elektrisch geheugen aan de rand van

het optische netwerk, waarbij de toegang tot het optisch gedeelte geregeld wordt via een zorgvuldig ontworpen protocol voor toegang tot het medium (Eng. Medium Access Control, MAC) zoals bv. uiteengezet in [71]. Zo kan het optisch netwerk zelf geheugenloos blijven.

Het mag duidelijk zijn dat al veelbelovende stappen genomen zijn in het overkomen van de huidige technologische beperkingen en we geloven dan ook dat OPS eens het onderzoeksstadium zal ontgroeien. Het resterende deel van dit hoofdstuk zullen we besteden aan een beknopte schets van OPS-architecturen in Sectie 1.2, gevolgd door de belangrijkste ontwerpskeuzes vanuit het standpunt van logische performantie (i.e. efficiëntie van bandbreedtegebruik, verlies van pakketten, kwaliteitsdifferentiatie van diensten) in Sectie 1.3. We ronden het hoofdstuk af in Sectie 1.4 met een overzicht van de in dit doctoraat gerealiseerde bijdragen, gevolgd door de obligatoire lijst van publicaties in Sectie 1.5.

1.2 Architecturen

De organisatie van een communicatienetwerk vertoont een zekere hiërarchie: men onderscheidt lokale netwerken (Eng. Local Area Networks, LAN) zoals binnen een bedrijfssite of campus, toegangsnetwerken (Eng. access network), netwerken op metropole schaal (Eng. Metropolitan Area Network, MAN) en ruggengraatnetwerken (Eng. Wide Area Network, WAN). Hoewel optische technologieën op elk van deze niveaus doordringen, zal de toepassing van OPS wellicht het snelst ingang vinden op MAN en WAN schaal: concurrentie met huidige erg kost-efficiënte LAN technologieën is moeilijk en toegangsnetwerken zijn vandaag nog zeer sterk gebaseerd op koper veeleer dan optische vezel.

De architecturen die worden voorgesteld op MAN-niveau onderstellen doorgaans eenvoudige topologieën: ze gebruiken een ster- of ring-configuratie. De motivatie hiervoor is dat de architectuur van een knoop in zo'n netwerk veel eenvoudiger—en daarom ook goedkoper—kan zijn dan wanneer men een willekeurig vermaasd netwerk zou willen construeren. De ster-gebaseerde netwerken vindt men terug in de aanpak van de zogenaamde Passieve Optische Netwerken (PON's). Op optisch niveau bevat het gros van de knopen geen enkele schakelfunctionaliteit: licht van verschillende golflengtes wordt samengenomen waar takken samenkomen, of gesplitst over verschillende uitgaande takken. Voorbeelden van PON-gebaseerde aanpakken zijn [72,73] en de eerder op toegangsnetwerken gerichte werken [74,75]. De voornaamse elementen uit een PON-architectuur zijn de optische lijnterminals (Eng. Optical Line Terminal, OLT) en optische netwerkknopen (Eng. Optical Network Unit, ONU). Deze laatste vormen de verbinding met de gebruikers van het netwerk via klassieke elektrische interfaces, terwijl de OLT-eenheden interconnetie met metro- of ruggengraatnetwerken zullen realiseren. Meerdere ONU's zullen via passieve splitters/combiners verbonden zijn met één enkele OLT. Merk op dat, hoewel zo'n PON pakketgebaseerd kan zijn, er niet echt sprake is van een OPS-netwerk, aangezien er geen schakel-elementen voorkomen en pakketten niet worden gerouteerd op basis van hun hoofding. De enige plaats waar van OPS sprake kan zijn is in een OLT.

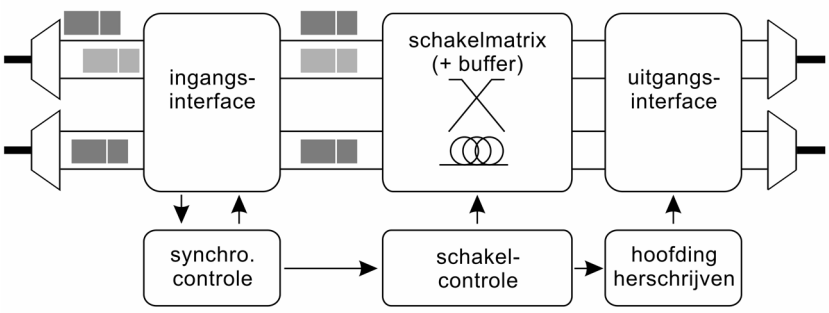
Ring-topologieën bieden een aantal inherente voordelen ten opzichte van stergebaseerde, waaronder een gemiddeld kortere vezellengte om een gegeven set knopen met elkaar te verbinden en een relatief eenvoudig te realiseren robuustheid tegen falingen zoals vezelbreuken [76]. Ring-gebaseerde OPS-architecturen werden onder meer voorgesteld in [6,77,78,79]. Sommige van de voorgestelde architecturen zijn enigszins verwant aan de PON-structuren met een centrale schakelknoop: één van de knopen op de ring heeft een speciale functie, wat zich onder meer uit in een complexere structuur. Zo'n centrale knoop zal dan bijvoorbeeld ook interconnectie van verschillende ringen voorzien [6]. In andere voorstellen zijn dan weer alle ringknopen equivalent [77–79] (maar ontbreekt bv. de mogelijkheid om meerdere optische ringen in eenzelfde knoop met elkaar te verbinden). Een studie die de complexiteit van de optische ringknopen afweegt tegenover de efficiëntie waarmee de aanwezige bandbreedte benut wordt, komt verder in dit proefschrift aan bod (zie Hoofdstuk 2, Sectie 2.2).

Het is duidelijk dat de beperkte optische functionaliteit van de knopen in dergelijke metro-netwerken zich zal vertalen in complexiteit op het logische (controle-)niveau. Het versturen van gegevens over zo'n netwerk zal worden geregeld door een aangepast MAC-protocol. Om verkeer tussen knopen op eenzelfde MAN uit te wisselen, zal aandacht besteed moeten worden aan het probleem van eerlijkheid [80]: er moet voor gezorgd worden dat één enkele knoop niet alle aanwezige bandbreedte inpalmt en daardoor andere knopen het verzenden van gegevens belet. Een typische oplossing hiervoor is het gebruiken van token-gebaseerde toegang [81,82]: een knoop kan dan slechts gegevens op de ring plaatsen wanneer hij een zogenaamd token in zijn bezit heeft, dat hij nadien zal doorgeven aan een volgende gegadigde. Wanneer het MAN opgesplitst is in verschillende deelnetwerken (bv. afzonderlijke ringen), zal ook de uitwisseling van gegevens tussen deze deelnetwerken efficiënt en eerlijk moeten verlopen. Een voorbeeld van MAC ontwerp voor geïnterconnecteerde ringen is te vinden in [71].

In een ruggengraatnetwerk zal men de eenvoudige topologieën niet langer handhaven. Men realiseert er doorgaans willekeurig vermaasde netwerken omdat die toelaten de propagatie-afstand tussen bron en bestemming te minimaliseren, eventueel de grootte van de schakelmatrices te optimaliseren, alsook de mogelijkheid bieden de belasting intelligent te verdelen (Eng. load balancing) over het netwerk. Als gevolg hiervan zal ook de functionaliteit van de knopen in dergelijke WAN's complexer zijn. Typisch vervult zo'n knoop de functie van kruisschakelaar: hij kan een willekeurige ingangspoort met een willekeurige uitgangspoort verbinden.

Een generieke structuur van een OPS-knoop is geschetst in Figuur 1.2. Die beslaat drie stadia [53]: (i) een ingangsinterface, (ii) een schakelmatrix en (iii) een uitgangsinterface. In het geval van slot-gebaseerd OPS zal de ingangsinterface synchronisatie-eenheden bevatten die inkomende pakketten aligneert. De belangrijkste functie van het eerste stadium van de OPS-schakelaar is het ontcijferen van de hoofding geassociëerd met elk pakket—een functie die ook in asynchrone concepten zoals OBS vervuld moet worden. Zoals eerder vermeld zal de hoofding informatie bevatten omtrent de bestemming van het pakket. Op die informatie zal de schakelknoop zich baseren om te beslissen naar welke uitgang het pakket zal worden doorgestuurd. De meerderheid van de OPS-concepten die tot op vandaag zijn

gedemonstreerd zet hiervoor de hoofding om van een optisch naar een elektronisch formaat [6,77,79,83,84] om de schakelmatrix te sturen. Zodoende tracht men met een pragmatische aanpak het beste van beide werelden te combineren: optische technologie omwille van bandbreedte- en schakelcapaciteiten, elektronische voor de controle- en beslissingsoperaties. Pas wanneer de functionaliteit van optische verwerking van de pakkethoofdingen rijper is, zal men de pakketten puur optisch kunnen schakelen.

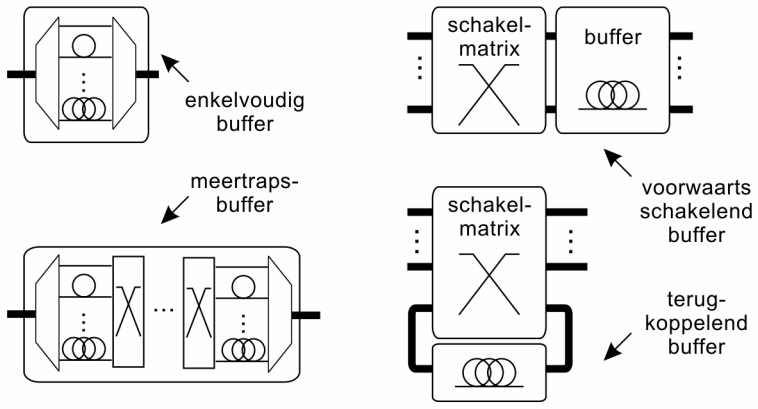


Figuur 1.2: Generieke structuur van een OPS knoop met illustratieve functies van de verschillende trappen.

De schakelmatrix in het centrale stadium realiseert de eigenlijke interconnectie van in- en uitgangen. Verschillende optische schakeltechnologieën zijn bekend. De meest tot de verbeelding sprekende zijn wellicht MEMS met microscopisch kleine twee-dimensionale opklappende [85] of zelfs drie-dimensionale kantelende [86] spiegeltjes. Voor optisch pakketschakelen zijn hun schakeltijden (in de orde van minstens enkele honderden microseconden of zelfs enkele milliseconden) evenwel te traag. Naast meer exotische voorstellen zoals bubblejet-technologie [87] en andere thermo-, electro-, of acousto-optische schakelaars [88–90], zijn de meer praktische en meest voorgestelde architecturen gebaseerd op zogenaamde "Arrayed Waveguide Gratings" (AWG) enerzijds [83,84,91] of optische versterkers in halfgeleider-technologie (Eng. Semiconductor Optical Amplifier, SOA) [92,93] anderzijds. Het passieve AWG-element heeft als kenmerkende eigenschap dat de uitgangspoort waarlangs licht het element verlaat, afhankelijk is van de frequentie, i.e. de golflengte, van dit licht. Door aan de ingangen van de AWG regelbare golflengte-convertoren te plaatsen, kan men bepalen aan welke uitgang inkomend licht wordt doorgeschakeld en op die manier een schakelaar realiseren. Het alternatief dat gebruik maakt van SOA's is een zogenaamde verdeel-en-selecteer-aanpak (Eng. broadcast-and-select): het optische signaal wordt door splitsers over de verschillende uitgangen verdeeld en de SOA wordt als schakelaar ingezet om per uitgang slechts het signaal van één ingangspoort door te laten. Een inherent voordeel van deze SOA-aanpak is dat multicast mogelijk is: hetzelfde ingangssignaal kan tegelijk naar verschillende uitgangen gezonden worden. In het voordeel van de AWG pleit dan weer dat de signaalkwaliteit aan de uitgang typisch beter is, door het ontbreken van splitsers die onontkoombaar zijn in de verdeel-en-selecteer-structuur.

Het laatste stadium van de schakelaar omvat de uitgangsinterface. Die zal moeten instaan voor regeneratie van het optische signaal, waar dit door de schakeloperatie wordt aangetast, en/of het (her)schrijven van de hoofding. Ook kan in dit stadium golflengte-conversie aanwezig zijn, vermits het een belangrijk hulpmiddel is om zogenaamde contentie op te lossen [94,95]. Die contentie treedt op wanneer op verschillende ingangsvezels gelijktijdig een pakket aanwezig is dat naar een zelfde uitgang verstuurd moet worden (waarbij de inkomende golflengte bovendien dezelfde kan zijn). Door één of meerdere pakketten naar een andere golflengte te converteren, wordt dit gelijktijdig verzenden over dezelfde uitgaande vezel mogelijk gemaakt.

De klassieke oplossing voor contentie is het voorzien van bufferruimte. Optisch kan dit, zoals eerder vermeld, gerealiseerd worden met behulp van vezelvertragingslijnen (FDL's). Net zoals bij elektrische schakelaars zijn er verschillende opties om buffering te voorzien [96]. Men onderscheidt voorwaarts schakelende (Eng. feed-forward) en terugkoppelende (Eng. feed-back) structuren, zoals geschetst in Figuur 1.3: in het eerste geval bevindt de buffer zich in lijn met de schakelmatrix (of is er eventueel mee verweven [92,97]), in het laatste is een deel van de uitgangspoorten van de schakelmatrix via FDL's verbonden met ingangspoorten zodat pakketten eventueel meermaals doorheen de buffer kunnen circuleren (bv. [6,84,98]). Verder kunnen de buffers gecatalogeerd worden in enkelvoudige en meervoudige stadia. Meervoudige bufferstadia worden doorgaans enkel in een voorwaarts schakelende configuratie gebruikt. In zo'n meervoudig buffer kan de vertragingstijd (i.e. de tijd die een pakket in de buffer doorbrengt) variëren: in elk stadium heeft men de keuze tussen verschillende vertragingen, zodat men met een beperkt aantal stadia toch een relatief wijd bereik aan vertragingen kan aanbieden en een FIFO buffer van bepaalde diepte implementeren [99]. In dit verband kunnen we ook opmerken dat sommigen suggereren om elektrische buffers te voorzien [100], mogelijks in combinatie met optische buffers [101]. Het mag evenwel duidelijk zijn dat dit, gezien de extra elektro-optische conversies en hoge snelheidvereisten voor het geheugen, de complexiteit van de knooparchitectuur niet ten goede komt.



Figuur 1.3: Illustratie van bufferstructuren.

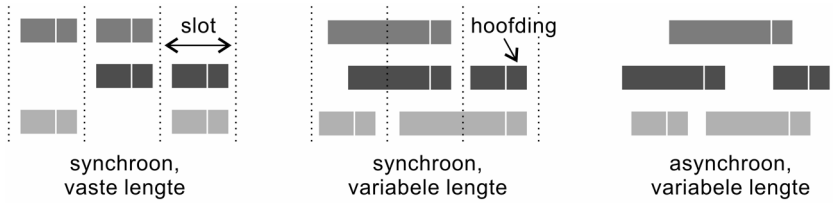
In een WAN zullen nagenoeg alle functies aangegeven in Figuur 1.2 aanwezig zijn, inclusief buffers, maar in een MAN zullen bepaalde onderdelen niet, of sterk vereenvoudigd, vertegenwoordigd zijn. In een PON-aanpak kan enkel de OLT-knoop enige gelijkenis vertonen—want randknopen zullen niet volledig optisch zijn—en dan bovendien zonder buffer. In ring-architecturen zal men in de ring-knopen evenmin optische buffers voorzien en kan de functionaliteit van de schakel-matrix beperkt worden tot zogenaamde add/drop-functies. Zoals eerder aangegeven zal in zulke bufferloze architecturen het gebruik van een aangepast MAC-protocol onontbeerlijk zijn.

1.3 Ontwerpskeuzes

De ontwerpskeuzes in OPS omvatten naast de technologische uitdagingen ook eerder conceptuele vragen. De eerste cruciale vraag is uiteraard hoe het optische pakket en bijhorende hoofding er precies uitzien. Daarnaast moet worden vastgelegd hoe het algoritme geconcipiëerd wordt om deze pakketten te schakelen.

1.3.1 Pakketformaat

Het idee dat aan de basis ligt van pakketschakelen is het opsplitsen van de te versturen gegevens in eenheden van beperkte lengte. Een eerste belangrijke keuze betreft het bepalen van deze lengte. In de elektrische wereld gebruikt het Internet Protocol (IP) pakketten van variabele lengte [102], terwijl men in de Asynchrone Transfer Mode technologie (ATM, [103]) gebruik maakt van eenheden van vaste lengte, die men er “cel” noemt. Ook in het optisch schakelen heeft men de keuze tussen vaste of variabele lengte pakketten, met in Figuur 1.4 de meest gangbare alternatieven.

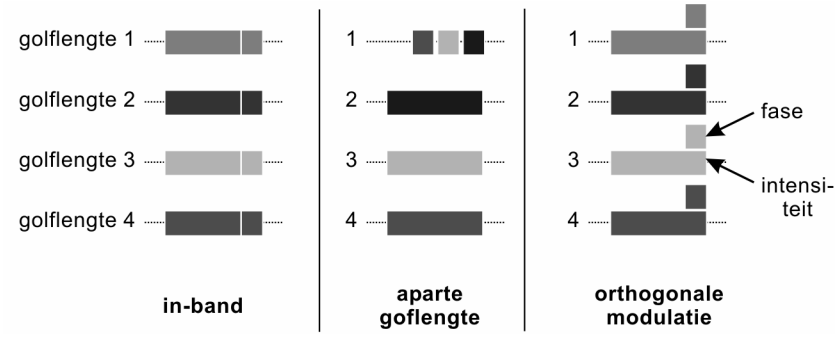


Figuur 1.4: De voornaamste alternatieven voor optisch pakketschakelen.

Het OBS concept hanteert, zoals eerder aangehaald, pakketten van variabele lengte die op asynchrone wijze worden geschakeld. Dit heeft als voordeel dat, wanneer de pakketten (of bursts) voldoende lang zijn, het absolute aantal pakketten nodig om een bepaalde hoeveelheid informatie te versturen relatief laag kan gehouden worden. Daardoor zal de belasting op de controle-eenheid in de schakelknopen beperkt gehouden worden: de schakelbeslissing wordt per pakket gemaakt en hangt niet af van de lengte ervan. Bovendien wordt dan slechts een minimale hoeveelheid bandbreedte besteed aan het versturen van de pakket-hoofding.

In OPS gebruikt men doorgaans synchrone pakketschakelaars die slot-gebaseerd schakelen. In zo'n geval ligt het voor de hand pakketten te hanteren van vaste lengte die binnen de duur van een slot vallen. Argumenten die in het voordeel pleiten van een slot-gebaseerd concept bestaan erin (i) dat het contentie-probleem efficiënter kan worden opgelost, want asynchroon schakelen verhoogt de verlieskans [104] (een fenomeen dat al eerder opdook bij vergelijken van Aloha met Slotted-Aloha in het domein van toegangsprotocollen voor gedeelde media), en (ii) dat aan de schakelmatrix minder strenge eisen worden gesteld qua blokkering, aangezien die matrix nu slechts zogenaamd herschikbaar niet-blokkerend (Eng. rearrangable non-blocking) moet zijn [97] (zie ook verder in Hoofdstuk 2). Het voornaamste minpunt van OPS met vaste lengte pakketten is uiteraard de hoge overlast die ontstaat doordat in elk slot een aparte hoofding moet worden meegestuurd en verwerkt. Om dit te verhelpen kan men, ondanks het slot-georiënteerde concept, toch werken met pakketten van variabele lengte. In dit geval wordt het pakket opgedeeld in eenheden van één slot lang, maar wordt enkel met het eerste slot een hoofding meegestuurd die de schakelbeslissing zal bepalen. Dat de volgende slots deel uitmaken van hetzelfde pakket kan bv. worden aangegeven met een zogenaamde "continuation bit" [105]. Die beslissing wordt steeds tegelijk genomen voor alle slots die samen het pakket vormen: men behandelt als het ware "treinen van slots" [106]. Een meer uitgebreide bespreking van de performantie van de twee OPS-varianten volgt in Hoofdstuk 3.

Doorgaans wordt voor de pakket-hoofding een lagere bitrate gebruikt dan voor de eigenlijke pakket-inhoud. In principe kan de pakket-inhoud zelfs een willekeurige bitrate hebben, aangezien deze in de schakelknopen niet be- of verwerkt dient te worden. Men spreekt in dit laatste geval van transparant optisch schakelen. Hoewel dit soms als een aantrekkelijk voordeel van optisch schakelen wordt voorgesteld, vereist zo'n aanpak een groot aantal adaptieve componenten [107] en semi-transparante netwerken die ontworpen zijn voor een gegeven signaalformaat en -bitrate kunnen dan ook (kost-)efficiënter zijn.



Figuur 1.5: Alternatieven voor het transporteren van de pakket-hoofding.

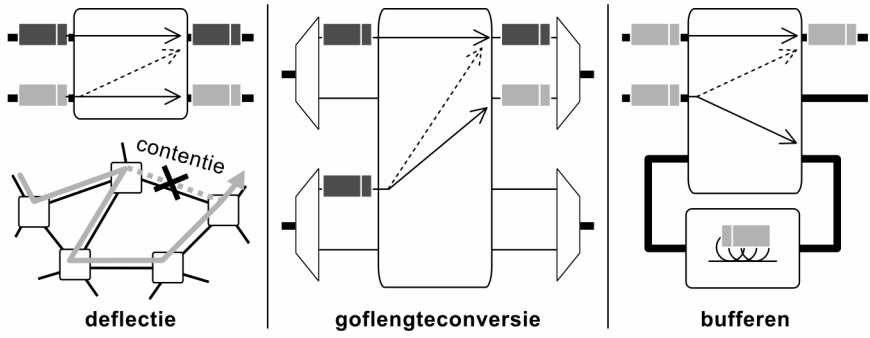
Voor het versturen van de hoofding zelf bestaan verschillende oplossingen. Een voor de hand liggende werkwijze is de hoofding vóór de pakketinhoud te zenden, op dezelfde golflengte, zoals geschetst in Figuur 1.5. Het voordeel hiervan is dat er

geen speciale maatregelen (zoals het compenseren van dispersie) getroffen moeten worden om pakket en bijhorende hoofding samen te houden. Een alternatieve aanpak is een aparte golflengte reserveren voor het doorsturen van de hoofdingen, wat typisch voorgesteld wordt in een OBS-context, waar de hoofding de naam "Burst Controle Pakket (BCP)" [108] of "Burst Hoofding Pakket (BHP)" [109] krijgt. Dit zorgt voor een erg eenvoudige afzondering van de hoofding om de schakelmatrix te sturen. Een aanpak die het voorzien van een aparte controle-golflengte vermijdt, is de hoofding in zogenaamde "sub-carrier multiplexing" (SCM) [79,110,111] of met behulp van een andere orthogonale modulatie-methode mee te sturen, zoals bv. FSK [83] of een combinatie van ASK en DPSK [112].

1.3.2 Schakel-algoritmes

Bij het eigenlijke doorschakelen van de pakketten vormt contentie het voornaamste probleem waaraan het hoofd moet worden geboden. Dit is het fenomeen waarbij op meerdere ingangen een pakket aankomt dat bestemd is voor dezelfde uitgaande poort, waarbij deze pakketten in de tijd overlappen. In het geval van optisch pakketschakelen zal de term "poort" slaan op een vezel die meerdere golflengtes in parallel draagt (cf. WDM).

Op drie manieren kan dit probleem worden opgelost [53]: (i) uitbuiten van de golflengte-dimensie door het toepassen van golflengte-conversie, (ii) buffering voorzien, en/of (iii) deflectie routering (Eng. deflection routing). In de eerste oplossing worden alle golflengtes op een vezel als equivalent beschouwd: het zijn parallele kanalen die naar dezelfde volgende schakelknoop leiden. Vermits in zo'n concept de precieze golflengte waarop een pakket wordt verstuurd geen betekenis[1] heeft, kunnen pakketten waarvoor contentie optreedt elk op een andere golflengte verstuurd worden door gepaste golflengte-conversie. (Bemerk dat wanneer de schakelaar 3R-regeneratie voorziet aan de uitgangen van de schakelmatrix, dit regeneratie-stadium de conversie kan verzorgen.)



Figuur 1.6: Mogelijkheden tot oplossen van contentie.

[1] In de zin dat ze niet bepalend is voor de schakelbeslissing, in tegenstelling tot zogenaamde golflengte-routering die bij optisch circuitschakelen voorgesteld wordt.

Optisch bufferen betekent dat men vezelvertragingslijnen zal voorzien om ervoor te zorgen dat de pakketten in de tijd gespreid, zonder overlappen, op de uitgaande vezel terecht kunnen komen. De verschillende mogelijke buffer-configuraties werden eerder besproken (zie Figuur 1.3) en vereisen extra schakel-elementen of een grotere schakelmatrix in vergelijking met bufferloze architecturen.

De zogenaamde deflectie-routering vereist geen bijkomende componenten en zal overtollige pakketten, waarvoor op de uitgaande vezel geen plaats meer is, naar een alternatieve uitgaande vezel sturen waar wel een golflengte vrij is. Het spreekt voor zich dat deze methode een bijzondere organisatie van de routering vereist, zodat een pakket ten allen tijde correct op de bestemming afgeleverd wordt en niet in een lus blijft rondcirkelen. Vergelijking van de drie methoden [98] voor het oplossen van contentie heeft trouwens uitgewezen dat deflectie enkel werkt wanneer het netwerk niet te zwaar belast wordt (cf. er moeten voldoende "alternatieve" uitgaande vezels licht belast zijn om overtollige pakketten op te vangen).

Een geschikt algoritme zal moeten beslissen welke pakketten in geval van contentie worden weerhouden om te worden doorgeschakeld en/of naar een buffer gestuurd wanneer er FDL's voorzien zijn. In het geval van buffers die een variabele vertraging kunnen aanbieden, zal het algoritme ook moeten instaan voor de beslissing over de duur van de vertraging van gebufferde pakketten. Wanneer er geen buffers zijn in de optische schakelknopen zal ofwel de belasting voldoende laag gehouden moeten worden (zodat het contentie-probleem niet optreedt), ofwel zal men contentie moeten vermijden door de toegang tot het OPS-netwerk via een gepast MAC-protocol te regelen (zie ook hoger).

Bij het ontwerp en de evaluatie van dergelijke schakel-algoritmen, zoals die in Hoofdstuk 3 in detail besproken worden, zal vanzelfsprekend ook aandacht besteed moeten worden aan differentiatie in de kwaliteit van dienstverlening [113] (Eng. Quality of Service, QoS). Het OPS-netwerk zal immers sterk heterogeen verkeer dragen, met uiteenlopende vereisten in termen van vertraging en pakketverlies.

1.4 Structuur doctoraat

Bovenstaand inleidend materiaal is gedeeltelijk gebaseerd op onze publicaties [23,31] waarvan de laatste terug te vinden is in Appendix A. In het vervolg van dit proefschrift zullen we een coherent overzicht trachten te geven van de originele bijdragen die we geleverd hebben in het domein van OPS. Hierbij zullen we vanzelfsprekend niet alle geboekte resultaten uitvoerig aan bod laten komen, maar veeleer een synthese presenteren waarbij we voor details verwijzen naar publicaties uit bijgevoegde selectie.

In het eerstvolgende hoofdstuk belichten we de onderzoeksresultaten met betrekking tot knoop-architecturen voor OPS. Een eerste deel daar behelst een evaluatie van alternatieven voor ring-gebaseerde metro-netwerken. In het bijzonder beantwoorden we de vraag of ruimtelijk hergebruik (Eng. spatial reuse) zin heeft en of het een goed idee is zogenaamde golflengtebanden te hanteren. Een tweede deel spitst zich toe op de vraag hoe schakelmatrices zoals die typisch worden voorgesteld voor OPS-ruggengraatnetwerken geschaald kunnen worden naar grote aantallen in- en uitgangspoorten. Dit laatste deel heeft aanleiding gegeven tot de publicatie van

onze werken [9,11,26] (laatstgenoemde is gereproduceerd in Appendix C). Het eerste deel omtrent metro-netwerken wordt voorgesteld in [30] (na te lezen in Appendix B) en [33].

In Hoofdstuk 3 behandelen we de schakel-algoritmes die we ontwierpen en evalueerden voor optische schakelaars met een teruggekoppelde FDL-buffer. Daarbij bespreken we zowel het geval van pakketten van vaste lengte als variabele lengte. Het mag duidelijk zijn dat we ons hierbij richten op de logische performantie van deze algoritmes en in het bijzonder aandacht besteden aan het (beperken van) pakketverlies. Details van deze studies zijn terug te vinden in [20,24,25] voor pakketten van vaste lengte, waarvan we de eerste twee hebben opgenomen in Appendices D en E. Pakketten van variabele lengte worden behandeld in de volgende Appendices F en G (resp. [32] en [29]).

Wanneer we aldus de architectuur van afzonderlijke knopen en hun performantie besproken hebben, gaan we in Hoofdstuk 4 over tot het beschouwen van een volledig netwerk. Meer bepaald bedenken we een algoritme dat de routering van een gegeven verkeerspatroon bepaalt voor een gegeven netwerk, met het oog op het minimaliseren van het pakketverlies. Dit werk wordt uitvoerig behandeld in onze publicatie [7] die vervat is in Appendix H.

Ter afronding van de thesis klimmen we in Hoofdstuk 5 op tot net onder de applicatie-laag en bespreken de effecten van schakelacties, die in het onderliggende optische netwerk genomen worden, op het gedrag van TCP-stromen. Een deel van de resultaten van die studie werden voorgesteld in [3,16] en verder uitgewerkt in [4] die in Appendix I is opgenomen.

1.5 Lijst van publicaties

Publicaties in internationale tijdschriften:

[1] D. Colle, P. Van Heuven, **C. Develder**, S. Van den Berghe, I. Lievens, M. Pickavet, P. Demeester, “MPLS recovery mechanisms for IP-over-WDM networks”, Photonic Network Commun. (PNET), Kluwer Academic Publishers, vol. 3, no. 1/2, Jan. 2001, pp. 23–40.

[2] S. De Maesschalck, D. Colle, A. Groebbens, **C. Develder**, A. Lievens, P. Lagasse, M. Pickavet, P. Demeester, F. Saluta, M. Quagliatti, “Intelligent optical networking for multilayer survivability”, IEEE Commun. Mag. , vol. 40, no. 1, Jan. 2002, pp. 42–49.

[3] D. Colle, S. Demaesschalck, **C. Develder**, P. Van Heuven, A. Groebbens, J. Cheyns, I. Lievens, M. Pickavet, P. Lagasse, P. Demeester, “Data-centric optical networks and their survivability”, IEEE J. Selected Areas in Commun., vol. 20, no. 1, Jan. 2002, pp. 6–20.

[4] **C. Develder**, D. Colle, S. Demaesschalck, M. Pickavet, P. Demeester, “Influence of GMPLS recovery mechanisms on TCP performance”, Photonic Network Commun. (PNET), vol. 4, no. 3/4, Jul.-Dec. 2002, pp. 321–345.

[5] J. Cheyns, **C. Develder**, E. Van Breusegem, E. Baert, A. Ackaert, M. Pickavet, P. Demeester, “Routing in an AWG based optical packet switch”, Photonic Network Commun. (PNET), Kluwer Academic Publishers, vol. 5, no. 1, Jan. 2003, pp. 69–80.

[6] L. Dittmann (ed.), **C. Develder**, D. Chiaroni, F. Neri, F. Callegati, W. Koerber, A. Stavdas, M. Renaud, A. Rafel, J. Solé-Pareta, W. Cerroni, N. Leligou, L. Dembeck, B. Mortensen, M. Pickavet, N. Le Sauze, M. Mahony, B. Berde, G. Eilenberger, "The European IST project DAVID: a viable approach towards optical packet switching", IEEE J. Selected Areas in Commun., vol. 21, no. 7, Sep. 2003, pp. 1026–1040.

[7] E. Baert, **C. Develder**, M. Pickavet, P. Demeester, "Routing strategies to minimize packet loss in an optical packet switched network with recirculating FDL buffers", zal verschijnen in Phot. Netw. Commun. (PNET), vol 7, no. 2, Mar. 2004.

[8] **C. Develder**, B. Van Houdt, C. Blondia, M. Pickavet, P. Demeester, "Analytical MMAP-based bounds for packet loss in optical packet switching with recirculating FDL buffers", submitted to PNET.

[9] J. Cheyns, **C. Develder**, E. Van Breusegem, E. Baert, D. Colle, M. Pickavet, P. Demeester, "Clos lives on in Optical Packet Switching", aanvaard voor publicatie in IEEE Commun. Mag.

[10] F. Callegati, **C. Develder**, W. Cerroni, M. Pickavet, P. Demeester, "Scheduling algorithms for a slotted packet switch with either fixed or variable length packets", aanvaard voor publicatie in Photonic Network Commun. (PNET), Kluwer Academic Publishers.

[11] C. Matrakidis, **C. Develder**, A. Stavdas, M. Pickavet, P. Demeester, "Multistage switches for optical packet switching based on SOA technology", to be submitted to Optical Networks Magazine.

Publicaties op internationale conferenties:

[12] P. Demeester, D. Colle, S. Demaesschalck, **C. Develder**, M. Pickavet, P. Van Heuven, "Resilience in IP over WDM based multilayer networks", (Invited) Proc. 26st European Conf. on Optical Commun. (ECOC 2000), Munich, Germany, 3-7 Sep. 2000.

[13] D. Colle, **C. Develder**, P. Van Heuven, M. Pickavet, P. Demeester, L. Raptis, G. Chatzilias, C. Mas, Y.I. Manolessos, J. Comellas, A. Rafel, J. Prat, J. Solé-Pareta, J. Moyano, S. Brunazzi, S. Rotolo, R. Stankiewicz, A. Gladish, "Recovery techniques for IP-over-WDM networks", (Invited) IP over DWDM Conf., Paris, France, 27-30 Nov. 2000.

[14] D. Colle, P. Van Heuven, A. Groebbens, **C. Develder**, M. Pickavet, P. Demeester, "Dimensioning reliable IP-over-WDM networks", IP over DWDM Conf., Paris, France, 27-30 Nov. 2000.

[15] P. Demeester, D. Colle, S. De Maesschalck, **C. Develder**, M. Pickavet, P. Van Heuven, "Resilience in IP over WDM based multilayer networks", (Invited) Optical Networks 2000, Göteborg, Sweden, 7 Dec. 2000.

[16] **C. Develder**, Didier Colle, Pim Van Heuven, Steven Van den Berghe, Mario Pickavet, Piet Demeester, "Influence of recovery time on TCP behaviour", MPLS World Congress, Paris, France, 6-9 Feb. 2001.

[17] D. Colle, **C. Develder**, P. Van Heuven, S. Demaesschalck, A. Groebbens, M. Pickavet, P. Demeester, "Resilience in IP-over-WDM networks", (Invited) Proc. 5th Working Conf. on Optical Netw. Design and Modelling (ONDM

2001) , Vienna, Austria, 5-7 Feb. 2001.

[18] A. Groebbens, D. Colle, **C. Develder**, S. De Maesschalck, M. Pickavet, P. Demeester, "Use of backup trees to improve resource efficiency of MPLambdaS protection mechanisms", Proc. 3rd Intl. Workshop on Design of Reliable Commun. Netw. (DRCN 2001), Budapest, Hungary, 7-10 Oct. 2001, pp. 152–159.

[19] D. Colle, S. De Maesschalck, **C. Develder**, A. Groebbens, M. Pickavet, P. Demeester, "Design and dimensioning issues in survivable data-centric optical networks", (Invited) Proc. 1st COST270 Workshop on Reliability of Optical Netw., Systems and Components, Dubendorf, Switzerland, 12-13 Dec. 2001.

[20] **C. Develder**, M. Pickavet, P. Demeester, "Assessment of packet loss for an optical packet router with recirculating buffer", Proc. 6th IFIP Working Conf. on Optical Netw. Design and Modelling (ONDM2002), Torino, Italy, 4-6 Feb. 2002, pp. 247–261.

[21] M. Pickavet, A. Ackaert, E. Baert, J. Cheyns, D. Colle, S. De Maesschalck, P. Demeester, **C. Develder**, A. Groebbens, I. Lievens, E. Van Breusegem, S. Verbrugge, Q. Yan, "Design of communication networks using heuristics", (Invited) Book of abstracts of "Arbeitskreis Mathematik in Forschung und Praxis", 23rd Symp. on Mathematik in der Telekommunikation, Duisburg, Germany, 18-19 Mar. 2002.

[22] M. Pickavet, **C. Develder**, E. Baert, P. Demeester, "A.I. Techniques for planning telecommunication networks", Proc. Intl. Conf. on Artificial Intelligence (IC-AI'02), Las Vegas, NV, 24-27 Jun. 2002.

[23] **C. Develder**, J. Cheyns, E. Van Breusegem, E. Baert, A. Ackaert, M. Pickavet, P. Demeester, "Node architectures for optical packet and burst switching", Tech. Digest Int. Topical Meeting on Photonics in Switching (PS2002), (invited) paper PS.WeA1, Cheju Island, Korea, 21-25 Jul. 2002, pp. 104–106.

[24] **C. Develder**, M. Pickavet, P. Demeester, "Strategies for an FDL based feed-back buffer for an optical packet switch with QoS differentiation", Proc. Conf. on Optical Internet (COIN2002), paper COIN.TuD1, Cheju Island, Korea, 21–25 Jul. 2002, pp. 114–116.

[25] **C. Develder**, M. Pickavet, P. Demeester, "Choosing an appropriate buffer strategy for an optical packet switch with a feed-back FDL buffer", Proc. 28th European Conf. on Optical Commun. (ECOC2002), Copenhague, Denmark, 8-12 Sept. 2002.

[26] **C. Develder**, J. Cheyns, M. Pickavet, P. Demeester, "Multistage architectures for optical packet switching using SOA-based broadcast-and-select switches", Tech. Digest Optical Fibre Conf. (OFC 2003), paper FS4, Atlanta, GA, USA, 23-28 Feb. 2003, pp. 794–795.

[27] J. Cheyns, E. Van Breusegem, **C. Develder**, A. Ackaert, M. Pickavet, P. Demeester, "Performance improvement of an internally-blocking optical packet/ burst switch", Proc. Intl. Conf. on Commun. (ICC2003), Anchorage, AK, 11-15 May 2003, vol. 2 , pp. 1304–1308.

[28] D. Colle, J. Cheyns, **C. Develder**, E. Van Breusegem, A. Ackaert, M.

Pickavet, P. Demeester, B. Feng, H. Lønsethagen, E. Zouganeli, J. Fernández-Palacios, J.F. Lobo, "GMPLS extensions for supporting advanced optical networking technologies", Proc. Intl. Conf. on Transparent Optical Networks (ICTON 2003), Warschau, Poland, 29 Jun. – 3 Jul. 2003.

[29] **C. Develder**, M. Pickavet, P. Demeester, "On trains and wagons: switching variable length packets in a slotted OPS network", Proc. Conf. on Optical Internet / Australian Conf. on Optical Fibre Technology (COIN/ACOFT 2003), paper TuA2-6, Melbourne, Australia, 13-16 Jul. 2003, pp. 233–236.

[30] **C. Develder**, R. Van Caenegem, E. Baert, M. Pickavet, P. Demeester, "Active versus passive OPS architectures for metro rings: a network dimensioning point of view", Proc. 29th European Conf. on Optical Commun. (ECOC 2003), paper Mo.4.4.1, Rimini, Italy, 21-25 Sep. 2003, vol. 1, pp. 100–103.

[31] **C. Develder**, J. Cheyns, E. Van Breusegem, E. Baert, D. Colle, M. Pickavet, P. Demeester, "Architectures for optical packet and burst switches" (invited), Proc. 29th European Conf. on Optical Commun. (ECOC 2003), paper We.1.4.4, Rimini, Italy, 21-25 Sep. 2003, vol. 3, pp. 376–377.

[32] **C. Develder**, J. Cheyns, M. Pickavet, P. Demeester, "Service differentiation mechanisms for variable length packets in an optical switch with recirculating FDL buffer", Tech. Digest Photonics in Switching (PS2003), paper PS.Mo.C8,Versailles, France, 28 Sep.-2 Oct. 2003, pp. 92–94.

[33] R. Van Caenegem, **C. Develder**, E. Baert, D. Colle, M. Pickavet, P. Demeester, "Architectures for OPS metro rings: comparing active versus passive nodes / A dimensioning point of view", aanvaard voor 8th IFIP Working Conf. on Optical Network Design and Modelling (ONDM 2004).

[34] J. Cheyns, E. Van Breusegem, **C. Develder**, D. Colle, M. Pickavet, P. Demeester, "Evaluating cost functions for OPS node architectures / A packaging driven approach", aanvaard voor 8th IFIP Working Conf. on Optical Network Design and Modelling (ONDM 2004).

Publicaties op nationale symposia:

[35] D. Colle, **C. Develder**, S. De Maesschalck, A. Groebbens, M. Pickavet, P. Demeester, "Design and planning of reliable communication networks", 1st FTW PhD Symposium, Gent, Belgium, 5 Dec. 2000.

[36] **C. Develder**, J. Cheyns, E. Van Breusegem, E. Baert, M. Pickavet, P. Demeester, "Optical Packet/Burst Switching", 2nd FTW PhD Symposium, Gent, Belgium, 12 Dec. 2001.

[37] **C. Develder**, E. Baert, M. Pickavet, P. Demeester, "Optical Packet Switched Networks with Recirculating FDL Buffers", 3rd FTW PhD Symposium, paper 9 (Proceedings available on CD-Rom), Interactive poster session, 11 Dec. 2002, Gent, Belgium.

[38] A. Groebbens, **C. Develder**, D. Colle, S. Demaesschalck, B. Lannoo, M. Pickavet, P. Demeester, "Optimization of backup tree structures, reducing spare capacity in optical networks, while retaining protection speed", 3rd FTW PhD Symposium, Gent, Belgium, 11 Dec. 2002.

Bijdragen tot internationale onderzoeksrapporten:

[39] R. Inkret, A. Kuchar, B. Mikac (eds.), E. Baert, S. Bauer, S. Bjørnstad, C. Bungarzenau, F. Callegati, D. Careglio, J. Cheyns, T. Cinkler, D. Colle, S. Demaesschalck, P. Demeester, **C. Develder**, V. Eramo, C. Gauger, D.R. Hjelme, G. Junyent, M. Karasek, M. Klinkowski, M. Köhn, M. Kowalewski, M. Lakovic, M. Ljolje, M. Marciniak, C. Mas, X. Masip-Bruin, F. Matera, M. Mattiello, C. Mauz, H.-P. Nolting, M. Nord, M. Pickavet, B. Puype, C. Raffaelli, S. Sanchez-Lopez, B. Sartorius, D.A. Schupke, M. Settembre, J. Solé-Pareta, N. Stol, S. Tomić, I. Tomkos, A. Tzanakak, S. Verbrugge, Q. Yan, I. Zacharopoulos, P. Zaffoni, "Advanced Infrastructure for photonic networks / Extended final report of Cost 266 action", 2003.

Hoofdstuk 2
Knooparchitecturen

2.1 Situering

In het voorgaande inleidende hoofdstuk hebben we een kort overzicht gegeven van de verschillende functies die een OPS-schakelaar moet vervullen. In dit hoofdstuk zullen we enkele specifieke implementaties behandelen. Een eerste studie in Sectie 2.2 tracht een oordeel te vellen over een architectuur voor een OPS-schakelaar bestemd voor ringvormige metro-netwerken. We zullen er de zin van ruimtelijk hergebruik en een golflengteband-concept onderzoeken.

In ruggengraatnetwerken is de voorgestelde architectuur typisch geavanceerder dan voor metro-netwerken en situeert de toepassing zich op langere termijn [6]. In deze context bestuderen we in Sectie 2.3 hoe pakketschakelende knopen van hoge capaciteit geconstrueerd kunnen worden door een meertrapsarchitectuur te gebruiken.

De eigen publicaties waar de in dit hoofdstuk samengevatte resultaten in detail worden besproken zijn: [30] (zie Appendix B) en [33] voor sectie 2.2 en voor sectie 2.3 hoofdzakelijk [26] uit Appendix C, alsook [9,11] en in mindere mate [23,31].

2.2 Metro-netwerken

Zoals aangegeven in het inleidend hoofdstuk, loopt een typische verbinding tussen twee communicatie-eindpunten doorheen drie netwerksegmenten: het toegangsnetwerk, een metro-netwerk (MAN) en een ruggengraatnetwerk (Eng. Wide Area Network, WAN). Het toegangsnetwerk zal instaan voor het verzamelen van verkeer van eindgebruikers en beslaat doorgaans hooguit een tiental kilometer. Bij een toegangsnetwerk is de kostprijs de meest dominante factor en zal men het netwerk zo eenvoudig mogelijk houden. Vandaar dat men hiervoor vandaag typisch koper-gebaseerde oplossingen (xDSL over telefoonlijnen, coaxiale kabel) vindt, of puur passieve optische netwerken [74,75] waarbij de optische vezel tot in of aan de huizen van de gebruikers kan lopen (Eng. Fiber To The Home, FFTH) [114,115]. Het metro-netwerk zal instaan voor de uitwisseling van informatie tussen verschillende toegangsnetwerken en de verbinding verzorgen met het ruggengraat-netwerk dat informatie over lange afstand (enkele honderden tot duizenden

kilometers) transporteert. De bandbreedte die in een MAN omgaat is dus van een hogere orde dan die in het toegangsnetwerk (dat voornamelijk de functie vervult van aggregatie van verkeersstromen). Hierbij is het nuttig op te merken [79] dat een niet onbelangrijk deel van dit verkeer binnen hetzelfde MAN blijft, zonder via het ruggengraatnetwerk te passeren (zie bv. de traffiek-patronen gebaseerd op die van netwerkoperator Telefonica in Spanje [116]). Het metro-netwerk krijgt, vergeleken met het ruggengraatnetwerk, te maken met een verkeerspatroon dat erg in tijd en ruimte fluctueert. Daarenboven moet het diensten van uiteenlopende aard kunnen ondersteunen, gaande van gegevensstromen met gematigde vereisten qua netwerk-vertraging (bv. voor Internet-surfers) tot toepassingen met strikte kwaliteitseisen (bv. bedrijfscommunicatie, videofonie). Een optisch pakketgeschakelde aanpak is in een dergelijke dynamische omgeving, die hoge bandbreedte-eisen stelt, zeer zinvol [117]. De metro-omgeving wordt dan ook gezien als de plek bij uitstek waar OPS op relatief korte termijn zijn nut kan bewijzen [6,118].

2.2.1 Alternatieven en motivatie

Een OPS-architectuur voor een metro-netwerk zal de concurrentie moeten aangaan met recente technologieën zoals Ethernet-gebaseerde optische netwerken tot zelfs 10 Gbit/s (realiseerbaarheid aangetoond in experimenten [119] en intussen in standaard IEEE 802.3ae vastgelegd [120]), of robuuste pakket-ringen (Eng. Resilient Packet Rings, RPR [121], momenteel onder standardisatie in IEEE 802.17). Vermits deze technologieën relatief goedkoop zijn, is het aangewezen de complexiteit van een OPS-alternatief beperkt te houden. Kost is hier immers een belangrijk criterium [122]. De recentste voorbeelden van op OPS-gebaseerde metro-architecturen spelen daar dan ook op in en stellen ring-gebaseerde netwerken met bufferloze schakel-knopen voor (zie bv. [77,78,79,118,123,124]).

In het Hornet-project [79] doet men beroep op de pragmatische combinatie van elektronica voor controle en optische componenten voor het eigenlijke transport van de bits en bytes. Op de vezelring worden verschillende golflengten gebruikt (in WDM), waarbij elke knoop op slechts één enkele golflengte gegevens kan ontvangen. Uiteraard kan die knoop wel alle golflengtes aanspreken om gegevens te verzenden, met behulp van een snel afstembare transmitter. Wanneer meerdere knopen dezelfde ontvangstgolflengte gebruiken, is het mogelijk dat een pakket niet rechtstreeks tot bij de uiteindelijke bestemmeling gezonden wordt: in een tussen-liggende knoop die dezelfde golflengte ontvangt als de finale ontvangende knoop zal het pakket van de ring gehaald worden en er nadien opnieuw op worden geplaatst (via een elektronisch buffer). De hoofding van het pakket wordt in Hornet via FSK-modulatie meegestuurd (zie Hoofdstuk 1).

Een ander project, op Japanse bodem, gebruikt een 2×2 schakelaar om pakketten op/van een metro-ring te brengen/halen. In plaats van een orthogonale modulatie-techniek, wordt hier een aparte controle-golflengte benut. Zoals voor-gesteld in [123], is er ook slechts één enkele golflengte voor het verzenden van de eigenlijke pakket-inhoud. Het is evenwel duidelijk dat het concept met een grotere schakelmatrix kan worden uitgebreid naar een WDM-ring met meerdere parallelle data-kanalen. De toegang wordt geregeld door een zogenaamde meesterknoop die lege slots op de ring brengt, waarbij aangeduid wordt welke bronknoop dit slot mag

gebruiken. Zogenaamd ruimtelijk hergebruik wordt toegelaten om de bandbreedte zo goed mogelijk te benutten: lege slots mogen ook gebruikt worden voor het verzenden van pakketten tussen twee knopen die gelegen zijn tussen de meesterknoop en de bronknoop waarvoor het lege slot gereserveerd is.

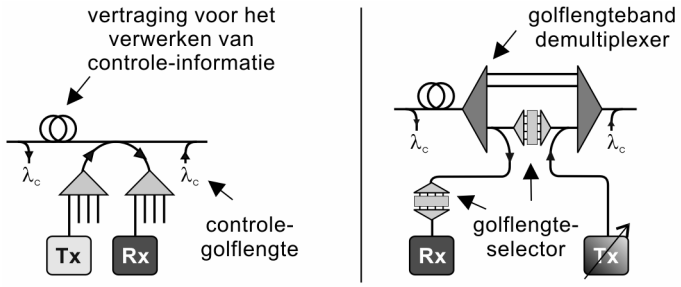
Ook door Europese onderzoekers werden ring-gebaseerde metro-netwerken voorgesteld, die conceptueel grote gelijkenissen vertonen met de zonet behandelde [77,78].

Louter op vlak van functionaliteit kunnen we twee knoop-architecturen onderscheiden: een Actieve Knoop en een Passieve Knoop. In de studie die nu volgt zullen we zo'n actieve en passieve structuur met elkaar vergelijken. De knoop-architecturen werden voorgesteld in [6,118]. Ze gebruiken een aparte golflengte als kanaal voor controlesignalen. Via dit kanaal zal toegang tot het netwerk geregeld worden door een MAC-protocol [6,71]. Die toegang gebeurt op een slot-gebaseerde wijze: het betreft een synchroon OPS-netwerk met pakketten van vaste lengte. (Merk op dat synchronisatie in een ring-structuur, waar alle data-kanalen via dezelfde vezel verstuurd worden, vrij eenvoudig te realiseren is, zeker in vergelijking met willekeurig vermaasde netwerk-structuren.)

De Passieve Knoop heeft als kenmerk dat hij niet in staat is een datastroom fysisch te termineren: het licht kan wel gedeeltelijk afgesplitst worden om de informatie te verwerken (en door te schakelen naar het toegangsnetwerk dat verbonden is met het ring-gebaseerde MAN), maar dat belet het niet verder op de ring te propageren. Met zo'n knoop komt men tot een netwerkarchitectuur die beduidend afwijkt van voorgaande geciteerde voorbeelden. Het concept dat men met zo'n passieve architectuur kan realiseren, betitelt men in [124,125] als DBORN, wat staat voor "Dual Bus Optical Ring Network". Kenmerkend is dat het golflengte-spectrum gescheiden wordt voor zenden en ontvangen. Een centrale knoop, die men Hub noemt, staat in voor de golflengte-conversie van het zend- naar het ontvangst-spectrum. Deze Hub kan ook meerdere MAN-ringen met elkaar verbinden. De "gewone" ringknopen zelf hebben de erg eenvoudige structuur die geschetst is in het linker deel van Figuur 2.1: via splitsers en koppelaars wordt licht van de vezel getapt, resp. erin gekoppeld, om zogenaamde add/drop-functionaliteit te voorzien. Er zijn geen actieve schakelcomponenten in het lichtpad dat de knoop doorkruist. Door zijn eenvoud kan die architectuur gerealiseerd worden met technologie die vandaag al op punt staat en zelfs commercieel beschikbaar is. Om de capaciteit van de knoop uit te breiden, kunnen extra zend- of ontvangstelementen (Tx, resp. Rx) toegevoegd worden.

De Actieve Knoopstructuur die we beschouwen is weergegeven in de rechtse helft van Figuur 2.1. Het binnenkomende lichtsignaal wordt, na de klassieke aftapping van het controle-signaal, naar een golflengteband-demultiplexer gezonden. Deze splitst het licht in verschillende spectrale delen, de golflengtebanden. In onze studie beschouwen we golflengtebanden die elk vier afzonderlijke golflengten bevatten. Een knoop heeft toegang tot ten minste één zo'n band, waarvoor er precies één zendende laser is met een snel instelbare golflengte en één ontvanger. (In Figuur 2.1 is de ontvanger golflengte-ongevoelig en wordt er via een SOA-schakelrij slechts één golflengte tegelijk doorgelaten.) In het doorgaande pad, tussen de golflengteband-demultiplexer en -multiplexer, wordt het licht afgesplitst dat

bestemd is voor de ontvanger (Rx) en wat afkomstig is van de zender (Tx) wordt erin gekoppeld. Het cruciale verschil met de passieve architectuur is dat er tussen dit passief splitsen en koppelen een rij van SOA-schakelaars geplaatst is. Deze worden gebruikt om pakketten die voor de betreffende knoop bestemd zijn niet langer te laten verder propageren op de ring. Het geheel van Rx, Tx en de SOA-elementen die voor een bepaalde band voorzien zijn, kunnen geïntegreerd worden op wat we een baby-bord noemen [126]. De capaciteit van de knoop kan verhoogd worden door meerdere baby-borden te installeren en dus tot meerdere golflengtebanden toegang te bieden.

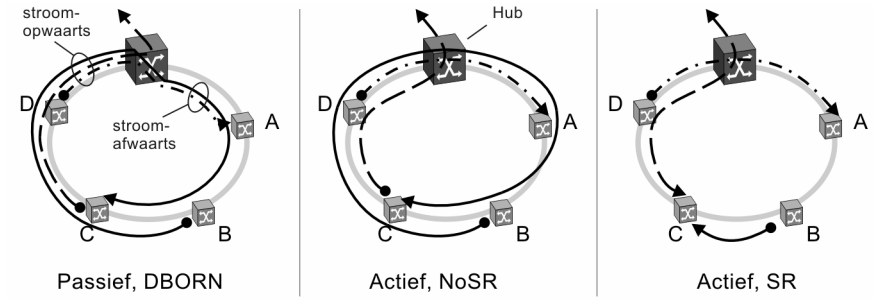


Figuur 2.1: Passieve vs Actieve Knoopstructuur.

Met een Actieve Knoopstructuur kan men, door de mogelijkheid pakketten daadwerkelijk van de ring te halen (in tegenstelling tot DBORN, waar enkel de centrale Hub daartoe in staat is), het eerder vermelde ruimtelijk hergebruik (Eng. spatial reuse, SR) uitbuiten. Dit concept impliceert dat eenzelfde golflengte op de ring gelijktijdig voor meerdere connecties kan worden benut. Dit kan zelfs wanneer die elk afzonderlijk de capaciteit van een volledige golflengte vereisen, zoals voor een connectie tussen knopen B en C enerzijds en D en A anderzijds, in het uiterst rechtse deel van Figuur 2.2. In die figuur schetsen we de drie netwerkconcepten die we verder met elkaar zullen vergelijken. Het eerste is DBORN, met de intussen welbekende spectrale scheiding van zenden en ontvangen. Een tweede maakt gebruik van de Actieve Knoopstructuur, maar onderstelt een gecentraliseerd MAC-protocol waar al het verkeer doorheen de Hub moet passeren. Dit impliceert dat sommige verkeersstromen eenzelfde linksegment meermaals moeten doorkruisen, zoals de stroom tussen B en C in Figuur 2.2. Wanneer het MAC-protocol volledig gedistribueerd is, kan die beperking van een centraal doorgangspunt opgeheven worden. (Merk op dat dit de complexiteit van het toegangsprotocol aanzienlijk kan verhogen, zeker wanneer de zogenaamde “eerlijkheid” gegarandeerd moet worden zodat één knoop de toegang tot het netwerk niet zou monopoliseren en evenmin de positie van de knoop in de ring z’n bandbreedtegebruik zou beïnvloeden.) In dat geval is ruimtelijk hergebruik mogelijk en duiden we de netwerk-architectuur aan met SR, terwijl het voorgaande geval NoSR genoemd zal worden.

De bedoeling—en meteen ook de originele bijdrage—van deze studie is de drie architecturen met elkaar te vergelijken in termen van de hoeveelheid middelen die nodig zijn om een gegeven hoeveelheid verkeer op te zetten. De drijfveer voor deze studie is te achterhalen welke architectuur de meest kost-efficiënte is: de kostprijs

(i.e. kapitaal, Eng. capital expenditure, CAPEX) van een netwerk dat aan een reële vraag moet beantwoorden zal immers sterk gerelateerd zijn aan de hoeveelheid middelen.



Figuur 2.2: Drie netwerkconcepten: (i) DBORN met de Passieve Knoop, (ii) Actieve Knoop zonder ruimtelijk hergebruik, NoSR en (iii) met ruimtelijk hergebruik, SR.

2.2.2 Probleembeschrijving en oplossingsmethodiek

Het probleem dat we moeten oplossen voor de vergelijking van de onderscheiden architecturen betreft een netwerk-planningsprobleem:

- ***Gegeven:***
  - Een verzameling knopen V die een metro-netwerk zal vormen (dat eventueel meerdere ringen kan beslaan);
  - Een verzameling (kandidaat-)linken tussen elementen v en w van V;
  - Een twee-dimensionale vraagmatrix D, waarbij elke rij i en elke kolom j overeenstemt met een knoop v, resp. w uit V zodat het element $D_{i,j}$ aangeeft hoeveel bandbreedte er van v naar w verzonden moet worden.
- ***Gevraagd:***
  - Welke knopen zullen door middel van welke verbindingen verbonden worden tot metro-ringen?
  - Hoeveel capaciteit zal daarvoor moeten worden geïnstalleerd? (Hierbij wordt met capaciteit bedoeld: het aantal gebruikte golflengten op de linken en de zend- en ontvangstcapaciteit in de ringknopen.)

Om dit probleem op te lossen hebben we—vermits de oplossing van een ILP formulering van het probleem te tijdrovend bleek—gebruik gemaakt van heuristieken geïnspireerd op het werk voorgesteld in Hoofdstuk 6 van [76].

We hebben de oplossing onderverdeeld in twee stappen. Een eerste stap omvatte het bepalen van de benodigde capaciteit voor een gegeven vraag in één enkele gegeven ring. Een tweede stap bestond erin het volledige ring-planningsprobleem aan te pakken door mogelijke ring-topologieën te genereren en van het in de eerste stap ontwikkelde algoritme gebruik te maken om de "kost" ervan te bepalen en tot meer geschikte keuzes van ringen te komen. Voor een meer gedetailleerde beschrijving van de algoritmes verwijzen we met graagte naar [126,127].

De beschouwde indicatoren van de kost van het resulterende metro-netwerk zijn hieronder samengevat:

- ***Rx/Tx-capaciteit**:* Voor de Actieve Knoopstructuur (NoSR, SR) is dit het aantal geïnstalleerde baby-borden. Voor de Passieve architectuur (DBORN) telden we het totale aantal ontvangst- en zend-elementen en deelden we dit vervolgens door twee. De factor ½ zorgt voor een min of meer eerlijke vergelijking met de actieve knoop-architectuur, aangezien één baby-bord daar precies één ontvangst- (Rx) en één zend-element (Tx) bevat.
- ***Linkcapaciteit:*** Dit omvatte het tellen van het aantal golflengen gebruikt op elk ringsegment (i.e. elke link) en het sommeren ervan over het gehele metro-netwerk. Bemerk dat voor de Actieve Knoop dit steeds een viervoud zal zijn: capaciteit wordt er geïnstalleerd per golflengteband en we brengen alle vier de golflengten van zo'n band in rekening, ook al worden ze niet volledig benut.
- ***Aantal gebruikte golflengten:*** In tegenstelling tot de linkcapaciteit tellen we hier het aantal verschillende golflengten per vezelring (gesommeerd over alle ringen indien er meerdere zijn). Dit aantal is een indicatie voor de kost van de centrale Hub, vermits de grootte van de optische schakelmatrix erdoor bepaald zal worden. Ook hier tellen we voor de Actieve Knoopstructuur alle golflengten in een band mee.

De dimensionerings-algoritmes die we gebruiken zijn er in de eerste plaats op gericht de linkcapaciteit en het aantal gebruikte golflengten in het MAN te minimaliseren en slechts in tweede instantie de Rx/Tx-capaciteit. Onze resultaten gaven immers aan dat het precies die kostmaten waren die erg schommelden voor verschillende ring-configuraties en sterk beïnvloed werden door gepaste keuzes in de planning.

Het ligt voor de hand dat het antwoord op de vraag welke architectuur het efficiëntst is, en dus bijvoorbeeld het minst aantal golflengten zal nodig hebben, kan afhangen van het precieze verkeerspatroon waarvoor we een metro-netwerk dimensioneren. Om die invloed van de vraagmatrix te kunnen inschatten hebben we dan ook meerdere verkeerspatronen beschouwd, met name:

- ***Uniform**:* In dit geval zal elk element $D_{i,j}$ van de vraagmatrix gelijkgesteld worden aan dezelfde hoeveelheid bandbreedte d, waarbij we de kost van het netwerk zullen bekijken voor toenemende d-waarden. Verkeer dat door een bepaalde knoop op de ring wordt getransporteerd of ervoor bestemd is, zal hier gelijkmatig verdeeld zijn over alle andere knopen.
- ***Server:*** Een uitgesproken asymmetrische verdeling van het verkeer zal hier ondersteld worden: alle verkeer komt van, of is bestemd voor, één bepaalde knoop. In termen van de vraagmatrix betekent dit dat $D_{i,s}=D_{s,j}=d$ voor een zogenaamde server s, terwijl alle andere vragen $D_{i,j}=0$ (met $i,j \neq s$). Opnieuw zullen we d laten toenemen.
- ***Buren:*** Zoals de naam aangeeft zal een verkeersstroom in dit geval steeds tussen twee aangrenzende knopen vloeien. We beschouwen (i) het geval waar verkeer stroomt in de circulatie-richting van het licht in de ring, ($D_{i,i-1}=d$, overige $D_{i,j}=0$) en (ii) het geval waar verkeer van een knoop bestemd is voor zijn voorganger ($D_{i,i+1}=d$, overige $D_{i,j}=0$). Bemerk dat het onderscheid tussen (i) en (ii) enkel een

rol zal spelen wanneer de uni-directionele ringen gebruikt worden, maar niet wanneer in een ring twee tegengesteld roterende vezels voorzien zullen worden (zie verder in 2.2.3).

- ***3Buren:*** Hier zal een knoop enkel verkeer uitwisselen met drie buurknopen. De elementen van de vraagmatrix worden in dit geval als volgt gedefiniëerd: $D_{i,\,i+1}=D_{i,\,i+2}=D_{i,\,i+3}=d$, terwijl andere $D_{i,j}=0$ zijn.
- ***Random:*** We zullen elk element $D_{i,j}$ van de vraagmatrix willekeurig genereren, waarbij de bandbreedte voor een knopenpaar (i,j) genomen wordt op basis van een een uniforme distributie over het interval $[0,2d]$, met als gemiddelde waarde d.
- ***Reëel:*** Dit is een specifieke vraagmatrix die in [116] werd voorgesteld en bovendien behoorlijk realistisch bleek te zijn, in de zin dat hij zeer gelijklopend was aan een verkeerspatroon tussen ADSL concentratiepunten van de telecom-operator Telefonica in de omgeving van Madrid [116].

2.2.3 Uni- versus bidirectionele ringen

Een eerste vraag die we op basis van de dimensionerings-resultaten wilden beantwoorden was of het zin heeft twee tegengesteld roterende vezelringen te gebruiken. Dit wil zeggen dat lichtsignalen, dus gegevens, in twee richtingen doorheen de ring circuleren. Het is een techniek die zeer duidelijk nuttig is om de ring robuust te maken door gegevensoverdracht ook mogelijk te maken in geval van bijvoorbeeld een vezelbreuk [76,128,129]. Hier beogen we evenwel te achterhalen of het concept ook nuttig is, zelfs onder fout-vrije voorwaarden, om de hoeveelheid benodigde hardware te beperken.

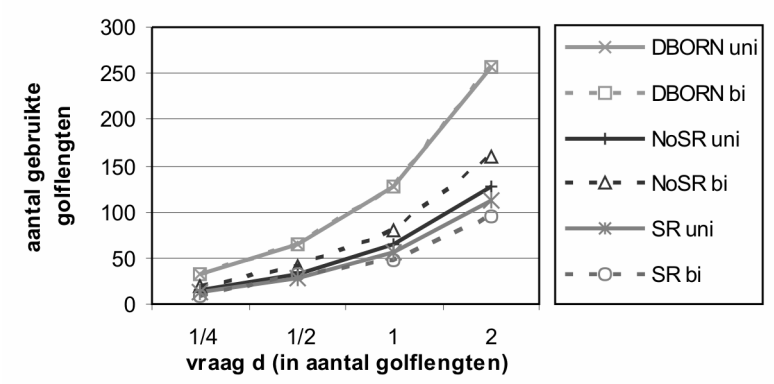
In de grafieken in Figuur 2.3 en Figuur 2.4 wordt een selectie getoond van onze resultaten voor het Uniforme resp. Random verkeerspatroon in een enkele ring. De curves in volle lijn geven de kost aan voor unidirectionele ringen, terwijl de stippellijnen de resultaten voor bidirectionele ringen aanduiden. Wanneer in het geval van bidirectionele ringen een Tx-element, of in geval van de Actieve Knoopstructuur een baby-bord, wordt geïnstalleerd, moet beslist worden op welke van de twee tegengesteld roterende ringen het geplaatst zal worden. In de gebruikte heuristiek [127] wordt eerst Rx/Tx-capaciteit toegevoegd voor vragen tussen knopen waarvoor het kortste pad in de rotatie-richting van de ring ligt, waarna de resterende vrije capaciteit op die ring zoveel mogelijk wordt ingevuld door andere vragen. De overige te voorziene bandbreedte wordt dan langs de andere ring geïnstalleerd.

Voor het DBORN-geval leren de grafieken dat het gebruiken van bidirectionele ringen geen winst oplevert in termen van de te installeren capaciteit. Zowel de Rx/Tx-capaciteit als het aantal golflengten kan niet gereduceerd worden door tegengesteld roterende ringen te voorzien. In tegendeel zelfs[2]: we stellen een toename van Rx/Tx-capaciteit vast en een lichte stijging van het aantal gebruikte golflengten.

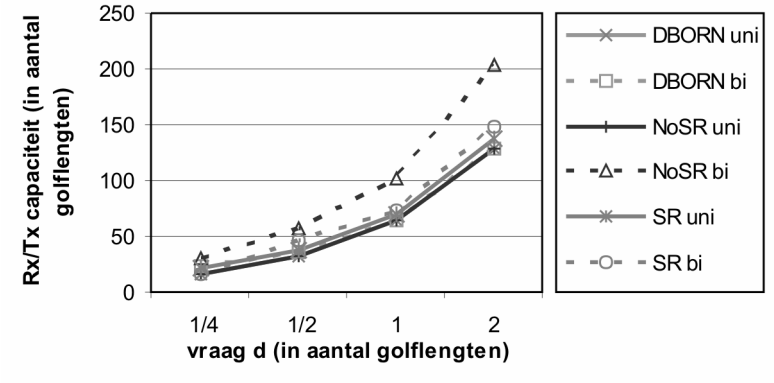
[2] Bemerk dat we voor het Uniforme vraagpatroon nauwelijks verschil waarnemen. Dit is evenwel te wijten aan het feit dat alle vragen precies even groot zijn en de bandbreedte van een golflengte steeds volledig kan worden ingevuld.

In geval van de Actieve Knoopstructuur in NoSR vinden we dat het installeren van bidirectionele ringen de Rx/Tx-capaciteit beduidend verhoogt en in iets minder uitgesproken mate ook het aantal gebruikte golflengten. Het enige voordeel dat bidirectionele ringen ons in dit geval opleveren, is dat de totale linkcapaciteit (i.e. het aantal gebruikte golflengten per link, gesommeerd over alle linken) licht gereduceerd kan worden, omdat een deel van het verkeer het kortste pad tussen bron en bestemming kan volgen. Het voordeel daarvan is evenwel beperkt, vermits alle verkeer doorheen de centrale Hub moet passeren. De spreekwoordelijke flessenhals wordt gevormd door de linken die verbonden zijn met de Hub. Dit is een effect waaraan niet afdoende verholpen kan worden door het verkeer in twee richtingen te laten stromen.

(a)



(b)



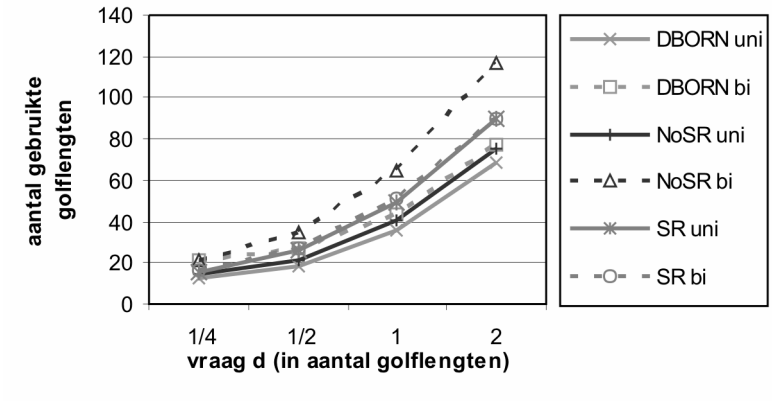
Figuur 2.3: Resultaten voor Uniform verkeer, enkele ring.

In het SR-geval bewijst het gebruik van bidirectionele ringen wél zijn voordeel. Dit is vooral het geval wanneer de fractie verkeer tussen naburige knopen binnen eenzelfde ring toeneemt (bv. het Buren patroon), eerder dan in min of meer symmetrische verdelingen van het verkeer (zoals Uniform of Random). Doordat ruimtelijk hergebruik toegelaten is, kan verkeer tussen bron en bestemming op

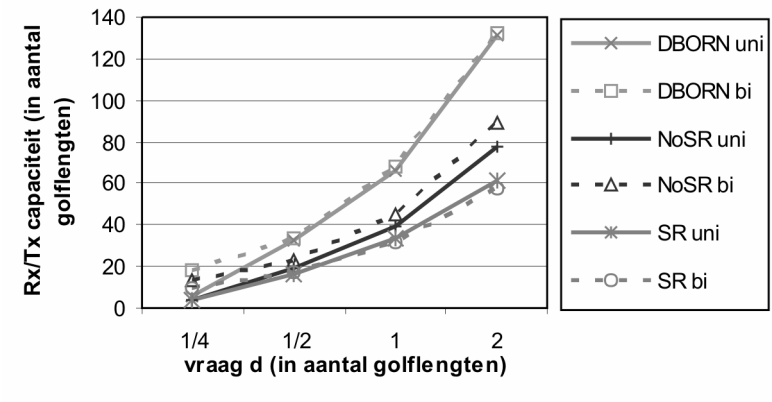
eenzelfde ring het kortste pad volgen. Dit laat toe de ongebruikte capaciteit ten volle te benutten.

We kunnen besluiten dat, vanuit een Rx/Tx-capaciteit of aantal gebruikte golflengten, het enkel bij ruimtelijk hergebruik (SR) voordeel kan opleveren bi-directionele ringen te installeren. En dan vooral wanneer er voldoende verkeer stroomt tussen knopen die tot dezelfde ring behoren.

(a)



(b)



Figuur 2.4: Resultaten voor Random verkeer, enkele ring.

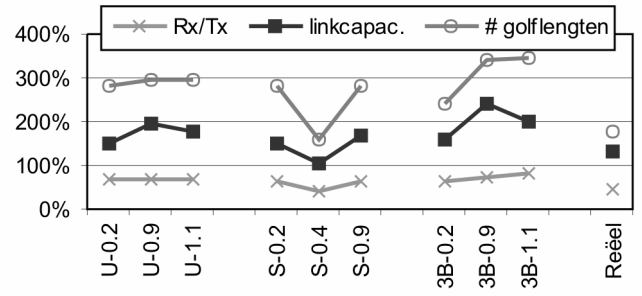
2.2.4 Ruimtelijk hergebruik

Het kenmerkendste verschil tussen de Actieve en Passieve Knoop-architectuur is dat de eerste een pakket daadwerkelijk uit de ring kan wegnemen en hierdoor ruimtelijk hergebruik toelaat. Het ligt dan ook voor de hand de vraag te stellen of dit grote voordelen oplevert. Om het antwoord daarop te achterhalen, dimensioneerden we ringen waarbij we voor de Actieve Knoopstructuur het concept van golflengte-banden acterwege lieten: we namen het aantal golflengten per band $B = 1$. De Passieve Knoopstructuur kent dit bandenconcept immers ook niet, zodat dit de

vergelijking in zekere zin eerlijker maakt. De invloed van het gebruik van het golflengtebandconcept wordt verder besproken in volgende Sectie 2.2.5.

De verkeerspatronen die we in deze studie beschouwden zijn Uniform (U), Server (S), 3Buren (3B) en Reëel. Om de vergelijking tussen Actieve en Passieve Knoopstructuren visueel te vereenvoudigen, hebben we er in Figuur 2.5 voor gekozen de kost voor de Passieve architectuur te delen door die voor de Actieve structuur (voor dezelfde vraag) en deze verhouding uit te zetten. Wanneer deze dus onder 100% ligt, betekent dit dat de Passieve structuur voordelig is, althans voor die betreffende kostmaat. Elk van de drie kostmaten (Rx/Tx-capaciteit, totale link-capaciteit en aantal gebruikte golflengten) is uitgezet. De labels op de X-as geven het betreffende vraagscenario aan waarbij in geval van een letter- en cijfer-combinatie de letter verwijst naar het vraagpatroon en het getal naar de waarde van *d* in de elementen van de vraagmatrix.

De resultaten leren ons dat de Passieve structuur enkel kan leiden tot een lagere Rx/Tx-capaciteit. De reden voor de hogere kost die geassociëerd is met de Actieve architectuur bestaat erin dat, om ruimtelijk hergebruik toe te laten, bron en bestemming dezelfde golflengte moeten gebruiken (cf. $B = 1$ impliceert een Tx van vaste golflengte), waardoor soms een extra Rx/Tx element voorzien moet worden. Het ruimtelijk hergebruik en dus de Actieve Knoopstructuur, bewijst enkel zijn nut wanneer de kost van het metro- netwerk gedomineerd wordt door de te installeren linkcapaciteit of het aantal golflengten per ring. Het is duidelijk dat dit voordeel volgt uit het niet spectraal scheiden[3] van golflengten voor zenden en ontvangen en het beter delen van die golflengtecapaciteit door ruimtelijk hergebruik.



Figuur 2.5: Kostverhouding Passieve/Actieve architectuur.

2.2.5 Golflengtebanden

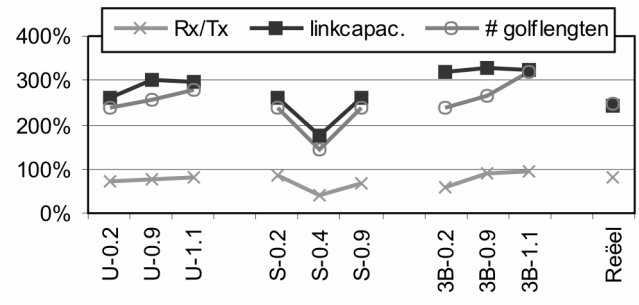
Zoals in voorgaande sectie aangegeven, is een tweede verschilpunt tussen de Actieve en Passieve architectuur dat in de eerste het gebruik van golflengtebanden wordt voorgesteld. In de vorige sectie hebben we voor de Actieve structuur

[3] Merk op dat het splitsen van het spectrum—in golflengten exclusief voor ofwel zenden, ofwel ontvangen (met goflengteconversie in één centrale knoop, Hub genaamd)—betekent dat er minimaal twee golflengtes moeten worden voorzien, terwijl een actieve architectuur genoeg heeft aan één golflengte in de ring.

golflengtebanden van één enkele golflengte bestudeerd (B=1). Hier beschouwen we de invloed van het gebruik van golflengtebanden, door het geval B=1 te vergelijken met het in [116] voorgestelde gebruik van B=4.

In Figuur 2.6 zetten we opnieuw een verhouding van de kostmaten uit tussen de twee scenario's in kwestie: we delen de kost voor gebruik van golflengtebanden (B=4) door die corresponderend met het geval zonder banden (B=1). De vraagscenario's zijn dezelfde als vorige sectie.

Uit die grafiek blijkt dat het voornaamste voordeel van het gebruik van golflengtebanden bestaat uit de reductie van de benodigde Rx/Tx-capaciteit. Maar wanneer het belangrijkste aandeel van de kapitaalkost voortkomt uit het voorzien van linkcapaciteit of het aantal golflengten gebruikt in de ring, is een golflengtebandconcept blijkbaar niet zinvol. Het aantal gebruikte golflengten ligt immers beduidend hoger dan het band-loze concept. Dit wijst erop dat het ruimtelijk hergebruik binnen een gegeven golflengteband beperkt blijft. De achterliggende reden hiervoor is dat in het voorgestelde bandenconcept er per knoop slechts één Rx/Tx element aanwezig kan zijn.



Figuur 2.6: Kostverhouding met/zonder gebruik van golflengtebanden (met: *B*=4; zonder: *B*=1 golflengte(n) per band).

2.3 Ruggengraatnetwerk

In dit onderdeel zullen we onze aandacht toespitsen op een knoop-architectuur voor het ruggengraatnetwerk. Het aantal in- en uitgangspoorten van een knooppunt in zo'n netwerk zal ook typisch hoger zijn dan in het geval van de zonet behandelde ring-gebaseerde metro-netwerken.

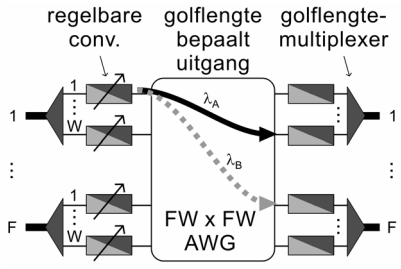
2.3.1 Alternatieven en motivatie

In het inleidende Hoofdstuk 1 gaven we al aan dat er twee architecturen zijn die in de literatuur overheersen als architectuur voor een OPS-schakelmatrix: AWG-gebaseerde enerzijds en SOA-gebaseerde "broadcast-and-select"-structuren anderzijds. De voornaamste motivatie voor de keuze van deze technologieën is dat alternatieven de voor OPS vereiste schakelsnelheden in de orde van nanoseconden niet kunnen halen. MEMS [46] hebben typische schakeltijden van de orde van een milliseconde of meer en de bubblejet-gebaseerde [87] en ook andere thermo-

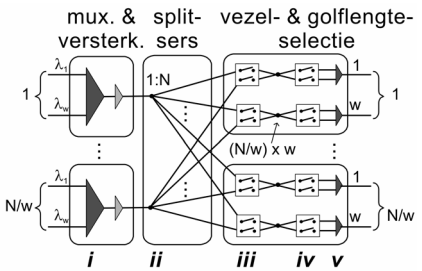
optische schakelmatrices [88–90] situeren zich in dezelfde of nog tragere snelheidsklasse (tot een tiental milliseconden). Bovendien zijn zowel de SOA-technologie [130] als de voor een AWG-aanpak obligate golflengte-convertoren [131] behoorlijk robuust, terwijl aan de stabiliteit en levensduur van bijvoorbeeld thermo-optische realisaties wel eens getwijfeld wordt.

Een typische schakelstructuur die gebruik maakt van een AWG, is geschetst in Figuur 2.7. Door de regelbare golflengte-convertoren aan de ingang in te stellen op de gepaste golflengte zal bepaald worden langs welke uitgangspoort van de AWG het licht de schakelmatrix verlaat.

Het principe van een "broadcast-and-select"-matrix, zoals die werd voorgesteld in [93], is geschetst in Figuur 2.8. De inkomende golflengten worden in een verzamel-stadium (i) via multiplexers in groepen van *w* samengenomen en (ii) via een splitser (na versterking via een EDFA) verdeeld over alle uitgaande poorten. Voor elk van die poorten wordt door middel van twee stadia SOA-schakelaars één enkel ingangssignaal overgehouden. Het eerste SOA-stadium (iii) houdt een signaal over dat alle golflengten bevat die in één van de de multiplexers werden gekoppeld. Het tweede SOA-stadium (iv) laat dit signaal slechts naar één ingangspoort van een AWG-component (v) propageren, zodat aan de uitgang effectief slechts één van de oorspronkelijke ingangssignalen wordt overgehouden. Om conflicten te vermijden, zal golflengte-conversie (als deel van regeneratoren, niet getoond in Figuur 2.8) ervoor zorgen dat elk van de signalen die in eenzelfde uitgaande vezel zal worden gekoppeld een verschillende golflengte heeft.



Figuur 2.7: AWG-gebaseerde schakelmatrix.



Figuur 2.8: SOA-gebaseerde "broadcast-and-select" matrix.

2.3.2 Schalen door gebruik van Clos-netwerken

Toen de inspirerende publicatie van Clos [132] verscheen was er nog lang geen sprake van optisch pakketschakelen, maar zijn aanpak—die erop gericht was telefooncentrales met erg veel in- en uitgaande koperparen met een minimaal aantal kruisschakelaars te realiseren—vindt toch toepassing in het OPS-domein. Wanneer we de grote aantallen golflengten die per vezel kunnen worden getransporteerd [44,45] en de gestage groei van de vraag naar bandbreedte in acht nemen, is het niet ondenkbaar dat er ook voor OPS relatief grote schakelmatrices nodig zullen zijn. Om de kost te drukken, of—zoals ook voor Clos het geval was—technologische

beperkingen te omzeilen, kan een schakelmatrix met een groot aantal poorten geconstrueerd worden door een interconnectie van kleinere schakelmatrices.

Vanzelfsprekend hebben ook andere onderzoekers dit idee om meerdere schakelstadia te hanteren in optische schakelaars bestudeerd. In [133] wordt een theoretische behandeling gegeven van schakelaars in drie domeinen: tijd, ruimte en golflengte. Er wordt besproken hoe deze gerealiseerd kunnen worden door een netwerk van schakelaars: ofwel door architecturen met componenten die in elk van de drie dimensies afzonderlijk schakelen, ofwel door een combinatie van tijd-ruimte schakelaars en ruimte-golflengte schakelaars.

Geïnspireerd op Clos behandelt MacDonald in [134] de uitbreidbaarheid van een optische kruisschakelaar (OXC). Terwijl in een typisch Clos-netwerk de grootte van de schakelmatrix in verschillende stadia verschilt, stelt MacDonald voor standaard bouwblokken te gebruiken (SKOL genoemd) en een modulaire architectuur te realiseren: indien nodig kan de capaciteit van een OXC uitgebreid worden door extra bouwblokken toe te voegen. Het nadeel van deze werkwijze is dat de maximale grootte van de OXC en de grootte van een bouwblok gerelateerd zijn, zodat in een initiële configuratie het bouwblok eventueel niet ten volle benut wordt.

In [135] wordt dan weer aandacht besteed aan de betrouwbaarheid van een MEMS circuitgeschakelde optische matrix. Net zoals bij Clos is een motivatie voor het invoeren van meerdere schakelstadia de reductie van het aantal componenten (hier MEMS spiegeltjes). Een bijkomend voordeel is hier dat zo'n schakelaar met meerdere trappen resulteert in een betere betrouwbaarheid: het aantal connecties dat verloren gaat voor een gegeven faalkans van een spiegeltje ligt beduidend lager. Eveneens toegespitst op het bepalen van de betrouwbaarheid van MEMS schakelknopen is [136]. Nog in een MEMS context worden in [137] interconnectie-patronen voorgesteld om het verlies van optische signaalsterkte zo laag mogelijk te houden.

Een ander resultaat in de sfeer van circuitschakelen is [138]. Daar wordt een architectuur voorgesteld die geoptimaliseerd is voor zogenaamd full-duplex verkeer, wat betekent dat voor elke verbinding van A naar B er ook één in de omgekeerde richting loopt. De schakelmatrices, die als bouwblok worden gebruikt in een schakelaar in meerdere stadia, worden er geacht licht in twee richtingen door te laten. Door dit aspect uit te buiten kan men in de resulterende architectuur het aantal componenten die nodig zijn voor een klassiek Clos-netwerk quasi halveren.

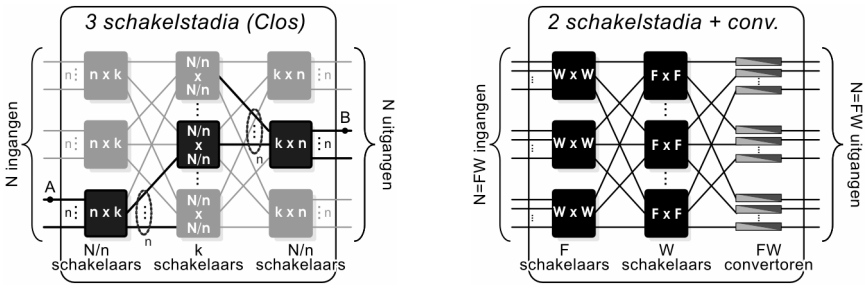
Het werk van [139] geeft aan hoe grote schakelmatrices voor golflengteschakelen met gebruik van WDM geconstrueerd kunnen worden. De voorgestelde architectuur voor golflengte-schakelende netwerken zonder golflengte-conversie maakt gebruik van drie types bouwblokken die allen gebaseerd zijn op FBG's (Eng. Fibre Bragg Grating) en door de fundamenteel verschillende werking niet te vergelijken zijn met hoger vermelde AWG- en B&S-architecturen. Om het aantal bouwblokken te minimaliseren wordt er een Clos-architectuur voorgesteld, waarbij de maximaal haalbare dimensie (rekening houdend met de fysische karakteristieken) wordt geschat en modulariteit wordt besproken.

Onze eigen studie die in de volgende subsectie wordt voorgesteld verschilt van bovenvermelde werken door zich specifiek toe te spitsen op de "broadcast-and-select"-architectuur voor optisch pakketschakelen. We tonen aan hoe het gebruik

van meerdere schakelende trappen technologische en/of fysische beperkingen kan omzeilen. We zullen hiervoor schakelmatrices in drie stadia gebruiken.

De generieke structuur van een Clos-netwerk met drie schakelstadia is geschetst in Figuur 2.9(a) voor een $N \times N$ schakelmatrix. De N in- en uitgangen zijn gegroepeerd in groepjes van n, waarbij voor elk van die groepen in het eerste en het laatste schakelstadium een schakelmatrix voorzien is. Het centrale schakelstadium omvat k bouwblokken die elk één verbinding hebben naar elk van de N/n blokken in het eerste en laatste stadium. De dimensies van de bouwblokken zijn dus $n \times k$, $N/n \times N/n$ en $k \times n$ in respectievelijk het eerste, het tweede en het laatste schakelstadium. De keuze van N zal de grootte van de uiteindelijke schakelaar bepalen, terwijl de keuze van n het totale aantal componenten zal beïnvloeden (zie verder). De waarde van k zal de blokkeringsgraad bepalen.

Zoals aangegeven in [132] kunnen verschillende vormen van blokkering worden onderscheiden. Een schakelmatrix wordt *strikt niet-blokkerend* (Eng. strictly non-blocking) genoemd wanneer het steeds mogelijk is een verbinding te maken tussen een nog vrije ingang en een nog vrije uitgang, onafhankelijk van en zonder te raken aan eerder gemaakte verbindingen. In *herschikbaar niet-blokkerende* matrices (Eng. rearrangeable non-blocking) kan een verbinding tussen een ingangs- en een uitgangspoort die beide nog vrij zijn ook steeds gerealiseerd worden, maar het is mogelijk dat dit het herconfigureren van eerder gemaakte verbindingen vereist. Wanneer niet kan gegarandeerd worden dat ten allen tijde een connectie gemaakt kan worden tussen vrije poorten, spreekt men van *intern blokkerende* architecturen (Eng. internally blocking).



(a) (b)

Figuur 2.9: Schakelmatrices in meerdere stadia: (a) een Clos-netwerk in drie stadia en (b) een structuur met twee schakeltrappen en één golflengte-conversie-trap.

De voorwaarde[4] opdat de structuur in Figuur 2.9a strikt niet-blokkerend zou zijn is $k \geq 2n-1$. Deze voorwaarde moet voldaan zijn wanneer de schakelaar asynchroon opereert en verbindingen worden opgezet en afgebroken op willekeurige ogenblikken ten opzichte van elkaar, zoals bij circuitschakelen of OBS. In die

[4] Veralgemeende voorwaarden voor Clos-netwerken in drie stadia waarbij de interne verbindingen hogere transmissiesnelheden hebben en/of er parallele kanalen zijn, worden beschreven in [142].

gevallen zal het inderdaad zo zijn dat de (levens)duur van connecties (i.e. circuits, of bursts in het geval van OBS) verschillen en het begin en einde ervan niet met elkaar samenvallen. Dit impliceert dat, eenmaal beslist is hoe de verbinding gerealiseerd wordt (in het bijzonder via welke schakelmatrix in het tweede Clos-stadium), we deze keuze moeten aanhouden voor de volledige duur ervan. Het is duidelijk dat een herschikbaar niet-blokkerende structuur niet voldoet.

Wanneer de matrix evenwel gebruikt wordt in een slot-gebaseerd regime volstaat een herschikbaar niet-blokkerende structuur wel [140]. In dit geval kan men in elk tijdslot voor alle op te zetten verbindingen gezamenlijk beslissen hoe ze worden gerealiseerd in het Clos-netwerk van schakelaars en dit onafhankelijk van voorafgaande of volgende tijdsloten. De voorwaarde voor herschikbaar blokkeren in de Clos-structuur van Figuur 2.9a is $k=n$. Het bewijs hiervoor is een welbekend resultaat uit de grafentheorie: het probleem van het vinden van een voldoend aantal schakelmatrices in het centrale stadium is equivalent met een graafkleurings-probleem in een bipartiete graaf van graad n en die is kleurbaar met n kleuren [141].

Wanneer het een schakelmatrix betreft voor OPS waarbij meerdere, zeg W, golflengten per vezel gebruikt worden (in WDM, zie Hoofdstuk 1), is het doorgaans toegelaten golflengte-conversie toe te passen. Dit betekent dat een inkomend pakket op een willekeurige golflengte de schakelaar mag verlaten, zo lang dit maar via de correcte uitgaande vezel gebeurt. In dit geval kunnen we het laatstse schakelstadium in de Clos-structuur elimineren en vervangen door golflengte-convertoren, met elk een vaste uitgaande golflengte. De voorwaarde is wel dat het aantal poorten n (waarbij een poort overeenstemt met één enkele golflengte), dat we in het eerste en laatste stadium bundelen, zo gekozen wordt dat ze alle n bij een zelfde vezel horen. De resulterende architectuur voor de keuze $n=W$ is geschetst in Figuur 2.9b.

2.3.3 Clos-gebaseerde B&S schakelmatrix

Het voornaamste pijnpunt van de in Figuur 2.8 geschetste “broadcast-and-select”-architectuur is de aanwezigheid van een splitser die een inkomend signaal verdeelt over elk van de uitgaande poorten. Het is duidelijk dat, omwille van de resulterende verzwakking van het lichtsignaal, we het aantal uitgaande poorten niet willekeurig hoog zullen kunnen kiezen. Wanneer we heel grote schakelmatrices willen construeren zullen we daarom Clos-gewijs verschillende kleinere bouwblokken met elkaar verbinden tot een groter geheel. Vanzelfsprekend zal tussen opeenvolgende stadia dan het signaal moeten geregenereerd worden (bv. met volledig optische 3R-regeneratoren [56]).

Een bijkomend voordeel is dat zo’n structuur met meerdere stadia ook de kost kan drukken ten opzichte van een monolithische architectuur (zelfs al mocht die desondanks splitsingsverliezen werkbaar zijn). In de veronderstelling dat de SOA-schakelcomponenten de kost zullen domineren, hebben we drie architecturen met elkaar vergeleken in functie van het aantal SOA-elementen dat ze vereisen: (i) de B&S-schakelaar in één stadium, zoals Figuur 2.8, (ii) een schakelaar in drie stadia, zoals Figuur 2.9a, (iii) een schakelaar met twee schakelstadia en een derde stadium bestaande uit golflengte-convertoren zoals in Figuur 2.9b.

Voor de B&S-architectuur met één schakelstadium wordt het aantal SOA-elementen gegeven door vergelijking 2.1. Vermits de architectuur aan in- en uit-

gangen voorzien is van elementen die golflengte-conversie realiseren, kan het aantal interne golflengten w onafhankelijk gekozen worden van het aantal golflengten W dat op een vezel wordt getransporteerd. Elementaire analyse leert ons dat het minimum aan SOA-elementen bekomen wordt voor de keuze $w=N^{1/2}$. Het resulterende aantal SOA-schakel-elementen wordt dan gegeven door formule 2.2.

$$S(N,w) = N \cdot (N/w + w) \qquad 2.1$$

$$S_{opt}(N) = 2 \cdot N \cdot N^{1/2} \qquad 2.2$$

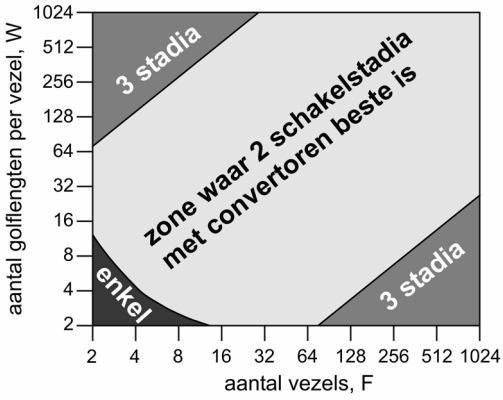
Het aantal SOA-schakel-elementen voor de Clos-structuur wordt gegeven door vergelijking 2.3. Opnieuw kunnen we dit aantal minimaliseren, aangezien door de aanwezigheid van golflengte-conversie de grootte van de groepjes n vrij gekozen kan worden. We vinden dat het minimum bereikt wordt voor de keuze $n=1/2 \cdot N^{1/2}$ en gegeven wordt door vergelijking 2.4.

$$C(N,n) = 2 \cdot N/n \cdot S_{opt}(N) + n \cdot S_{opt}(N/n) = 2 \cdot N \cdot (2 \cdot n^{1/2} + N^{1/2} \cdot n^{-1/2}) \qquad 2.3$$

$$C_{opt}(N) = 4 \cdot 2^{1/2} \cdot N \cdot N^{1/4} \qquad 2.4$$

In de laatste architectuur van een schakelmatrix voor F vezels met elk W golflengten, zullen we zoals aangegeven in Figuur 2.9b het laatste schakelstadium vervangen door golflengte-convertoren. Opnieuw kunnen we hiervoor het aantal SOA-schakel-elementen berekenen en vinden we vergelijking 2.5. Met de gemaakte keuze $n=F$ kan dit aantal niet verder geoptimaliseerd worden.

$$T(F,W) = F \cdot S_{opt}(W) + W \cdot S_{opt}(W) = 2 \cdot F \cdot W \cdot (W^{1/2} + F^{1/2}) \qquad 2.5$$

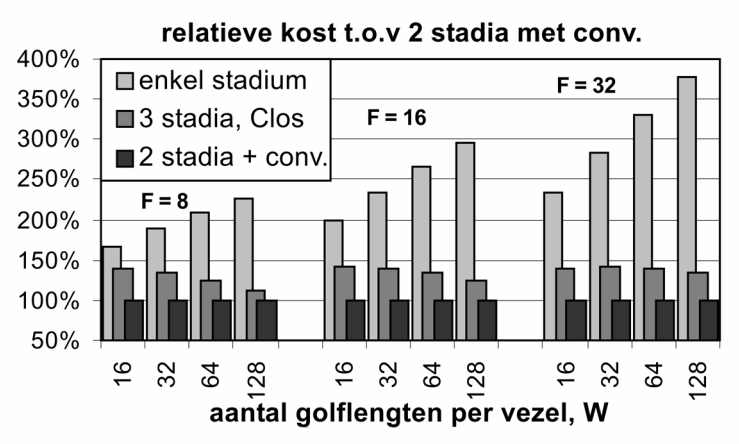


Figuur 2.10: Welke architectuur gebruikt het minst schakelende SOA's voor een gegeven grootte van de schakelmatrix?

Het antwoord op de vraag welke architectuur tot het laagste aantal SOA-schakel-elementen leidt voor gegeven dimensies van de te construeren schakelknoop kan met behulp van bovenstaande vergelijkingen gevonden worden. In Figuur 2.10 hebben we dit grafisch weergegeven voor een bereik van F en W gaande van 2 tot 1024 (merk op dat de dimensie van de knoop gegeven wordt door $F \cdot W \times F \cdot W$, dus $N=F \cdot W$ in vergelijkingen 2.1–2.4). De zone waartoe een punt met coördinaten (F,W)

behoort, is ingekleurd met een grijswaarde die aangeeft aan welke van de drie architecturen de meest voordelige is. Uit die figuur blijkt dat de B&S-structuur met een enkelvoudig schakelstadium slechts als beste uit deze vergelijking naar voor komt voor beperkte groottes van de schakelmatrix ($N=F\cdot W<12$). Voor wat de meest praktisch bruikbare waarden van W en F lijken te zijn, vinden we dat de aanpak met twee schakelstadia en golflengte-convertoren de meest voordelige is in termen van aantal gebruikte SOA-schakel-elementen. De Clos-configuratie met drie volwaardige schakelstadia blijkt enkel nuttig voor matrices waar het aantal vezels beperkt is en het aantal golflengten per vezel zeer hoog, of vice versa.

Om een idee te geven van hoe sterk de drie architecturen van elkaar verschillen in termen van vereist aantal schakelende SOA-elementen, hebben we die aantallen relatief ten opzichte van elkaar uitgezet in Figuur 2.11. Om de vergelijking te vereenvoudigen hebben we voor een ruime selectie waarden voor F en W het aantal benodigde SOA-schakel-elementen uitgezet nodig voor elk van de drie beschouwde architecturen, gedeeld door het aantal vereist voor de meertraps-architectuur met golflengte-conversie als laatste stadium. Hieruit blijkt dat vooral voor een monolithische B&S-architectuur het aantal schakelende SOA's beduidend hoger ligt. Het voordeel van de architectuur met convertoren ten opzichte van een volwaardige Clos-structuur met drie schakelstadia is beduidend kleiner, maar nog steeds belangrijk (enkele tientallen procenten).



Figuur 2.11: Relatieve kost in termen van aantal schakelende SOA's vergeleken met de architectuur met twee schakelstadia en golflengte-conversie als finaal stadium.

2.4 Besluiten

In de context van knoop-architecturen voor metro-netwerken hebben we onze aandacht gericht op OPS ring-architecturen, vermits deze als meest geschikte kandidaat beschouwd worden en als dusdanig in verschillende projecten en onderzoeksgroepen voorgesteld worden als MAN-architecturen van de (nabije?)

toekomst [77,78,79,118,123]. De meeste van die architecturen gebruiken een Actieve Knoopstructuur, waar een optisch pakket ofwel kan doorgelaten worden, ofwel van de ring gehaald kan worden. We hebben die architectuur vergeleken met een Passieve Knoopstructuur, die op netwerk-niveau leidt tot een logische bus-structuur [124]. In die vergelijking ligt precies de originaliteit van onze bijdrage tot het onderzoeksgebeid omtrent MAN-architecturen.

We hebben deze gerealiseerd aan de hand van het oplossen van een netwerk-planningsprobleem: hoeveel middelen zijn nodig om een gegeven hoeveelheid verkeer tussen een gegeven verzameling eindpunten (knopen) te dragen? Uit onze resultaten blijkt dat de Actieve Knoopstructuur enkel voordelen biedt wanneer het aantal golflengten per vezel een doorslaggevende kostfactor is: op vlak van zend- en ontvangst-capaciteit in de ringknopen biedt het gebruik van een eenvoudigere Passieve Knoopstructuur geen meerkost met zich mee. En passant hebben we ook vastgesteld dat het gebruik van bidirectionele ringen, waarbij licht in twee tegengestelde richtingen doorheen de metro-ring propageert, enkel tot lagere kosten kan leiden wanneer ruimtelijk hergebruik mogelijk is en er een voldoende hoeveelheid verkeer binnen een zelfde metro-ring blijft. Het golflengteband-concept zoals voorgesteld in [116]—waarbij in elke knoop afzonderlijk per band slechts de bandbreedte geassocieërd met één golflengte aangesproken kan worden—voor een Actieve Knoopstructuur bleek weinig zin te hebben wanneer het aantal gebruikte golflengten doorslaggevend is voor de kost.

Het spreekt voor zich dat de in dit hoofdstuk gepresenteerde studie van de MAN-ring-architecturen slechts één enkel facet is van een volledige evaluatie van de Actieve en Passieve Knoopstructuren. Complementaire studies omvatten de vraag hoe efficiënt de capaciteit in een gegeven ring kan benut worden onder variërende verkeerspatronen, door gebruik te maken van een geschikt MAC-protocol [71]. Vergelijking tussen de ring-architectuur en een PON-gebaseerde metro-architectuur [143] vormt een aanvulling van onze hoger behandelde studie. Uiteraard kan ook de vergelijking gemaakt worden met andere (bestaande, commerciële technologieën) waarmee een OPS-architectuur zal moeten concurreren [116]. Een studie voorgesteld in [118] toont dat de voorgestelde optische ring-architecturen ook praktisch realiseerbaar zijn, zelfs wanneer fysische beperkingen (op bv. maximale lengte van de ringen en het aantal knopen dat in serie kan geplaatst worden) in rekening gebracht worden.

In de context van meer geavanceerde knooparchitecturen, zoals die typisch worden voorgesteld voor vermaasde ruggengraatnetwerken [6,91,92], hebben we aandacht besteed aan de schaalbaarheid ervan. Het vernieuwende van ons werk bestond erin aan te tonen hoe voor een specifieke SOA-gebaseerde “broadcast-and-select”-architectuur een Clos-geïnspireerde structuur met meerdere schakelstadia gebruikt kan worden om grote schakelmatrices te construeren. We hebben daarbij een OPS-specifieke variant bedacht op de Clos-architectuur waarbij het derde stadium gereduceerd werd van een volwaardig schakelstadium tot loutere golflengte-conversie. Er bleek overigens dat zelfs voor bescheiden groottes deze variant nuttig is met het oog op het beperken van het aantal schakelende SOA-componenten. Zo’n Clos-achtige aanpak bewijst overigens ook z’n voordelen voor andere schakeltechnologieën: in [9] tonen we bijvoorbeeld hoe we, dankzij het

invoeren van meerdere schakeltrappen, met een gelimiteerd bereik van de golflengte-convertoren toch AWG-gebaseerde schakelmatrices met een groot aantal in- en uitgangspoorten kunnen construeren.

Merk op dat deze studies dienen aangevuld te worden met analyses betreffende de fysische haalbaarheid ervan (bijvoorbeeld de maximale splitsingsfactor voor B&S-architecturen, het al dan niet vereist zijn van 2R- of 3R-regeneratie-stadia tussen verschillende schakelstadia, etc.).

Hoofdstuk 3
Schakel-algoritmes voor OPS

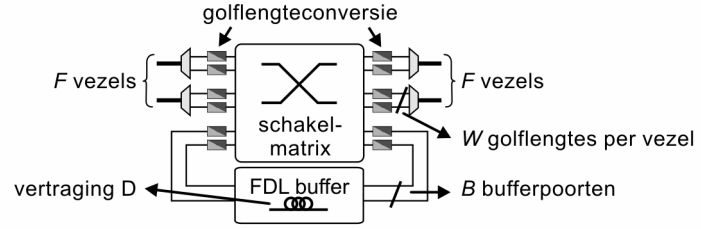
3.1 Situering

In het vorige hoofdstuk hebben we aandacht besteed aan enkele specifieke fysische knooparchitecturen voor optisch pakketschakelen. Dit hoofdstuk zullen we wijden aan de algoritmes die in een dergelijke knoop gebruikt kunnen worden om de schakelbeslissingen te nemen.

Aan de rand van het OPS-netwerk situeert zich de interface met elektronische apparatuur, zoals bv. IP-routers. Hier zullen de optische pakketten, of bursts in het geval van OBS, worden gevormd. Afhankelijk van het gebruikte pakketformaat, en in het bijzonder de lengte ervan (vast of variabel, etc.), zullen binnenkomende gegevens moeten worden geaggregeerd of eventueel gesplitst over meerdere optische pakketten. Hier treedt dus het klassieke probleem op van segmentatie en wedersamenstelling (Eng. Segmentation And Reassembly, SAR). Een voorbeeld van zo'n interface tussen elektronische IP-routers en een optisch ring-gebaseerd metronetwerk is te vinden in [144]. In het geval van optisch pakket- of burstschakelen gaat men er veelal van uit dat de optische pakketten of bursts (fragmenten van) meerdere IP-pakketten bevatten. Om de vullingsgraad bij pakketten van vaste lengte hoog te houden, gebruikt men doorgaans een timer: wanneer na verloop van tijd niet voldoende nieuwe gegevens zijn aangekomen om een optisch pakket volledig op te vullen, zal men de reeds aanwezige data toch versturen om de wachttijd te begrenzen. Analyse van dergelijke aggregatie-procedures zijn onder meer te vinden in [145,146,147]. Zulke tijdschakelaargebaseerde methoden hebben trouwens als bijkomend voordeel dat ze de zogenaamde zelfgelijkenis (Eng. self-similarity) reduceren [108,151,152,153], tenminste tot op bepaalde tijdschalen [154]. Zulke reductie van de zelfgelijkenis is erg welkom, gezien die het moeilijker maakt de performantie van het schakelen hoog te houden (i.e. verlieskans en vertraging te beperken) [148,149,150].

Om de optische pakketten het OPS-netwerk in te sturen, wat typisch gebeurt in een metro-omgeving, is een toegangsprotocol vereist. Wanneer het een optisch netwerk betreft zonder buffermogelijkheden, is zo'n MAC-protocol niet triviaal. Er moet immers voor gezorgd worden dat elk toegangspunt een eerlijke kans krijgt om verkeer in het netwerk te brengen. Dit gebeurt dikwijls door krediet-gebaseerde systemen, waarbij zogenaamde tokens worden toegekend [80–82]. Een alternatief,

of mogelijke aanvulling van een krediet-gebaseerd MAC, is met reservaties te werken en knopen expliciete toegang te laten vragen aan een centrale controle-knoop [155]. Anderzijds kan men op basis van metingen van de hoeveelheid verkeer de vereiste bandbreedte inschatten [156]. Waar verschillende metro-netwerken of delen ervan samenkomen, zoals in een Hub die meerdere ringen verbindt, moet ervoor gezorgd worden dat de interconnectie niet gemonopoliseerd wordt door een beperkt aantal subnetwerken. Het verzorgen van die gegevensoverdracht tussen de verschillende ringen vertaalt zich in een scheduling probleem [157]. Ook daarin kan men een reservatie-systeem gebruiken, of continue metingen van de hoeveelheid verkeer die tussen deelnetwerken wordt uitgewisseld [71].



Figuur 3.1: OPS-schakelaar met teruggekoppeld FDL-buffer.

In een ruggengraatnetwerk zal een optische pakketschakelende knoop veelal wél buffercapaciteit bezitten, aangezien het contentieprobleem in willekeurig vermaasde netwerken typisch moeilijk anders op te lossen valt (tenzij de belasting laag is, uiteraard). In dit hoofdstuk zullen we schakel-algoritmes ontwerpen en evalueren voor een veel voorgestelde knoop-architectuur (zie bv. [6,84,98,158]) met een teruggekoppelde buffer bestaande uit vezelvertragingslijnen (Eng. Fiber Delay Lines, FDL's). De structuur is geschetst in Figuur 3.1. De kern ervan wordt gevormd door een strikt niet-blokkerende matrix, bv. zoals die behandeld in Hoofdstuk 2. Hoewel de analyse in dit hoofdstuk niet afhankelijk is van de specifieke technologie waarin de matrix gerealiseerd wordt, is een cruciale onderstelde eigenschap van de schakelaar dat hij tot golflengte-conversie in staat is. We zullen die conversie immers gebruiken om contentie op te lossen. Een aantal uitgangspoorten van de matrix, aangeduid als B, zal worden verbonden met een bufferblok bestaande uit FDL's die opnieuw met de ingangspoorten geconnecteerd zijn. Merk op dat we de term poort steeds zullen gebruiken als verkorte vorm van golflengtepoort: een poort komt steeds overeen met één enkele golflengte en kan dus op elk tijdstip maximaal één optisch pakket dragen. De voor de verdere studies relevante parameters van de architectuur zijn op de figuur aangebracht.

Het vervolg van dit hoofdstuk is opgesplitst in twee delen: in Sectie 3.2 zullen we schakel-algoritmes voor vaste pakketlengtes behandelen, terwijl in Sectie 3.3 pakketten van variabele lengte aan bod komen.

Sectie 3.2 is gestoeld op onze publicaties [20, 24, 25], waarvan de eerste twee in Appendices D en E kunnen gelezen worden. De Appendix F [32] ligt aan de basis van Sectie 3.3.1, terwijl we ons voor 3.3.2 baseerden op [29] uit Appendix G.

3.2 Pakketten van vaste lengte

Vanzelfsprekend werd het probleem van het vinden van geschikte schakel-algoritmes voor het geval van een synchrone schakelaar voor pakketten van vaste lengte reeds uitvoerig bestudeerd in het kader van ATM [103]. Voorbeelden zijn [159,160], maar omdat ze betrekking hebben op gebruik van elektrisch RAM-geheugen zijn die resultaten niet van toepassing op het door ons bestudeerde geval van OPS met FDL-buffers. In het geval van een vezelvertragingslijn ligt het tijdstip waarop het pakket de buffer zal verlaten immers vast op het ogenblik dat het in de buffer geplaatst wordt. Bij RAM daarentegen kan een gebufferd pakket op een willekeurig gekozen ogenblik uit de buffer gehaald worden.

In de specifieke context van synchroon pakketschakelen met gedeeld recirculerend buffer is [161] wellicht het meest relevante werk uit de ATM-gerelateerde literatuur. Daarin wordt ook een teruggekoppeld buffer met een vaste vertraging beschouwd en een ondergrens op de pakketverlies-verhouding (Eng. Packet Loss Rate, PLR) berekend. Die analyse is evenwel onnauwkeurig voor praktische groottes van een OPS-schakelaar (cf. om een analytische behandeling mogelijk te maken onderstelde [161] een oneindig aantal schakelpoorten).

Beroemd vroeg werk in het kader van OPS vinden we in [97], maar de architectuur die er beschouwd werd betrof een voorwaarts schakelend buffer, zonder golflengte-multiplexering (WDM). In het kader van het Europese project KEOPS [92], toonde onderzoek van een niet-gedeeld uitgangsbuffer, maar wel met WDM, het voordeel aan van golflenge-conversie om contentie op te lossen [94]. Verder werk bewees dat ook met beperkte golflengte-conversie [162], of een beperkte verzameling van convertoren [163], verlies ten gevolge van contentie sterk gereduceerd kan worden.

De schakel-algoritmes in bovenvermelde OPS studies herleiden zich eenvoudig gesteld tot het bepalen van de kortste, in een FDL beschikbare, vertraging voor een pakket dat door contentie niet zonder wachten kan worden doorgeschakeld. Hoewel ze duidelijk gerelateerd zijn aan ons werk dat zo meteen behandeld wordt, beschouwen die studies geen teruggekoppelde buffer en evenmin een buffer die gedeeld wordt door verkeer bestemd voor alle uitgaande poorten. Recent werk dat wel zo'n recirculerende buffer beschouwde situeert zich onder meer in de context van fotonisch slotschakelen (Eng. Photonic Slot Routing, PSR) [164]. Maar dit PSR-concept verschilt sterk van het OPS-concept dat wij beschouwen: in PSR worden alle golflengtes steeds tesamen geschakeld. Pakketten op verschillende golflengten vormen er immers samen een multi-slot met één welbepaalde bestemming. Studies voorgesteld in onder meer [165,166] kaderen wel in het OPS-domein en beschouwen een gedeeld teruggekoppeld buffer, maar dan zonder WDM. Het gecombineerd gebruik van golflengte-conversie en vertraging door FDL's komt er dus niet aan bod. Bovendien wordt ook geen aandacht besteed aan het probleem dat zich stelt wanneer niet alle recirculerende bufferpoorten dezelfde vertraging kennen.

3.2.1 Een geschikt schakel-algoritme

In het geval van OPS met pakketten van vaste lengte gebruikt men een synchrone, slot-gebaseerde schakelaar (herinner u Figuur 1.4). Pakketten die via de inkomende vezels de schakelaar bereiken worden in de tijd gealigneerd aan de slotgrenzen. In elk tijdslot moet de controle-logica van de schakelaar beslissen wat er met de aankomende pakketten gebeurt. In het algemeen geval van schakelen in een WDM omgeving omvat die beslissing drie aspecten: (i) het bepalen van de uitgaande vezel waarlangs het pakket de schakelaar zal verlaten, (ii) het bepalen van de golflengte, (iii) het bepalen van de eventuele vertraging die het pakket moet krijgen.

Het bepalen van de uitgaande vezel gebeurt op basis van de hoofding van het pakket en behelst wat men in het Engels "forwarding" noemt. Dit kan bijvoorbeeld MPLS-gebaseerd zijn [28]. We zullen onderstellen dat de bestemming van een pakket eenduidig uit die hoofding volgt en we zullen geen deflectie toepassen. Dit betekent dat we niet toelaten dat een pakket naar een andere uitgaande vezel geschakeld zal worden om contentie op te lossen. De motivatie hiervoor is dat deze techniek enkel werkt wanneer het netwerk licht belast is [98].

Op welke golflengte een pakket doorgestuurd zal worden, kunnen we vrij kiezen, vermits golflengte-conversie in de onderstelde architectuur mogelijk is en die golflengte geen invloed zal hebben op de behandeling van het pakket in een volgende schakelaar: schakelbeslissingen worden enkel genomen op basis van de inhoud van de pakket-hoofding (in tegenstelling tot bv. bij golflengte-routering [167]). Welke golflengte precies gekozen wordt zal evenmin impact hebben op latere pakketten, want in het volgende tijdslot zal het pakket de schakelaar volledig verlaten hebben.

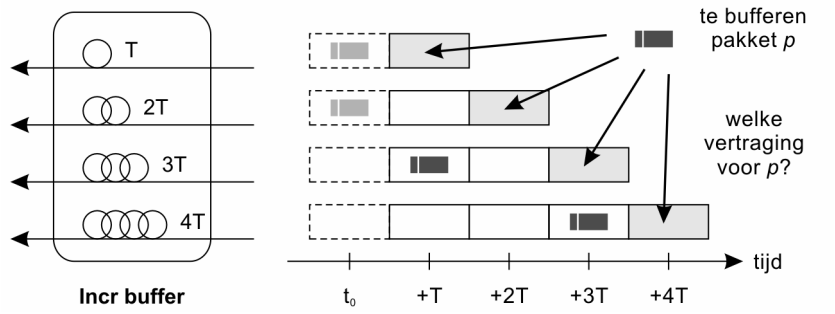
Het bepalen van welke vertraging aan een pakket gegeven wordt komt neer op de keuze van een recirculerende bufferpoort, en/of het aantal circulaties. Een beslissing over het aantal keren dat een pakket doorheen een FDL wordt gestuurd, zullen we niet a priori nemen bij aankomst van het pakket in de schakelaar. We zullen voor een aankomend pakket enkel kiezen of we het direct naar een uitgaande vezel, ofwel naar de FDL-buffer sturen. Wat er nadien mee gebeurt wordt beslist wanneer het pakket opnieuw aan de ingang van de schakelmatrix verschijnt. In [168] wordt dit gecatalogeerd als een PostRes strategie. Er blijft dus enkel nog de keuze over welke van de recirculerende bufferpoorten we gebruiken. Die keuze is vanzelfsprekend enkel van belang wanneer de bufferpoorten niet allemaal leiden naar een FDL van dezelfde lengte. We zullen daarom twee structuren beschouwen:

- ***Fix**:* In dit geval leidt elke teruggekoppelde bufferpoort tot een zelfde vertraging van één enkel tijdslot: voor elke poort $i = 1...B$ is de bijhorende vertraging $D_i = T$ (waarbij T de duur van het tijdslot is).
- ***Incr:*** Hier gebruiken we oplopende vertragingen voor elk van de B recirculerende bufferpoorten, i.e. $D_i = i \cdot T$.

Het schakel-algoritme dat we toepassen omvat twee selectie-fases: (i) selecteer voor elke van de F uitgaande vezels afzonderlijk maximaal W pakketten die rechtstreeks zullen worden doorgeschakeld, (ii) selecteer uit de overblijvende pakketten maximaal B pakketten die naar de recirculerende buffer gezonden worden. Pakketten die niet geselecteerd werden in één van deze fases zullen verloren gaan,

omdat golflengte-conversie en de aanwezige buffer niet volstaan om de contentie en overbelasting op te lossen. De selectie in elk van de fasen is willekeurig, maar voor de eerste fase geven we voorrang aan pakketten die al eens doorheen een FDL vertraagd werden tegenover pakketten die net de schakelaar bereiken via een ingangsvezel.

Wanneer de buffer FDL's van verschillende lengte bevat, zal in de tweede fase beslist moeten worden welke vertragingslijn voor een gegeven pakket gekozen wordt. Het gestelde vraagstuk wordt in Figuur 3.2 geïllustreerd voor een Incr-bufferstructuur. Gezien de schakelaar geen kennis heeft over welke pakketten in de toekomst via de ingangsvezels zullen aankomen, kunnen we niet a priori bepalen welke keuze van vertragingslijn tot de minste contentie zal leiden.



Figuur 3.2: Illustratie van keuze van vertraging wanneer niet alle FDL's dezelfde vertraging realiseren. De grijs gekleurde posities zijn mogelijke keuzes om het pakket *p* te bufferen. De pakketten aanwezig in de buffer voor t_0+T en t_0+3T zijn bestemd voor dezelfde uitgaande vezel als *p*.

De enige beschikbare kennis over de toekomst omvat eerder genomen buffer-beslissingen. Om te beoordelen of gebruik van die kennis meetbare voordelen (i.h.b. lagere PLR) oplevert hebben we volgende vier strategieën beschouwd:

- ***MinV (minimale vertraging)**:* Voor elk pakket dat we toelaten in de buffer kiezen we de nog vrije bufferpoort met de kleinste vertraging. Deze strategie zal dus de FDL poorten sequentieel opvullen indien deze volgens toenemende vezellengte geordend zijn. In het voorbeeld van Figuur 3.2 zal de vertragingslijn met lengte *T* gekozen worden.
- ***GeenO (geen overbelasting):*** Om een pakket *p* te bufferen kiezen we de nog vrije bufferpoort verbonden met de vertragingslijn van lengte *L*, waarvoor geldt dat het aantal pakketten dat de buffer zal verlaten op tijdstip t_0+L, bestemd voor dezelfde uitgaande vezel als *p*, niet meer bedraagt dan $W-1$ (waarbij t_0 het tijdstip is waarop de keuze voor *p* gemaakt wordt). Als er geen vrije bufferpoort gevonden kan worden die aan deze voorwaarde voldoet, gaat het pakket *p* verloren. Zodoende zal de *GeenO* strategie ervoor zorgen dat overbelasting van een uitgaande vezel nooit veroorzaakt wordt door gebufferde pakketten alleen. Wanneer er slechts één golflengte per uitgaande vezel zou zijn, zou voor Figuur 3.2 de keuze vallen op vertraging 2*T*.

- ***VermijdO (vermijd overbelasting):*** Zoek voor een pakket p een bufferpoort zoals in strategie *GeenO*. Wanneer dit niet lukt, kies dan de nog vrije bufferpoort met kleinste vertraging (zoals in *MinV*). Ook hier trachten we dus te vermijden dat overbelasting van een uitgaande vezel veroorzaakt wordt door alleen maar gebufferde pakketten, maar kiezen ervoor een pakket toch nog toe te laten in de buffer indien er een vrije bufferpoort is.
- ***Balans:*** We zullen pakketten bestemd voor dezelfde uitgaande vezel zo goed mogelijk spreiden in de tijd. Om een bufferpoort te kiezen voor een pakket p, tellen we voor elke beschikbare vertraging L het aantal pakketten dat de buffer op t_0+L zal verlaten voor dezelfde uitgaande vezel als waarvoor p bestemd is en noteren dit aantal N_L (met opnieuw de notatie t_0 voor het tijdstip waarop we de buffer-keuze maken). We sturen p dan naar de vrije bufferpoort die leidt naar de kortste vertragingslijn met laagste N_L . Voor Figuur 3.2 zou dit de poort met vertraging $2T$ zijn.

Wat betreft de volgorde waarin we de keuze maken voor pakketten toegelaten tot de buffer wordt opnieuw voorrang verleend aan pakketten die al eerder een doorgang door een FDL gemaakt hebben, maar verder is die volgorde willekeurig. Merk op dat de *GeenO* en *VermijdO* enkel verschillen van *MinV* wanneer het aantal recirculerende bufferpoorten het aantal golflengten per vezel overstijgt ($B > W$).

3.2.2 Methodologie

Om de performantie van de verschillende algoritmes voor verschillende waarden van de buffer-parameters te onderzoeken hebben we geopteerd voor simulatie. Analytische of numerieke methoden voor een teruggekoppeld buffer zijn immers beperkt tot relatief kleine schakelaars (bv. [160,161,165]), of geven enkel ondergrenzen [159,160]. Desalniettemin werd, zoals in [8] getoond, ons vertrouwen in de simulator wel gesterkt door vergelijking van simulatieresultaten met waarden voor de pakketverlies-verhouding bekomen met numerieke methoden, en dit voor verkeer beschreven door een zogenaamd MMAP-model [169].

Tabel 3.1: Gebruikte parameters van de pakketschakelaar voor de studies omtrent schakelen van pakketten van vaste lengte.

Symbool	*Waarde*	*Betekenis*
F	6	Aantal in- en uitgangsvezels
W	32	Aantal golflengten per vezel
B	0…64	Aantal golflengtepoorten van de schakelmatrix dat gebruikt wordt voor circulatie doorheen het FDL-buffer
D	1…64	Lengte van een FDL, uitgedrukt als de vertraging in aantal slots

Om de betrouwbaarheid van de resultaten te bewerkstelligen [170], hebben we gebruik gemaakt van de beroemde Mersenne Twister generator [171] voor de waarden van toevalsveranderlijken en een methode gebaseerd op "batch means" [172] om relatieve fouten kleiner dan 10% met 98% betrouwbaarheid te bekomen. De gemeenschappelijke parameters die we gebruikten staan in Tabel 3.1 samengevat en zijn gebaseerd op de 256×256 schakelmatrix zoals die voorzien was voor de demonstratie beschreven in [173]. Voor een meer gedetailleerde beschrijving van de simulator verwijzen we naar Appendix D.

Tabel 3.2: Gebruikte verkeersmodellen.

Model	*Distributie en parameters*	*Toelichting*
Poisson	$p(x) = \frac{e^{-\lambda} \cdot \lambda^x}{x!}, \quad x \in \aleph$	x is het aantal pakketten gegenereerd in een slot λ is de verkeers-intensiteit (in pakketen/slot)
GeoAanUit	$p(d) = \frac{1}{m+1} \cdot \left(1 - \frac{1}{m+1}\right)^{d-1}, \quad d \in \aleph \setminus \{0\}$ $m_{aan} = 4$ $m_{uit} = m_{aan} \cdot \left(\frac{1}{\lambda} - 1\right)$	d is de duur van de te genereren aan- of uit-periode in slots m is de gemiddelde waarde van die duur
ParetoAanUit	$p(d) = \frac{\alpha \cdot b^{\alpha}}{d^{\alpha+1}}, \quad d \in [b, +\infty)$ $\alpha_{aan} = 1.35$ $\alpha_{uit} = 1.5$ $b_{aan} = 1$ $b_{uit} = b_{aan} \cdot \frac{1 - 1/\alpha_{uit}}{1 - 1/\alpha_{aan}} \cdot \left(\frac{1}{\lambda} - 1\right)$	d is de duur van de te genereren aan- of uit-periode in slots b is de minimale duur van die periode α is de zogenaamde vormparameter: voor grotere waarden is de staart van de distributie nauwer, i.e. hogere d waarden hebben lagere probabiliteiten

Om de performantie voor voldoende uiteenlopende verkeersprofielen te vatten hebben we resultaten verzameld voor drie beschouwde modellen. Het eerste is het welbekende Poisson-proces, waarbij pakketten aankomen met exponentieel verdeelde tijden tussen opeenvolgende aankomsten. De twee andere modellen voor de aankomsten van pakketten gebruikten een zogenaamd aan/uit-model: aan-perioden, waarin in elk tijdslot een pakket aankomt, worden afgewisseld met uit-perioden, tijdens dewelke geen pakketten arriveren. Voor het GeoAanUit-model hebben we de geometrische distributie gebruikt voor de duur van zowel de aan- als uit-perioden en voor het ParetoAanUit-model de Pareto-verdeling. Dit laatste model werd gekozen omdat het welbekend is dat een aggregaat van zulke ParetoAanUit-

bronnen zelfgelijkend (Eng. self-similar) verkeer genereert [148]. De analytische gedaante van de gebruikte verdelingen en hun parameters, zijn samengevat in Tabel 3.2. Voor de ParetoAanUit-bronnen hebben we een minimale duur van de aan-periode van één enkel slot gekozen, terwijl de waarden van de vormparameter α gebaseerd zijn op metingen van reëel verkeer [149]. De gemiddelde duur van de aan-periode voor de GeoAanUit-modellen werd zo gekozen dat deze overeenkwam met die van het gebruikte ParetoAanUit-verkeer, zodat het verschil tussen deze modellen enkel ligt in de variantie van de duur van de aan- en uit-perioden.

De bestemmingen van de pakketten waren uniform verdeeld over alle uitgaande vezels. We hebben dit gerealiseerd door voor elke combinatie van een in- en uitgaande poort een afzonderlijke verkeersstroom te genereren voor een gegeven verkeersklasse (i.e. prioriteit, zie verder sectie 3.2.5). Voor het GeoAanUit-model hebben we zelfs 32 individuele bronnen voor elke (in,uit,prioriteit)-combinatie gebruikt, omdat het model enkel $\lambda<1$ kan genereren, wat met $W=32$ golflengten per vezel niet altijd volstaat om de gewenste belasting te bereiken. Onder de term belasting verstaan we in het vervolg de input-belasting, i.e. het gemiddeld aantal pakketten per tijdseenheid dat aan de schakelaar wordt aangeboden via zijn ingangen gedeeld door het maximale aantal dat er zou kunnen aankomen (i.e. $F \cdot W$ per tijdslot).

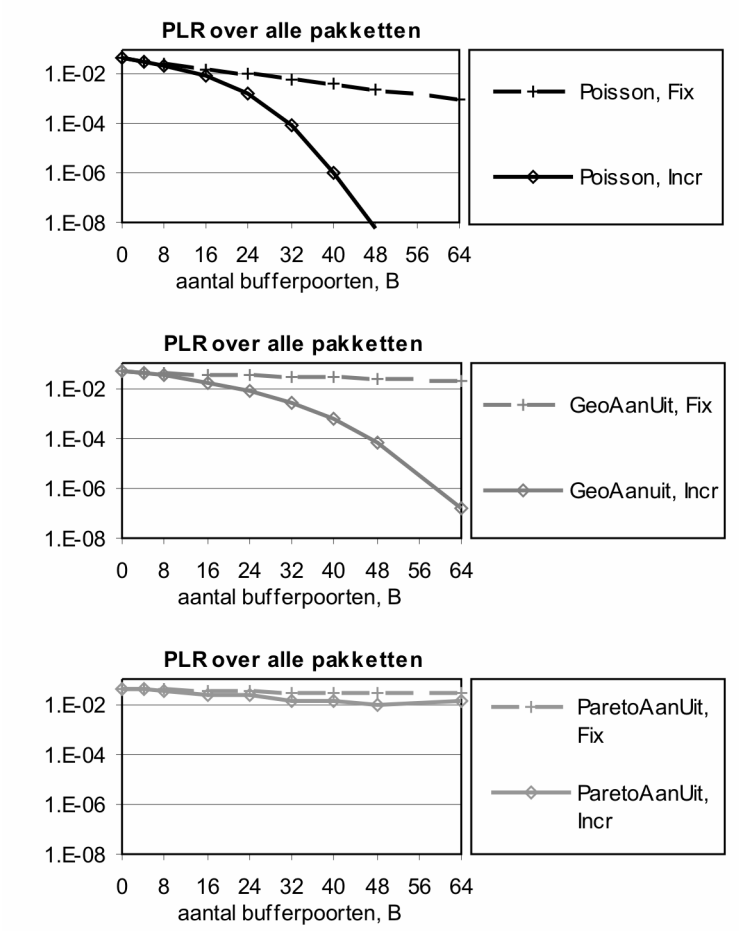
3.2.3 Keuze van de buffer-architectuur

Zoals hoger aangegeven kan voor de teruggekoppelde buffer geopteerd worden elk van de recirculerende poorten te verbinden met vertragingslijnen van verschillende lengte. Een eerste experiment dat we met onze simulator uitvoerden bestond dan ook uit het onderzoek naar het verschil in performantie tussen een schakelaar met een Fix-buffer, waarbij elk van de B bufferpoorten een vertraging van één enkel slot realiseert, en een Incr-buffer waarbij de bufferpoorten een toenemende vertraging veroorzaken (zie hoger).

In Figuur 3.3 tonen we de pakketverlies-verhouding (Eng. Packet Loss Rate, PLR) voor een toenemend aantal recirculerende bufferpoorten B. Zoals we intuïtief verwachten, zien we dat de Incr structuur beduidend lagere verliezen veroorzaakt voor toenemende B-waarden. Voor een hoger aantal bufferpoorten neemt immers het verschil in opslagcapaciteit tussen de Incr- en Fix-structuren toe: een Incr-buffer kan weliswaar maar evenveel pakketten opnemen in een bepaald tijdslot (namelijk B), maar het totale aantal pakketten dat potentieel aanwezig kan zijn in de buffer ligt hoger dan voor de Fix-buffer ($B \cdot (B+1)/2$ tegenover B voor Fix).

Wat de verschillen tussen de verkeerspatronen betreft, wordt hier het resultaat uit de klassieke wachtlijntheorie bevestigd: het toevoegen van bufferruimte levert meer op voor verkeer zonder correlatie in de tijd, zoals voor het Poisson-model. Maar ook wanneer er correlatie op relatief korte tijdschaal is, blijkt een FDL-gebaseerde buffer vrij efficiënt te zijn voor voldoende bufferdiepte (cf. het GeoAanUit-model met een Incr-buffer). Wanneer er correlatie op lange tijdschalen optreedt, zoals verkeer met zelfgelijkenis waar ParetoAanUit een voorbeeld van is, helpt het duidelijk niet een buffer te hebben dat slechts op relatief korte tijdschaal werkt. Dit resultaat was al bekend voor wachtlijnsystemen in de context van elektrische buffers (zie bv. [150] en referenties daarin).

Het verrast ons niet dat de prijs die betaald wordt voor de lagere verlieskans met een Incr-buffer erin bestaat dat de gemiddelde vertraging die pakketten ondergaan verhoogt. Desalniettemin zijn deze vertragingen behoorlijk klein[5] (enkele tientallen slots) en kunnen ze verwaarloosd worden tegenover vertraging die in regionale of toegangsnetwerken voorkomt. Een wellicht belangrijker nadeel van de Incr-structuur is dat de volgorde van de pakketten niet langer behouden blijft. Wanneer we voorrang verlenen aan pakketten die uit de buffer komen tegenover



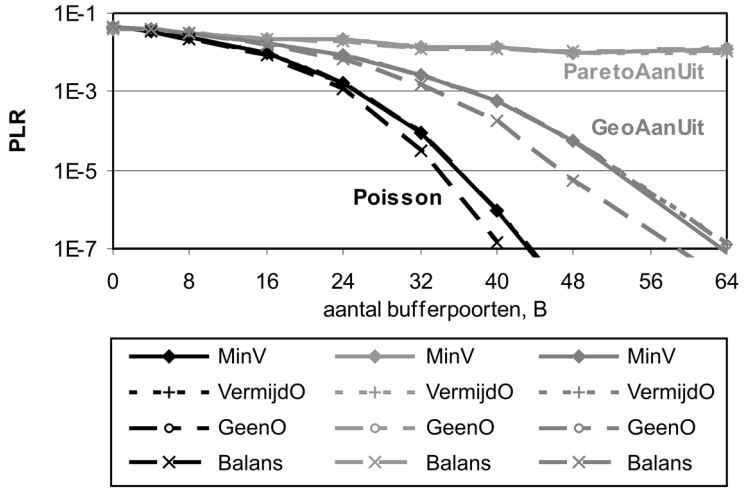
Figuur 3.3: Verlieskans (PLR) voor toenemend aantal bufferpoorten B: vergelijking *Incr* en *Fix-buffer*structuur. De belasting van de knoop was 0.95.

[5] In [6] bijvoorbeeld wordt een duur van een tijdslot voorgesteld van de orde van een microseconde, zodat een vertraging in de orde van duizend slots nog steeds slechts in het milliseconden-bereik ligt; vgl. met eind-tot-eind vertragingen in het Internet van vandaag in de buurt van een paar honderd milliseconden [174].

pakketten van een inkomende vezel, zal een schakelaar met Fix-buffer de volgorde van pakketten bewaren. In het geval van een Incr-buffer is dit niet langer waar: een pakket dat later aankomt kan een voorganger voorbijsteken en eerder aan de uitgang van de schakelaar (of zelfs de buffer) verschijnen. Voor een Incr-buffer met B=64 poorten blijkt bijvoorbeeld dat bij een belasting van 0.95 net iets minder dan 10% van alle pakketten uit volgorde[6] aankomt.

3.2.4 Bufferstrategieën

Om de hoger geïntroduceerde bufferstrategieën met elkaar te vergelijken hebben we het experiment uit de vorige sectie (waar we de MinV strategie hanteerden) herhaald voor de overige strategieën. De resultaten hiervan zijn te zien in Figuur 3.4. De schakel-algoritmes waarbij overbelasting vermeden wordt (GeenO, VermijdO) leiden tot quasi dezelfde verlieskansen als MinV, ook daar waar ze verschillende beslissingen kunnen nemen (voor $B > W$). De Balans-strategie bewijst wel z'n nut, tenminste toch voor die verkeerstypes waar buffercapaciteit het contentieprobleem kan verzachten (Poisson, GeoAanUit).



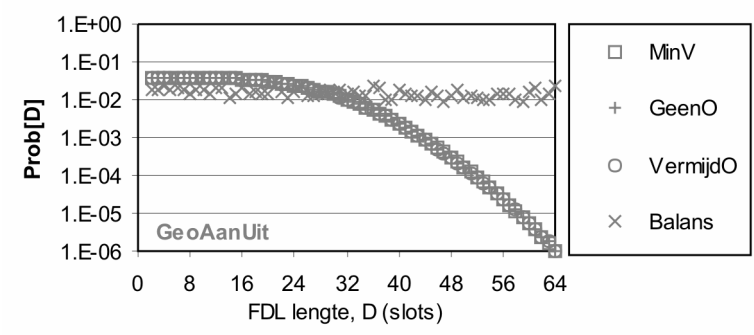
Figuur 3.4: Verlieskans (PLR) voor toenemend aantal bufferpoorten B: vergelijking van de bufferstrategieën. De belasting van de knoop was 0.95. Merk op dat de curves voor MinV, VermijdO, GeenO telkens zo goed als samenvallen.

De verklaring voor Balans' betere performantie in termen van PLR ligt precies in het feit dat pakketten over de tijd gespreid worden: daar waar de overige strategieën systematisch voor de kortst mogelijke vertragingslijn kiezen wanneer een

[6] Een pakket p wordt beschouwd als "uit volgorde" wanneer een ander pakket p', dat tussen dezelfde in- en uitgaande vezel geschakeld wordt, de schakelaar later binnenkomt maar toch vóór p verlaat.

pakket gebufferd wordt, zal Balans ze eerder verdelen over de beschikbare vertragingen. Het gevolg is dat minder pakketten gelijktijdig de buffer verlaten voor een zelfde uitgaande vezel en daardoor contentie met nieuw aankomende pakketten beperkt blijft. Dat pakketten effectief beter gespreid worden blijkt wanneer we op Figuur 3.5 kijken naar het aantal keer dat een bepaalde poort van de Incr-buffer gebruikt werd om een pakket te recirculeren. Aangezien het spreiden in de tijd impliceert dat lange vertragingslijnen vaker gebruikt worden, vinden we dat voor de Balans-strategie de gemiddelde vertraging typisch hoger ligt dan voor de overige buffer-algoritmes.

Tot slot willen we nog opmerken dat voor een Fix-buffer de hoger vermelde strategieën gereduceerd worden tot twee gevallen: MinV, VermijdO en Balans maken dan dezelfde bufferbeslissingen en enkel GeenO onderscheidt zich ervan wanneer $B > W$. Het gevolg van GeenO is dat een pakket nooit meer dan één keer doorheen de buffer circuleert. De spreekwoordelijke prijs die ervoor betaald wordt is een uiterst beperkte toename van de PLR. Voor een Fix-buffer volstaat het dus de meest eenvoudige strategie te volgen: zoveel pakketten tot de buffer toelaten als er bufferpoorten zijn.



Figuur 3.5: Verdeling van het aantal keer dat een bepaalde bufferpoort met een vertragingslijn van lengte D gebruikt wordt. Illustratie voor GeoAanUit in een knoop met *Incr-buffer* van B=64 poorten. De waarden voor MinV, GeenO en VermijdO vallen zo goed als samen.

3.2.5 Kwaliteitsdifferentiatie

Zoals al aangegeven in het inleidende Hoofdstuk 1 zal een OPS-netwerk het transport verzorgen van een heterogene verzameling verkeersstromen. Omdat elk van die types stromen verschillende eisen stelt aan het transportnetwerk in termen van bijvoorbeeld verlieskans en vertraging, moet het OPS-netwerk gediversifiëerde diensten kunnen aanbieden. Vermits het aanzienlijke aantal individuele stromen dat een typisch ruggengraatnetwerk doorkruist, denkt men doorgaans aan kwaliteits-differentiatie op basis van verkeerstypes, eerder dan aan het garanderen van

bepaalde verlieskansen e.d. voor individuele stromen[7]: veeleer een DiffServ-aanpak [175] dan een IntServ-gebaseerde [176] oplossing.

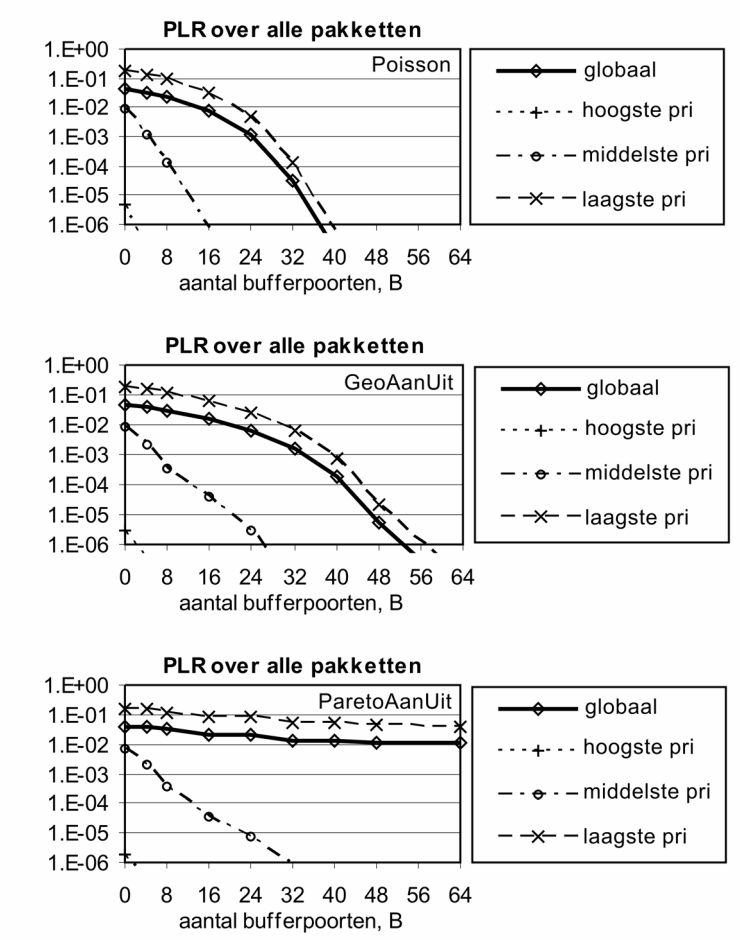
In die DiffServ-filosofie hebben wij onderzocht of de eerder voorgestelde schakel-algoritmes makkelijk uit te breiden zijn naar het aanbieden van gediversifiëerde diensten. De meest voor de hand liggende oplossing was het voorzien van een strikt prioriteitsschema. Elk pakket zal behoren tot een bepaalde prioriteitsklasse en we geven strikte voorrang aan pakketten van hogere prioriteit. Dit doen we door bij de selectie voor rechtstreekse doorschakeling of buffering eerst de pakketten van hogere prioriteiten te beschouwen en de lagere prioriteitsklasses pas toegang te geven tot een uitgaande vezel of een bufferpoort wanneer er nog plaats overblijft. De bufferstrategieën kunnen worden uitgebreid met een strikt prioriteitsschema door voor de buffering van een bepaald pakket p enkel rekening te houden met eerder gebufferde pakketten van dezelfde of hogere prioriteit.

De vraag of zo'n prioriteitsschema ook sterke diversifiëring in termen van verlieskans (PLR) oplevert wordt beantwoord door de resultaten in Figuur 3.6. Net zoals voorgaande resultaten wordt de verlieskans uitgezet voor toenemend aantal bufferpoorten, maar nu voor elk van de prioriteitsklassen. We beschouwden drie klassen: 50% verkeer van hoogste prioriteit, 25% verkeer van gemiddelde prioriteit en 25% van de laagste prioriteit. De PLR-curves tonen aan dat zelfs onder deze behoorlijk extreme voorwaarden (cf. groot aandeel hoogste prioriteit en totale belasting van 0.95) de diversifiëring uitgesproken is: zodra er enige bufferruimte is ($B > 0$) duikt de PLR voor de hoogste prioriteitsklasse onder 10^{-6}. Met uitzondering van het notoire ParetoAanUit-model kunnen we gelijkwaardige verlieskansen bereiken voor de laagste prioriteit door voldoende buffer te voorzien.

Uiteraard weerspiegelt het prioriteiten-mechanisme zich ook in de vertraging. Analyse van de simulatieresultaten leert dat enkel in de laagste prioriteitsklasse een aanzienlijk deel (30–40%) van de pakketten vertraagd wordt. Dankzij de aanwezigheid van golflengte-convertoren hoeft slechts een kleine fractie (0.5–1%) van de pakketten van gemiddelde prioriteit en een verwaarloosbaar deel (minder dan één op 10^6) van de hoogste prioriteit gebufferd worden. Vermits herordening enkel veroorzaakt kan worden door bufferen, duikt dit fenomeen—op de laagste prioriteitsklasse na—nauwelijks op.

Men zou kunnen vermoeden dat de succesvolle differentiatie ten koste gaat van de globale performantie en in het bijzonder leidt tot een verhoging van de globale verlieskans. Vergelijking van de PLR met die wanneer prioriteitsinformatie genegeerd wordt en alle klassen dezelfde behandeling ondergaan, leert evenwel dat dit niet het geval is. We hebben tussen simulatieresultaten van beide gevallen, waarbij het aankomstenpatroon van de pakketten identiek was, geen significante verhoging van de totale gemiddelde verlieskans vastgesteld.

[7] Voor een aanzet tot het bepalen van omstandigheden waarin garanties voor een aggregaat ook gelden voor individuele stromen verwijzen we naar [177].

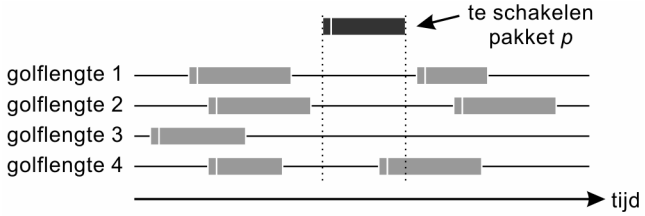


Figuur 3.6: Verlieskans (PLR) voor toenemend aantal bufferpoorten B: differentiatie door een strikt prioriteitsschema. De belasting van de knoop was 0.95, waarvan 50% hoogste, 25% gemiddelde en 25% laagste prioriteit.

3.3 Pakketten van variabele lengte

Het probleem van het schakelen van variabele lengte pakketten, ook in een optisch pakketgeschakelde context, is al grondig aan analyse onderworpen. Een specifieke moeilijkheid die ontspruit aan de niet langer vaste grootte van de schakeleenheden is het invullen van de zogenaamde leegtes die ontstaan tussen opeenvolgende pakketten (Eng. void filling, VF). Vermits er asynchroon geschakeld wordt en de pakketlengte niet constant is, is het beduidend moeilijker de bandbreedte volledig op te vullen (vergelijk met het fragmentatieprobleem bij geheugen-allocatie met

variabele groottes van geheugenblokken [178]). De leegtes worden gecreëerd doordat de tijd dat een pakket gebufferd wordt slechts gekozen kan worden uit een discreet aantal vertragingen die bepaald worden door de vezellengtes van FDL's. Daardoor is het niet altijd mogelijk een gebufferd pakket direct te laten aansluiten op een voorafgaand pakket op dezelfde uitgaande vezel en golflengte [179]. Het werk [180] beschrijft het "void filling"-algoritme voor asynchroon pakketschakelen en toont aan dat dit aanzienlijke voordelen oplevert tegenover algoritmes die de onstane leegtes niet invullen. Het algoritme is gelijkaardig aan eerder gerapporteerde resultaten zoals "horizon scheduling" [181], of LAUC-VF [109]: er wordt gezocht naar dat kanaal waar een pakket *p* zonder overlap met een ander kan worden geschakeld en de tijd tussen *p* zelf en diens voorganger zo klein mogelijk is. Zo zal in het voorbeeld geschetst in Figuur 3.7 gekozen worden voor golflengte 2. Andere strategieën om leemtes te vullen bleken minder efficiënt in het opkrikken van de performantie [182,183]. Het voornaamste nadeel van leemte-vullende algoritmes is dat ze het bijhouden van nogal wat toestandsinformatie vereisen (lijsten van leegtes per uitgaande golflengte). Voorbeelden van minder veeleisende, maar desalniettemin behoorlijk performante algoritmes zijn te vinden in [10], specifiek voor de architectuur die we in dit hoofdstuk beschouwen (Figuur 3.1). Wanneer de golflengte niet vrij gekozen mag worden, maar vast is voor een gegeven verkeersstroom (zoals in [179]) treedt het probleem van leemte-vullen alleen op als de aankomsten van een pakket niet altijd even lang op voorhand gekend zijn (zoals bij OBS met gedifferentiëerde offsets [184]).



Figuur 3.7: Het beslissingsprobleem voor pakketten met variabele lengte: op welke golflengte zullen we pakket *p* doorschakelen?

Algoritmes voor zo een knoop met een gedeelde buffer van teruggekoppelde FDL's worden in [168] geanalyseerd, in het bijzonder voor een OBS-concept. Er wordt een classificatie gemaakt van de strategieën in PostRes- en PreRes-algoritmes. In de laatstgenoemde zal, wanneer een pakket tot de buffer wordt toegelaten, meteen ook een uitgaande vezel en golflengte worden gereserveerd waarnaar het pakket geschakeld zal worden eenmaal het uit de FDL komt. Nadeel van zo'n PreRes-strategie is dat deze in bijvoorbeeld een OBS context kan interfereren met kwaliteitsdifferentiatie gebaseerd op verschillende offsets van de pakkethoofding.

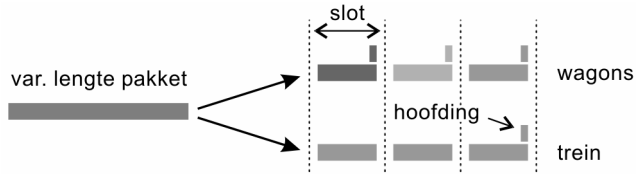
De voornoemde onderzoekswerken handelen allemaal over het schakelen van pakketten zonder deze in kleinere eenheden op te splitsen. In de slot-gebaseerde schakelaar zal men evenwel pakketten in slot-eenheden moeten verdelen, waardoor we de vraag kunnen stellen of het in sommige gevallen niet beter kan zijn deze slots

onafhankelijk te behandelen, eerder dan het pakket dat ze samen vormen als één schakeleenheid te beschouwen. Die vraag wordt in volgende sectie beantwoord.

3.3.1 Treinen versus wagons

Zoals aangeduid in Figuur 3.8 zijn er twee fundamenteel verschillende mogelijkheden om in een slot-gebaseerde, synchrone pakketschakelaar met pakketten van variabele lengte om te gaan. De eerste is de "trein van slots" [106] als één geheel te behandelen en een beslissing te nemen voor de volledige trein wanneer het eerste slot aankomt. In dit geval is het duidelijk dat enkel dit eerste slot een volwaardige hoofding moet bevatten met alle informatie nodig om te beslissen langs welke uitgaande poort de trein z'n weg doorheen het netwerk zal vervolgen. Een alternatief is elk van de onderscheiden wagons onafhankelijk van elkaar te schakelen. In dit geval dient elk van de wagonnetjes een eigen hoofding mee te krijgen en kan de pakketschakelaar opereren zonder zich van hun samenhoren bewust te zijn. De algoritmes uit vorige Sectie 3.2 kunnen dan zonder wijziging worden toegepast. Vanzelfsprekend dienen aan de uitgangen van het OPS-netwerk de pakketten wel opnieuw worden samengesteld (SAR).

Een vergelijking van de trein- versus wagonbehandeling werd al eerder uitgevoerd in [105] voor het specifieke geval van een gedeelde bus-structuur voor lokale of regionale netwerken van hoge snelheid. De auteurs analyseerden er de reductie van de belasting op de controle-eenheid in het geval van een treinconcept (cf. minder te behandelen hoofdingen) en bestudeerden het verloop van vertraging in functie van de hoeveelheid doorgeschakeld verkeer (Eng. throughput). Ze concludeerden dat vanuit het standpunt van benutting van de aanwezige bandbreedte de wagon-aanpak voordelig kan zijn wanneer de gemiddelde treinlengte beperkt is.



Figuur 3.8: Schakelen van pakketten van variabele lengte in slot-gebaseerd OPS: treinen of wagonnetjes?

In het kader van OPS werd al aandacht besteed aan de vergelijking tussen het slot-gebaseerd schakelen van pakketten van vaste lengte en een asynchroon schakelconcept met pakketten van variabele lengte [185,186]. Die studies vergeleken evenwel de performantie van een OPS-netwerk voor twee situaties met elk zowel een ander verkeersmodel als verschillende parameters voor de dimensies van de schakelknopen. Voor het slot-gebaseerde model werd gewerkt met aankomsten van pakketten van vaste lengte en men vergeleek de pakketverliezen met die voor een asynchroon verkeersmodel met pakketten van variabele lengte. In het laatste geval werden ook schakelknopen met meer bufferruimte gebruikt. Ondanks die uiteenlopende onderstellingen besluiten de auteurs van [185] dat slot-gebaseerd schakelen tot lagere pakketverliezen leidt. Ze tonen ook dat door extra golflengtes per vezel te

gebruiken voor een asynchrone schakelaar lagere pakketverlies-verhoudingen bereikt kunnen worden dan een slot-gebaseerde (met dus minder goflengten per vezel), of toch tenminste voor lichte belasting.

Hier richten we onze aandacht op een slot-gebaseerde OPS-schakelaar en we trachten uit te vissen of een wagon- dan wel trein-gebaseerd schakel-algoritme het best presteert in termen van logische performantie (i.e. pakketverlies, vertraging, kwaliteitsdifferentiatie). Hierbij zullen we—in tegenstelling tot [185,186]—voor beide gevallen hetzelfde verkeersmodel en pakketschakelaars met exact dezelfde dimensies beschouwen.

3.3.1.1 Methodologie

Om de trein- en wagonconcepten tegenover elkaar af te wegen, beschouwen we één enkele pakketschakelaar en zoals in de overige studies in dit hoofdstuk bepalen we de performantie door middel van simulatie. De beschouwde dimensies van de pakketschakelaar staan in Tabel 3.3 samengevat.

Het verkeersmodel dat we gebruikten is vanzelfsprekend een treinmodel, gelijkaardig aan [187]. Voor zowel de lengtes van de treinen als de tijd tussen twee opeenvolgende treinen gebruikten we een gediscretiseerde negatieve exponentiële verdeling, waarbij tijdsloten behorend tot eenzelfde trein steeds in opeenvolgende slots komen. Hoewel de exponentiële distributie niet noodzakelijk de meest realistische is wanneer treinen bv. als IP-pakketten geïnterpreteerd worden, geloven we toch dat de kwalitatieve conclusies die verder getrokken worden vrij algemeen geldend zijn.

Tabel 3.3: Gebruikte parameters in de trein vs wagons studie.

Symbool	*Waarde*	*Betekenis*
F	6	Aantal in- en uitgangsvezels
W	8	Aantal golflengten per vezel
B	0…64	Aantal golflengtepoorten van de schakelmatrix dat gebruikt wordt voor circulatie doorheen de FDL-buffer
D	$2 \cdot L$	Vertraging van de FDL-buffer
L	1.5…20	Gemiddelde treinlengte (eenheid=tijdslot)

Om de logische performantie te evalueren hebben we vanzelfsprekend vooral belang gehecht aan het bepalen van het pakketverlies (dat optreedt wanneer de voorziene buffering en golflengte-conversie falen in het oplossen van contentie). Aangezien we een trein als een ondeelbare gegevenseenheid beschouwen, rekenen we hem als volledig verloren zodra één enkele wagon de uitgaande vezel niet bereikt. Het is welbekend dat schakel-algoritmes niet altijd eerlijk zijn, in de zin dat ze pakketten discrimineren op basis van hun lengte: in het asynchrone pakket-

schakelen met leemte-vulling [180] maken korte pakketten meer kans om succesvol doorgeschakeld te worden dan lange [188]. Een ander belangrijk aspect is de mate waarin het algoritme kwaliteitsdifferentiatie kan realiseren. We zullen ook daaraan de nodige aandacht besteden. De belasting van de controle-eenheid die de schakelaar stuurt is een criterium dat we gemakkelijk kunnen evalueren: het aantal beslissingen dat moet worden genomen is gelijk aan het aantal hoofdingen die de schakelaar binnenkomen. Vermits dit slechts één hoofding per variabele lengte pakket betreft in de trein-aanpak tegenover één per slot voor het wagon-gebaseerd schakelen, zal het aantal operaties van het wagonconcept hoger liggen met een factor evenenredig aan de gemiddelde treinlengte in slots gemeten. Merk overigens op dat we ondersteld hebben dat de hoofdingen op een orthogonaal kanaal verstuurd worden (zoals geschetst in Figuur 3.8): het feit dat in een wagon-aanpak er meer hoofdingen verstuurd moeten worden heeft geen repercussies op de treinlengte gemeten in slots.

Het schakel-algoritme dat we gebruiken is in beide gevallen hetzelfde. Zowel wanneer we op basis van wagons als treinen beslissen, ondernemen we in elk tijdslot de volgende stappen: (i) selecteer voor elk van de F uitgaande vezels zoveel pakketten als er op de vezel in het beschouwde tijdslot t_0 vrije golflengtes zijn en schakel deze rechtstreeks door; (ii) selecteer uit de overblijvende pakketten die op t_0 aankomen er zoveel als er op dit ogenblik t_0 vrije poorten naar het recirculerend buffer zijn en schakel ze naar de corresponderende FDL's. Bemerk dat in het geval van een wagon-gebaseerde aanpak het aantal vrije uitgangen op een uitgaande vezel steeds W zal bedragen en het aantal vrije bufferpoorten B: de wagons uit het vorige tijdslot zijn dan al volledig doorgeschakeld. De controle-eenheid dient dus niks te onthouden over beslissingen die genomen werden in het verleden[8]. Dit is duidelijk wel het geval voor het in hun geheel schakelen van treinen: wanneer een trein wordt doorgeschakeld naar een bepaalde poort, dient men het ogenblik te onthouden waarop die poort terug vrijkomt. We kunnen dan ook verwachten dat, gezien de mindere complexiteit, de controle-eenheid in het geval van wagonschakelen sneller en/of met minder (dure) hardware de vereiste beslissingen kan nemen.

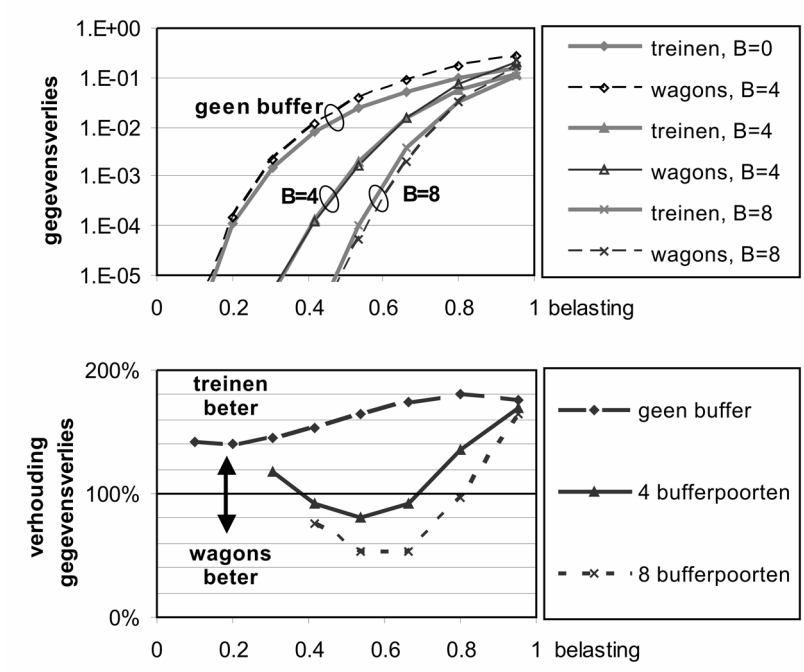
3.3.1.2 Invloed van de belasting

Met een eerste set van simulatieresultaten trachten we te achterhalen of er een bepaalde belasting bestaat waarvoor één van de schakelconcepten beter performeert. De studie [185] leerde immers dat een schakelconcept met variabele pakketten voor lage belastingen tot lagere pakketverliezen kan leiden. Aangezien de verlieskans duidelijk zal afhangen van de hoeveelheid buffer hebben we een staalkaart van drie buffergroottes beschouwd: geen buffer, vier, of acht bufferpoorten (B = 0, 4, 8). In Figuur 3.9a is voor toenemende ingangsbelasting de fractie verloren gegevens uitgezet (i.e. de gesommeerde lengte van de verloren treinen gedeeld door de

[8] We opteren immers opnieuw voor een PostRes strategie [168] waarbij pakketten die de buffer verlaten op dezelfde manier behandeld worden als nieuw aankomende: er wordt op het ogenblik van bufferen geen golflengte op een uitgaande vezel gereserveerd. Bovendien hebben in het onderzochte geval alle FDL's dezelfde lengte.

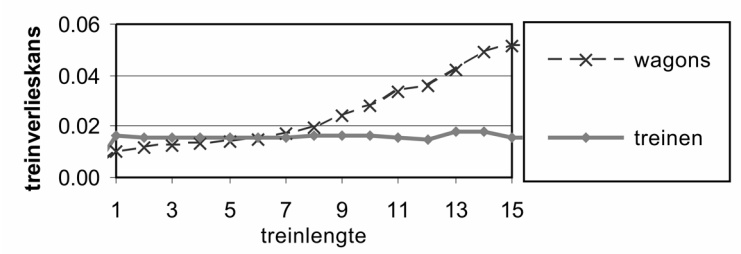
gecumuleerde lengte van treinen die bij de schakelaar aankwamen), die uiteraard een stijgend verloop kent.

Om de verschillen in gegevensverlies tussen trein- en wagonconcepten duidelijker te onderscheiden, hebben we in Figuur 3.9b de verhouding van het gegevensverlies gerealiseerd met de wagon-aanpak gedeeld door die in geval van het treinconcept. Wanneer er geen buffer is (B=0), zien we dat de wagon-aanpak nooit beter presteert: de vernoemde verhouding ligt hoger dan 100%. Wanneer er een recirculerende buffer voorzien wordt, stellen we evenwel vast dat wagon-gebaseerd schakelen wel voordeel oplevert. Maar dit is slechts waar voor een bepaald bereik van de belasting: voor erg hoge of erg lage belasting blijft het treinschakelen doeltreffender. De verklaring voor het slechter presteren van een wagon-aanpak luidt dat, wanneer er verschillende treinen in de tijd overlappen, het zo zal zijn dat van meerdere treinen een (beperkt) aantal slots verloren zal gaan, zodat het aantal treinen dat we als verloren beschouwen typisch hoger zal liggen dan in het treinschakelen. Voor een schakelaar met een buffer is dit evenwel niet altijd het geval, vermits een concept met vaste schakeleenheden die bufferruimte efficiënter kan benutten dan één met variabele. Wanneer de belasting niet te hoog is, zodat de contentie in grote mate kan worden opgelost, leidt dit tot lagere verliezen voor het wagonconcept. Voor lage belastingen is het zo dat er voldoende bufferruimte is zodat zelfs een treingeschakeld concept er genoeg aan heeft om de meeste contentieproblemen op te lossen: daar levert het efficiënter uitbaten van bufferruimte door de wagon-aanpak geen netto voordeel op.



Figuur 3.9: Wagons vs treinen: (a) gegevensverlies, (b) verhouding wagon/trein.

Zoals eerder aangegeven kan een schakel-algoritme voor pakketten van variabele lengte van oneerlijkheid blijk geven en langere pakketten discrimineren. Om te evalueren of dit voor de trein- en wagon-gebaseerde schakel-algoritmes ook zo is, hebben we in Figuur 3.10 de verlieskans uitgezet in functie van de lengte van de trein in geval van een buffer met B=4 poorten en een belasting van 0.62. Doordat de diepte van de buffer, i.e. de lengte van de FDL's, zo gekozen is dat vrijwel alle pakketten (ongeveer 95% om preciezer te zijn) in een FDL passen, zien we dat in het geval van een treinconcept geen noemenswaardige oneerlijkheid voorkomt. Voor een wagonschakelend algoritme vinden we evenwel een duidelijke discriminatie van de langere treinen, wat het evidente gevolg is van het feit dat de individuele wagons onafhankelijk van elkaar behandeld worden. Het schakel-algoritme geeft alle wagons die in een zelfde tijdslot aankomen immers dezelfde kans om door-geschakeld te worden: gegeven een bepaalde wagonverlieskans is het duidelijk dat de treinverlieskans min of meer evenredig zal zijn met de lengte gemeten in aantal slots of wagons.



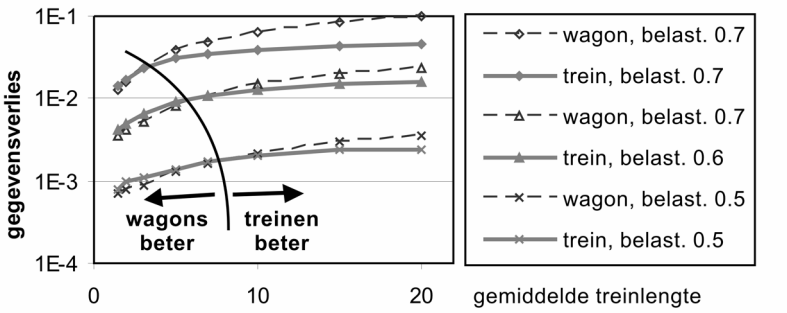
Figuur 3.10: Illustratie van discriminatie: verlieskans in functie van de treinlengte, voor een belasting van l=0.62 in een schakelaar met B=4 bufferpoorten.

3.3.1.3 Invloed van de granulariteit

Zoals een studie voor bus-structuren met een MAC-protocol aantoonde [105], kan het antwoord op de vraag of trein-gebaseerd dan wel wagon-gebaseerd schakelen te verkiezen is, afhangen van de gemiddelde lengte van de treinen. Anders geformuleerd komt dit erop neer dat voor een gegeven absolute lengte (gemeten in bits, of dus seconden voor een gegeven bandbreedte) het al dan niet beter zijn van een wagon-gebaseerde aanpak kan afhangen van de keuze van de lengte van een wagon, of met andere woorden van de (slot-)granulariteit. De keuze van een bepaalde slotlengte zal uiteraard bepalend zijn voor het in meer of mindere mate verspillen van bandbreedte door zogenaamde "padding" die ervoor zorgt dat een trein een geheel aantal slots bedraagt (men spreekt ook soms van "excess load" [179]). Maar zelfs wanneer we dit effect buiten beschouwing laten, zal ook de performantie in termen van gegevensverlieskans beïnvloed worden.

Om dit te onderzoeken hebben we in Figuur 3.11 het gegevensverlies uitgezet voor toenemende gemiddelde treinlengte. Hierin zien we het intuïtief verwachte resultaat dat een wagon-aanpak enkel voordeel kan opleveren wanneer de gemiddelde treinlengte voldoende klein is. Wanneer we naar de verliescurves kijken voor

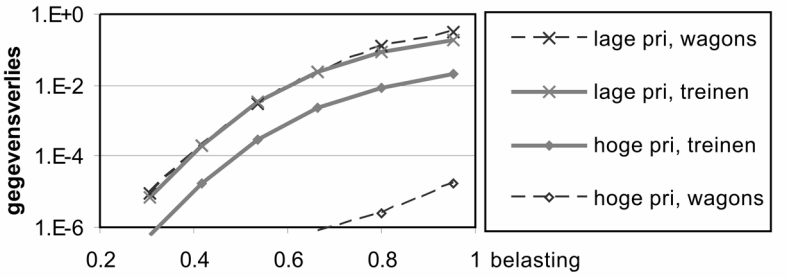
verschillende belastingen (we tonen ze voor ρ = 0.5, 0.6, 0.7), stellen we—niet algeheel tot onze verbazing—vast dat het keerpunt licht verschuift naar langere treinlengtes (dus fijnere granulariteit) met lagere belasting.



Figuur 3.11: Wagons vs treinen: gegevensverlies voor toenemende gemiddelde treinlengte in een schakelaar met B=4 bufferpoorten.

3.3.1.4 Kwaliteitsdifferentiatie voor treinen vs wagons

In onze studie omtrent het synchroon schakelen van pakketten met een vaste lengte is gebleken dat een eenvoudig prioriteitsmechanisme volstaat om uitgesproken differentiatie te realiseren in termen van verlieskans en vertraging. In deze sectie onderzoeken we of eenzelfde techniek ook voor het slot-gebaseerd schakelen van pakketten van variabele lengte voldoening kan schenken. Hiertoe passen we het schakel-algoritme zo aan dat bij de selectie voor doorschakelen naar een uitgaande vezel of naar de buffer onvoorwaardelijk voorrang gegeven wordt aan pakketten van hogere prioriteit. Hierbij zullen we eerder genomen beslissingen niet herroepen en niet aan zogenaamde preëmptie doen. Dit betekent dat in het geval van het treinschakelen we de trein steeds in zijn geheel laten: ofwel wordt hij volledig doorgeschakeld, ofwel gaat hij volledig verloren. Enkele wagons aan de staart kunnen niet weggelaten worden ten voordele van een later aangekomen trein van hogere prioriteit.



Figuur 3.12: Kwaliteitsdifferentiatie voor wagons vs treinen: gegevensverlies per prioriteitsklasse (40% hoogste prioriteit), in functie van toenemende belasting voor een schakelaar met B=4 buffer poorten.

In Figuur 3.12 hebben we de verlieskans uitgezet in functie van de belasting voor een schakelaar met B=4 bufferpoorten, waarvoor we het verkeer in twee prioriteitsklassen opdeelden, met 40% verkeer van de hoogste prioriteit. We stellen vast dat net zoals in het schakelen van pakketten van vaste lengte dit voor de hand liggend prioriteitsschema een sterke differentiatie oplevert. Ook in het treinconcept slagen we erin de verliezen voor de prioriteitsklassen duidelijk te scheiden, maar veel minder uitgesproken: waar de verlieskansen voor een wagon-aanpak tot zowat vier grootteordes verschillen, is de differentiatie in geval van treinschakelen beperkt tot één grootteorde of minder.

3.3.2 Kwaliteitsdifferentiatie voor asynchroon pakketschakelen

Uit voorgaande studie omtrent treinen en wagons blijkt dat een heel eenvoudig prioriteitsmechanisme slechts een beperkte graad van kwaliteitsdifferentiatie bereikt. In deze sectie zullen we enkele alternatieven voor differentiatie bestuderen voor het schakelen van pakketten van variabele lengte die asynchroon aankomen.

In de literatuur bestaat er een behoorlijk brede waaier aan mogelijkheden tot het verlenen van gedifferentiëerde diensten in een IP-over-WDM-scenario [189]. Reservatie is een welgekende manier om bepaalde kwaliteitsgaranties te kunnen realiseren: golflengte-convertoren [163] of buffers [190] kunnen exclusief voorbehouden worden voor bepaalde verkeersklasses, waarbij de hoogste prioriteit tot meer middelen toegang krijgt. Zulke reservaties voor bepaalde klassen kunnen zowel statisch vastliggen, als dynamisch aangepast worden [190]. We zullen ons in de hierna volgende studie evenwel beperken tot mechanismen zonder reservatie, waarbij het duidelijk is dat die mechanismen eenvoudig kunnen worden uitgebreid met een reservatiesysteem.

In een OBS-aanpak maakt men gebruik van gedifferentiëerde offsets: voor verkeer van hogere prioriteit worden de hoofdingen langer op voorhand uitgestuurd voor de eigenlijke pakket-inhoud [184]. Er is ook aangetoond dat zogenaamde segmentatie van de bursts helpt om de performantie (i.h.b. gegevensverlies) te verbeteren [191]. Hierbij is het toegelaten om langere bursts slechts gedeeltelijk door te sturen en om contentie te vermijden een portie aan de kop kan weglaten. De auteurs van [192] merkten op dat kwaliteitsdifferentiatie in zo'n geval gerealiseerd kan worden door gegevens van hogere prioriteit in de staart van een burst te positioneren, omdat die minder kans maakt verloren te gaan. Het is evenwel duidelijk dat dit een zeer impliciete manier is om differentiatie te bekomen. Bovendien is ze niet bruikbaar om differentiatie te verkrijgen tussen verkeersstromen met verschillende bron- en eindknopen. Dit verkeer zal immers nooit in dezelfde optische pakketten terechtkomen, zelfs al valt een deel van de gevolgde route samen. In onze studie zullen we segmentatie als middel tot gedifferentiëerde performantie dan ook niet beschouwen. We zullen zoals in de hoger uiteengezette trein-studie steeds pakketten in hun geheel schakelen of laten verloren gaan.

Een andere aanpak om op basis van prioriteitsindicaties verschillende dienstkwaliteit te bekomen is het met opzet laten verloren gaan van gegevens van lagere prioriteit—ook als er op dat ogenblik voldoende ruimte is om eventuele contentie op

te lossen—om op die manier meer middelen vrij te maken voor verkeer van hoge prioriteit. Deze filosofie wordt onder meer toegepast in de techniek van proportionele dienstkwaliteit (Eng. proportional QoS), zoals in [193]. Hoewel dit inderdaad lagere verliezen oplevert voor verkeer van hoge prioriteit is het zo dat de verlieskans, gemiddeld genomen over de totale hoeveelheid geschakeld verkeer, toeneemt. We zullen die techniek van opzettelijk verliezen van gegevens hier verder niet in aanmerking nemen. Het is evenwel evident dat, wanneer de hieronder bestudeerde algoritmes voor kwaliteitsdifferentiatie niet volstaan om voor verkeer van hoge prioriteit voldoende lage verliezen te bereiken, we deze algoritmes eenvoudigweg met intentioneel verliezen kunnen uitbreiden.

Samengevat zullen we in onze studie enkel methodes beschouwen zonder segmentatie, zonder reservaties en zonder opzettelijk gegevensverlies. Verder zal het zo zijn dat eenmaal een pakket op een uitgaande vezel werd toegelaten, deze beslissing niet zal herroepen worden. We vervolgen met een beschrijving van de bestudeerde alternatieven.

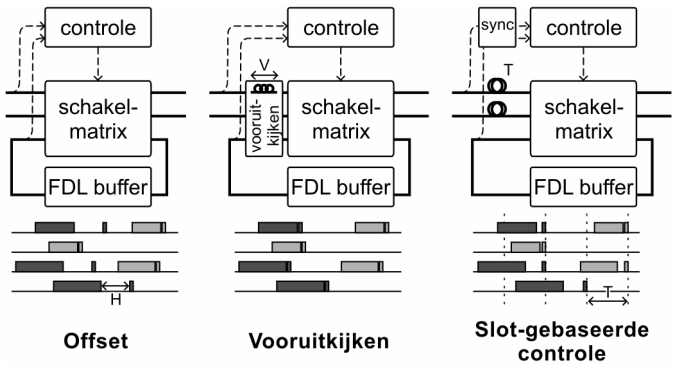
3.3.2.1 Drie alternatieven voor kwaliteitsdifferentiatie

Het schakel-algoritme dat we toepassen bestaat net als in vorige studies uit verschillende stappen en is opnieuw een PostRes-mechanisme dat voor gebufferde pakketten niet meteen een golflengte op de uiteindelijke uitgaande vezel reserveert: (i) gebruik LAUC-VF [109] om een vrije golflengte te vinden op een uitgaande vezel, (ii) als dit niet lukt, pas het LAUC-algoritme [109] toe om de meest geschikte recirculerende bufferpoort te vinden, (iii) laat het pakket verloren gaan wanneer ook dit mislukt. De drie onderzochte technieken voor differentiatie zijn:

- ***Offset differentiatie (offset):*** Dit is de hoger vermelde techniek die courant is voor zogenaamd burstschakelen (OBS-JET). Doordat hoofdingen van pakketten van hogere prioriteit eerder aankomen dan die van lagere prioriteit zal verkeer van hogere prioriteit meer kans hebben om te worden doorgeschakeld zonder verloren te gaan of zelfs gebufferd te worden.
- ***Vooruitkijken (vooruit):*** In dit geval wordt er geen onderscheid gemaakt in offset van de pakkethoofding. De schakelaar krijgt na het ontvangen van de hoofding evenwel een zekere tijd V om de uiteindelijke schakelbeslissing te nemen. Dit kan worden gerealiseerd door, zoals in Figuur 3.13 geschetst, een vertragingslijn te plaatsen tussen het afsplitsen van de hoofdinginformatie naar de controle-eenheid en de ingang van de schakelmatrix. Op die manier kan een pakket van lagere prioriteit verdrongen worden door een broertje van hoger belang dat tot V later aankomt. (Dezelfde filosofie van het inbouwen van een zekere bedenktijd om de schakelbeslissing te nemen wordt gevolgd in [27], maar wordt daar toegepast in een totaal andere context.)
- ***Slot-gebaseerde controle (slot):*** Het laatste alternatief neemt, opnieuw zonder onderscheid in hoofding-offset, de schakelbeslissingen in groep voor alle pakketten die tijdens het zelfde zogenaamde slot van duur T aankomen en wel op het einde van dat slot. Op die manier kan aan een pakket van hogere prioriteit inderdaad voorrang gegeven worden ten opzichte van tot een tijd T eerder aangekomen verkeer van lagere prioriteit. Om dit te bewerkstelligen volstaat een

gesynchroniseerd controle-kanaal dat de hoofdingen bevat, of minstens een (elektronische) synchronisatie-eenheid voor de schakelcontrole, zoals in de figuur geschetst.

Bemerk dat een studie voorgesteld in [194] voor één enkele bufferloze schakelaar dezelfde performantie zal bekomen als Vooruitkijken: daar heeft men geen vertragingslijnen om de bedenktijd in te bouwen (zoals in Figuur 3.13), maar onderstelt men dat voor alle pakketten, ook die van lage prioriteit, de hoofding voldoende ver vooruit wordt gestuurd zodat een genomen schakelbeslissing even later kan worden herzien. Merk ook op dat dit vereist dat men via het controlekanaal hoofdingen, die al werden doorgelaten voor lage prioriteitspakketten, kan annuleren ten voordele van later aankomend verkeer van grotere prioriteit.



Figuur 3.13: Drie alternatieven voor kwaliteitsdifferentiatie voor asynchroon geschakelde pakketten van variabele lengte.

3.3.2.2 Methodologie

Net zoals in vorige studies zullen we de voorgestelde alternatieven vergelijken door middel van simulatie. De structuur van de schakelknoop is ook hier die van Figuur 3.1, met als waarden voor de parameters: F=6 in- en uitgaande vezels met W=8 golflengtes elk en een variërend aantal bufferpoorten B=0…64 die allen dezelfde vertraging D realiseren.

Voor het verkeersmodel hebben we Poisson-aankomsten gebruikt met een links-getrunceerde exponentiële verdeling voor de pakketlengtes met gemiddelde lengte L en minimale lengte $L/2$. De bestemming van het aankomend verkeer was uniform gespreid over alle uitgaande vezels. We beschouwden twee prioriteits-klasses, met 40% verkeer van hoogste prioriteit, waarbij de klasse werd aangegeven in de hoofding van de onderscheiden pakketten. Die hoofding wordt al dan niet vooruit gestuurd op de eigenlijke pakket-inhoud, afhankelijk van het beschouwde mechanisme voor kwaliteitsdifferentiatie: de tijd tussen hoofding en inhoud is voor Offset-differentiatie afhankelijk van de prioriteit nul of H, bij Vooruitkijken steeds nul en gelegen in het half-open interval $[0,T)$ voor Slot-gebaseerde controle. We verwaarlozen de tijd nodig voor het verwerken van de hoofding.

Om het verschil in performantie tussen de onderscheiden QoS-mechanismen te evalueren gaan we opnieuw vooral aandacht besteden aan de verlieskans, maar we

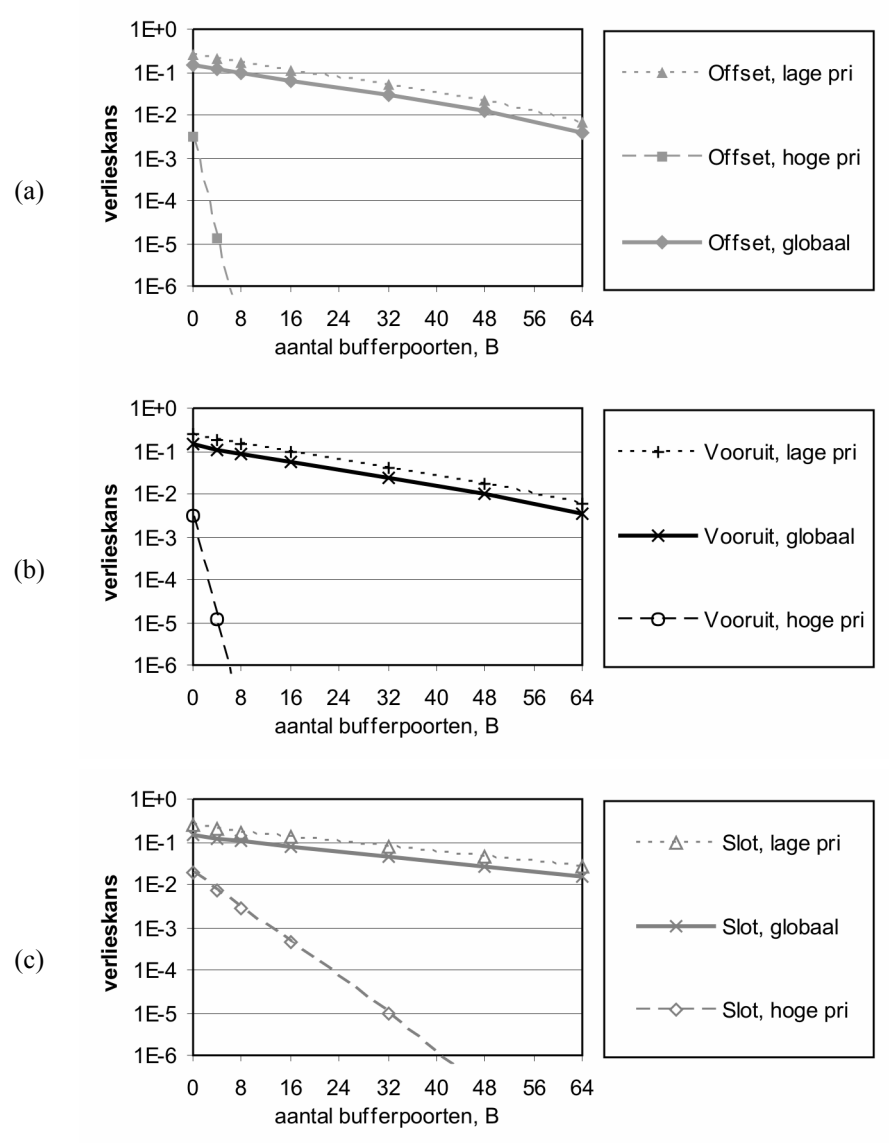
zullen ook de gemiddelde vertraging bekijken. Het doel van de mechanismen is uiteraard duidelijke verschillen in verlieskans per prioriteitsklasse te bereiken. Maar ook binnen een zelfde prioriteitsklassse kan de verlieskans variëren en is discriminatie op basis van pakketlengte niet ondenkbaar [188]. We zullen dan ook ruimte maken voor het kijken vanuit dit perspectief van eerlijkheid.

3.3.2.3 Invloed van het aantal bufferpoorten

Ons eerste experiment bestond erin voor een vaste belasting van 0.8 te kijken naar het verloop van de performantie-maten voor een toenemend aantal bufferpoorten. Het resultaat hiervan is geschetst in Figuur 3.14. De verlieskans is er uitgezet voor volgende parameters van de verschillende QoS-technieken: (i) in Offset was de offset van de hoogste prioriteitsklasse $H=2\cdot L$, (ii) bij Vooruitkijken stelden we de $V=2\cdot L$ en (iii) voor de slotresolutie bij Slot namen we $T=2\cdot L$. De diepte van de buffer, i.e. de gerealiseerde vertraging werd gelijk gesteld aan $D=2\cdot L$ (omdat vanaf die waarde de laagste verlieskans gerealiseerd kan worden vergeleken met minder grote bufferdieptes [168]).

Uit de grafiek in Figuur 3.14 blijkt duidelijk dat het gebruik van slot-gebaseerde controle in hogere verlieskansen resulteert dan Offset of Vooruit, vooral bij een toenemend aantal bufferpoorten B. Waar het verschil in globale verlieskans nog enigszins beperkt blijft, vinden we dat de verlieskans van verkeer van de hoogste prioriteit beduidend hoger ligt (verschillende grootteordes zodra er buffer-ruimte is) dan bij de Offset- en Vooruitkijkende technieken. Desalniettemin vinden we toch een duidelijke differentiatie en zelfs een verlieskans die aanvaardbaar kan zijn wanneer er voldoende bufferruimte is (cf. in de orde van 10^{-6} en lager voor meer dan $B=32$ bufferpoorten).

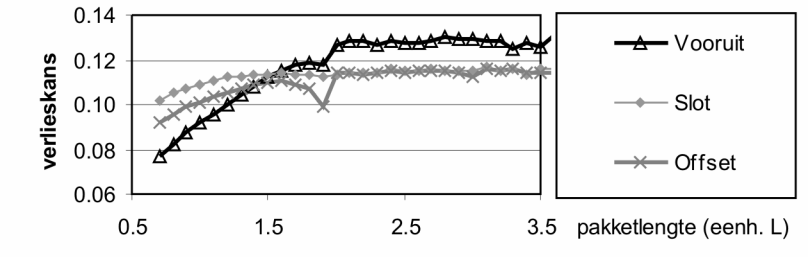
Vergelijken we Offset en Vooruitkijken, dan nemen we slechts een miniem verschil waar: op de logaritmische grafieken valt het relatief kleine voordeel van Vooruitkijken (tot ca. 20% lagere waarden voor de verlieskans) nauwelijks af te lezen. Wanneer we daarentegen de discriminatie van lange pakketten bekijken, stellen we wel een duidelijk onderscheid vast. In Figuur 3.15 hebben we de globale verlieskans in functie van de pakketlengte uitgezet. Deze grafiek bevestigt de verwachting dat korte pakketten in een asynchrone schakelaar bevoordeeld worden: ze ondervinden een lagere verlieskans. In geval van de Vooruitkijkende strategie is de discriminatie op basis van pakketlengte het meest uitgesproken: er is een bijna lineaire toename van de verlieskans tot pakketgroottes gelijk aan de tijd dat we vooruitkijken $V=2\cdot L$. Deze sterkere discriminatie is te wijten aan het preëmptieve karakter van het Vooruitkijken: er wordt een schakelbeslissing genomen op het ogenblik dat een pakket p aankomt. Voor later aankomende pakketten q van dezelfde of lagere prioriteit wordt die beslissing voor p gerespecteerd. Wanneer evenwel later een pakket van hogere prioriteit toekomt kan de schakelbeslissing voor pakket p aangepast worden, wat tot een andere configuratie van de vrije ruimtes op de uitgaande vezels kan leiden dan degene op basis waarvan beslissingen voor q genomen werden. Het gevolg is dat de allocatie van pakketten op golflengtes van uitgaande vezels niet optimaal is (in termen van de vrije ruimte die tussen pakketten overblijft). De langere pakketten zijn de eerste die daaronder lijden.



Figuur 3.14: Pakketverlies-verhouding voor een toenemend aantal recirculerende bufferpoorten *B*: (a) Offset-gebaseerde differentiatie, (b) Vooruitkijken, (c) Slot-gebaseerde controle.

In Appendix F [32] hebben we ook het gemiddeld aantal circulaties doorheen de buffer in kaart gebracht voor toenemende buffergroottes. Eén circulatie correspondeert uiteraard met een vertraging van $D=2 \cdot L$, de bufferdiepte. Vanzelfsprekend neemt de gemiddelde vertraging toe wanneer er meer bufferruimte

is: pakketten worden vertraagd in plaats van verloren te gaan. Net zoals voor de verlieskans is de rangorde volgens dalende performantie (hier: toenemende vertraging): (i) Vooruitkijken, (ii) Offset, (iii) Slot-gebaseerde controle, waarbij het onderscheid tussen (i) en (ii) heel beperkt is.



Figuur 3.15: Pakketverlies in functie van pakketlengte, voor een pakketschakelaar met F=6 in- en uitgaande vezels van W=8 golflengten elk en B=8 bufferpoorten, met een belasting van 0.8.

3.3.2.4 Invloed van de klasse-offset

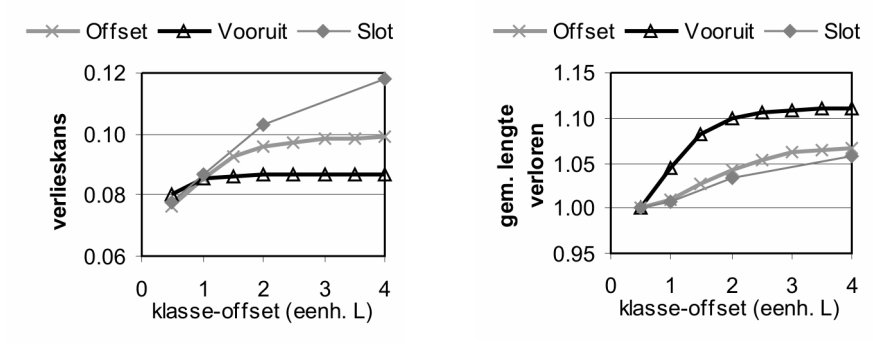
Vanuit het standpunt van kwaliteitsdifferentiatie is de mate waarin de performantie-maten verschillen voor elk van de prioriteitsklassen de belangrijkste graadmeter. De parameter waarmee die separatie beïnvloed kan worden is afhankelijk van het gebruikte differentiatie-mechanisme, maar we zullen die hier met een gemeen-schappelijke noemer "klasse-offset" aanduiden. Voor de Offset-gebaseerde methode gaat het dan over het verschil H in offset van de hoofding voor elk van de prioriteits-klassen, terwijl het bij Vooruitkijken de duur V van de vooruitkijk-periode betreft. In geval van slot-gebaseerde controle is het de duur T van het tijdslot.

Om de invloed van de klasse-offset te evalueren, hebben we een reeks simulaties uitgevoerd met toenemende waarden voor die klasse-offset. De resulterende globale gemiddelde verlieskans en vertraging zijn in Figuur 3.16a uitgezet voor een belasting van 0.8 van een schakelaar met B=8 bufferpoorten, elk met vertraging D=4·L. De klasse-offset variëerde tussen L/2 en 4·L.

Door de klasse-offset te verhogen vinden we een toenemende globale verlieskans: door pakketten van hoge prioriteit meer voorrang te geven zal hun verlies wel gereduceerd worden, maar dit gaat ten koste van pakketten van lage prioriteit, die de globale verlieskans domineren. Uiteraard is er een grens aan de mate waarin de performantie kan worden verbeterd: zodra pakketten van hoge prioriteit die van lagere prioriteit als het ware niet meer "zien", i.e. totale separatie is bereikt, verandert de verlieskans niet meer. Wanneer die totale separatie gerealiseerd wordt hangt af van de distributie van de pakketlengte. Voor de gebruikte exponentiële distributie is minder dan 5% van de pakketten langer dan 2·L. We zien dan ook een stagnerende verlieskans vanaf ongeveer die waarde. Bij Vooruitkijken treedt die stagnering het vroegst op, terwijl voor een Slot-gebaseerd schakelconcept er meer ruimte lijkt te zijn voor variaties in de klasse-offset.

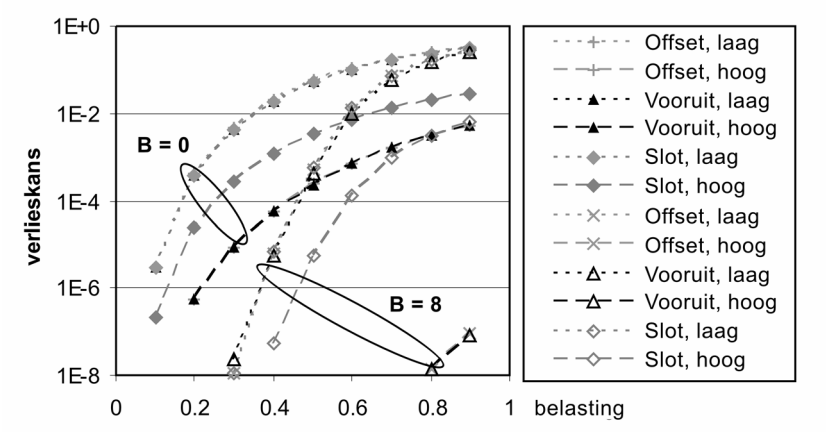
De invloed van de klasse-offset op discriminatie van langere pakketten blijkt uit Figuur 3.16b. Daar wordt voor dezelfde toenemende waarden van de klasse-offset

de gemiddelde grootte van de verloren gegane pakketten uitgezet. Met toenemende klasse-offset en dus klassenseparatie, verergert ook de discriminatie (voornamelijk pakketten langer dan gemiddeld gaan verloren). Zoals eerder vastgesteld zien we ook hier dat Vooruitkijken het sterkst discrimineert.



(a) (b)

Figuur 3.16: Invloed van de klasse-offset: (a) globale verlieskans en (b) gemiddelde grootte van de verloren gegane pakketten voor toenemende klasse-offset *H*=*A*=*T* in een schakelaar met parameters *F*=6, *W*=8, *B*=8, *D*=4*L*. De klasse-offset en pakketlengte zijn uitgezet in veelvouden van *L*.



Figuur 3.17: Pakketverlies-verhouding voor toenemende belasting voor een schakelaar met parameters *F*=6, *W*=8, *B*=0 of 8, *D*=2*L* en klasse-offset *H*=*A*=*T*=2*L*. Er was 40% verkeer van hoogste prioriteit. (Merk op dat curves voor laagste prioriteit voor gegeven *B* zo goed als samenvallen, alsook die van Vooruit en Offset voor de hoogste prioriteit.)

3.3.2.5 Invloed van de belasting

Het liefst zouden we differentiatie bewerkstelligen voor een zo groot mogelijk bereik aan belasting. Om na te gaan of de beschouwde technieken daar ook in slagen, hebben we in Figuur 3.17 de verlieskans uitgezet voor een belasting variërend tussen 0.1 en 0.9. Die curves kunnen bijvoorbeeld gebruikt worden om de "aanvaardbare belasting" te bepalen, met name: hoe zwaar mag een schakelaar belast worden om een gegeven verlieskans niet te overschrijden?

Wanneer we de resultaten voor hoge en lage prioriteitsklasses met elkaar vergelijken, observeren we een afnemend verschil in verlieskans met toenemende belasting voor zowel de Offset-gebaseerde differentiatie als het Vooruitkijken. Voor het bufferloze geval (B=0) bijvoorbeeld daalt de verhouding tussen de verlieskansen van lage en hoge prioriteit van een factor van circa 700 tot zowat 60. Voor de Slot-gebaseerde controle vinden we opnieuw een beduidend zwakkere differentiatie, maar ze neemt minder uitgesproken af met stijgende belasting. Ook blijkt de Slot-gebaseerde aanpak minder te kunnen profiteren van extra bufferruimte om daarmee de verlieskans van de hoogste prioriteit sterk te reduceren.

Als we de grafieken bekijken vanuit het standpunt van het bepalen van de aanvaardbare belasting, noteren we dat het voorzien van enige bufferruimte een aanzienlijk hogere last toelaat, zeker wat de hoogste prioriteiten betreft. Maar ook voor de lagere prioriteiten vinden we bijvoorbeeld dat voor een verlieskans van 10^{-3} de toelaatbare belasting verschuift van ca. 0.2 tot iets meer dan 0.5.

3.4 Besluiten

We hebben in dit hoofdstuk een brede waaier aan mogelijke schakel-algoritmes besproken voor een optische pakketschakelaar met een recirculerend buffer gedeeld door verkeer voor alle uitgaande vezels.

Voor het schakelen van pakketten van vaste lengte in een slot-gebaseerde schakelaar hebben we algoritmes bedacht en vergeleken voor het benutten van een buffer dat verschillende vertragingen kan bieden (het Incr-buffer). Het originele aspect van deze studie ligt in de nieuwe Balans-strategie, die zoveel mogelijk vermijdt dat pakketten die bestemd zijn voor dezelfde uitgang de buffer gelijktijdig verlaten en daardoor lagere verlieskansen realiseert dan de overige strategieën. De resultaten gekend voor (elektrische) schakelaars met conventioneel geheugen (i.e. RAM) werden terloops bevestigd: meer bufferruimte (i.e. meer bufferpoorten, of een Incr-buffer in plaats van de Fix-buffer) leidt tot lagere verliezen en voor verkeer met zelfgelijkenis (zoals het gebruikte ParetoAanUit-model) kan een beperkte bufferruimte weinig soelaas brengen.

Om gedifferentiëerde diensten te kunnen aanbieden, bleek in het geval van vaste pakketten in een synchrone schakelaar het invoeren van een eenvoudig prioriteitsmechanisme te volstaan om duidelijk lagere verlieskansen te realiseren voor verkeer van hogere prioriteit. En dit zonder te leiden tot een hogere globale verlieskans.

Wanneer dezelfde slot-gebaseerde schakelaar gebruikt wordt voor pakketten van variabele lengte, bleek een trein-gebaseerde aanpak—waarbij beslissingen voor

slots die één pakket vormen tesamen genomen worden, namelijk wanneer het eerste slot van zo'n zogenaamde trein aankomt—over het algemeen beter te presteren dan de wagon-aanpak waarbij elk slot afzonderlijk behandeld wordt. Enkel voor een bepaald bereik aan middelgrote belasting van de schakelaar kan de wagon-aanpak iets lagere verlieskansen realiseren, als de gemiddelde lengte van de treinen niet te lang wordt (een aantal slots). Wanneer evenwel differentiatie voor verschillende verkeersklassen gerealiseerd moet worden, blijkt dit eenvoudiger voor de wagon-aanpak: een eenvoudig prioriteitsmechanisme, dat strikte voorrang geeft aan pakketten van hogere prioriteit, volstaat. In de trein-aanpak blijkt zo'n eenvoudige prioritaire behandeling veel minder doeltreffend. Deze studie is, voor zover we in de literatuur hebben kunnen terugvinden, de eerste in zijn soort met betrekking tot een schakelaar voor een vermaasd OPS-netwerk.

Het realiseren van kwaliteitsdifferentiatie voor variabele pakketten werd verder onderzocht voor asynchroon schakelen waarbij dezelfde filosofie als in de trein-aanpak gevolgd werd: schakelbeslissingen worden voor het gehele pakket genomen dat ofwel in zijn volledigheid doorgeschakeld wordt, ofwel niet. Een eenvoudige slot-gebaseerde controle, waarbij schakelbeslissingen gelijktijdig genomen worden voor pakketten die in eenzelfde slot aankomen, blijkt tot vergelijkbare globale verlieskansen te leiden. Het is echter niet in staat dezelfde sterke differentiatie te bereiken als bv. het uit OBS bekende mechanisme van Offset-differentiatie. Een aanpak waarbij de offset niet verschilt voor verschillende pakketten is het Vooruitkijken: in termen van verlieskansen en vertraging is die strategie evenwaardig of zelfs een tikkeltje beter dan Offset-differentiatie. Het minpunt van die Vooruitkijk-strategie is wel dat langere pakketten sterker gediscrimineerd worden. Het is de verdienste van onze studie om de welbekende Offset-gebaseerde differentiatie te vergelijken met het Slot-gebaseerde en een nieuw Vooruitkijkend alternatief waarvoor we in de literatuur geen performantie-resultaten terugvonden.

We hebben van de verschillende technieken voor differentiatie ook de robuustheid onderzocht door de voornaamste parameters voor separatie van de prioriteitsklassen te variëren: (i) de "klasse-offset", (ii) belasting, (iii) het aantal bufferpoorten en (iv) de bufferdiepte. We kunnen besluiten dat (i) de klasse-offset nodig om voldoende separatie te bekomen afhangt van de distributie van de pakket-lengte, waarbij voor Vooruitkijken een iets kortere klasse-offset volstaat; (ii) de separatie licht afneemt voor toenemende belasting; (iii) extra bufferpoorten de verlieskans beduidend kunnen reduceren, maar de Slot-gebaseerde aanpak de bufferruimte minder efficiënt kan benutten dan de overige differentiatietechnieken; (iv) toenemende bufferdiepte helpt om verlieskans te reduceren tot op een zeker punt, gerelateerd aan de pakketlengte-distributie.

Hoofdstuk 4
Routering in een OPS-netwerk

4.1 Situering

Nu we de architectuur van een individuele schakelaar en de performantie ervan geanalyseerd hebben, lijkt het nuttig onze blik te verruimen en een netwerk van dergelijke schakelaars te beschouwen. Zo'n netwerk wordt geïnstalleerd om te voldoen aan een gegeven vraag naar bandbreedte. In deze context situeren zich twee gerelateerde deelproblemen: (i) hoeveel bandbreedte moet worden voorzien om aan een gegeven vraag te voldoen en (ii) hoe dienen de gegeven verkeersstromen doorheen dit netwerk gerouteerd te worden? Om deze problemen gezamenlijk op te lossen voor ring-gebaseerde metro-netwerken hebben we een heuristisch plannings-algoritme beschouwd en met behulp daarvan in Hoofdstuk 2 metro-architecturen met elkaar vergeleken. In dit hoofdstuk daarentegen zullen we enkel het routerings-probleem (ii) behandelen in vermaasde ruggengraatnetwerken opgebouwd uit OPS-schakelaars die we in het vorige hoofdstuk nader onderzochten.

Het bepalen van de routering in een vermaasd netwerk is een welbekend probleem. Vooral in het kader van golflengteschakelen werd het grondig onderzocht en herleidt het zich tot het toekennen van routes en golflengtes aan elk van de verkeersstromen (Eng. routing and wavelength assignment, RWA) [195]. Omdat het een NP-compleet probleem betreft [196], neemt men er veelal zijn toevlucht tot heuristieken [195,197–200].

In de pakketgeschakelde aanpak die wij hier beschouwen vervalt de golflengte-toekenning: bandbreedte aanwezig op verschillende golflengtes van een vezel kan gedeeld worden over alle connecties die deze vezel doorkruisen. In die optiek is routering in een OPS-netwerk nauwer verwant aan routering van IP-stromen. Routering in IP wordt typisch bepaald via gedistribueerde routeringsprotocols zoals OSPF [201] of IS-IS [202]. Die methoden gebruiken kortste-pad algoritmes om de routes tussen knopenparen te bepalen en IP routeringstabellen in te vullen. Door het gebruik van een kortste-pad algoritme worden doorgaans een beperkt aantal linken zwaar belast.

Om de verkeerslast gelijkmatiger te verdelen, heeft men zogenaamde last-verdeling (Eng. load balancing) bedacht. Een eerste voor de hand liggende manier bestaat erin de kostmaten aan te passen [203] en ze niet langer te laten bepalen door de fysische afstand. Meer flexibiliteit bekomen we door af te stappen van een

kortste-pad routering en de routes zorgvuldig te kiezen door uitgekiende methodes die men aanduidt als “traffic engineering”. MPLS-gebaseerde netwerken blijken zich hiertoe uitstekend te lenen [205,206]. Wanneer men daarenboven zoals in [204] een stroom tussen gegeven bron en bestemming gaat opsplitsen in deelstromen, slaagt men erin een quasi-optimale spreiding van de belasting over een gegeven netwerk te bereiken.

In de hier behandelde studie zullen we ook zo’n labelschakelend concept onderstellen, maar we splitsen gegeven vragen naar bandbreedte niet op in deelstromen. We trachten een algoritme te vinden dat de verkeersstromen zodanig verdeelt dat de verlieskans geminimaliseerd wordt. Hiertoe zullen we gebruik maken van een schatting van die verlieskans. Die schatting zal gebaseerd zijn op performantieresultaten van één enkele schakelaar, zoals bekomen met de simulator gebruikt in vorig hoofdstuk.

Merk op dat het beschouwde routeringsprobleem gesteld wordt in een netwerk met gegeven capaciteit. Wanneer het de bedoeling is zowel routering als het bepalen van de benodigde netwerkcapaciteit gekoppeld op te lossen, vervalt men in het geval van WDM-gebaseerde netwerken veelal in het zogenaamde “grooming”-probleem [207]. Voorbeelden van algoritmes voor zulke capaciteitsplanning zijn onder meer te vinden in [208], het daarop geïnspireerde [209] voor asymmetrisch[9] IP-verkeer, of [210] waar ook de vereiste bandbreedte voor robuustheid tegen mogelijke falingen in rekening gebracht wordt.

In dit hoofdstuk zullen we ons evenwel tot routering beperken en de netwerkcapaciteit als gegeven onderstellen. In de volgende Sectie 4.2 zullen we de gebruikte methodiek, inclusief twee bestudeerde algoritmische varianten, beschrijven. De voorgestelde algoritmen worden geëvalueerd in 4.3. Vervolgens zullen we in Sectie 4.4 het nut aantonen van het schatten van het pakketverlies om op basis daarvan de routering te bepalen. Besluiten staan samengevat in Sectie 4.5.

Eigen publicaties die in dit hoofdstuk verwerkt zijn: [7] uit Appendix H en [21,22] voor de gebruikte heuristische oplossingmethoden.

4.2 Methodologie

Het routeringsprobleem dat we behandelen luidt: vind in een gegeven netwerk de routes voor een gegeven set verkeersstromen tussen knopenparen zodat het pakketverlies, meer bepaald de maximale waarde ervan gevonden over alle stromen, zo laag mogelijk blijft. Het netwerk bestaat hierbij uit OPS-knopen met gekende performantie (i.h.b. in termen van pakketverlies) en tussenliggende verbindingen met gekende bandbreedte. Een meer formele beschrijving is:

- ***Gegeven:***
  - $G=(N,V)$ is een gegeven netwerk bestaande uit knopen $n \in N$ en verbindingen $v \in V$, ook wel takken of linken genoemd.

[9] Het is immers typisch voor IP dat de hoeveelheid verkeer die van een knoop A naar een knoop B stroomt kan verschillen van de hoeveelheid die in tegenovergestelde richting vloeit [211].

- $c(v)$ is de bandbreedte van verbinding $v \in V$
- $D_{b,e}$ is hoeveelheid verkeer die tussen beginpunt $b \in N$ en eindpunt $e \in N$ verstuurd moet worden
- PLR($L_f \mid L_1,...,L_F$) is de verlieskans die optreedt voor een verbinding, met belasting L_f, vertrekkend uit een knoop met in totaal F linken waarvoor de overige uitgaande linken een belasting L_i hebben ($i = 1...F$ en $i \neq f$)

- ***Gevraagd:***
  - vind een routering $x_{ij}(v)$, waarbij $i,j \in N$ en $x_{ij}(v)=1$ wanneer de stroom tussen i en j langsheen verbinding $v \in V$ passeert en anders $x_{ij}(v)=0$ (wat impliceert dat we een vraag niet zullen opsplitsen in deelstromen) en wel
  - zo dat $\max\{\text{PLR}(L_v|...), v \in V\}$ minimaal is.
- ***Onder randvoorwaarden:***
  - bandbreedte respecteren:

$$L_v = \frac{\sum_{b,e} f_{b,e}(v)}{c(v)} \leq 1, \quad \forall v \in V ,$$

$$\text{met } f_{b,e}(v) = D_{b,e} \cdot x_{b,e}(v)$$

  - alle vraag wordt gerouteerd tussen de gevraagde knopen:

$$\sum_{v \in I(n)} f_{b,e}(v) - \sum_{v \in I'(n)} f_{b,e}(v) = \begin{cases} D_{b,e} & \text{voor } n = b \\ 0 & \text{voor } n \neq b, e \\ -D_{b,e} & \text{voor } n = e \end{cases},$$

    waarbij *I(n)* de verzameling takken is die vertrekken uit knoop *n*, en *I'(n)* de verzameling takken toekomend in *n*.

Het bepalen van de verlieskans, i.e. de formulering van de PLR-functie wordt behandeld in een volgende Sectie 4.4. Eerst zullen we de algoritmes bespreken die we uitdachten om voor bovenstaand probleem een oplossing te vinden.

4.2.1 Heuristische routeringsalgoritmes

Als oplossing van bovenstaand probleem hebben we gekozen voor een heuristische aanpak. We hanteren een zogenaamde zoom-in filosofie [22], waarbij in een eerste fase vanuit een *globaal* perspectief het volledige netwerk bekeken wordt. De tweede, *lokale*, fase zoomt in op een beperkt deel van het netwerk om de oplossing te verfijnen. Elk van deze fases is iteratief: vertrekkend van de voorlopige oplossing, die bekomen werd in een voorafgaande stap, wordt de route voor een of meerdere vragen aangepast om een lagere verlieskans te bekomen. De initiële oplossing waarvan we vertrekken is de kortste-pad routering op basis van het aantal knopen die doorkruist worden om een gevraagde verkeersstroom te realiseren.

In de *globale fase* wordt gezocht naar de link tussen twee knopen waar het hoogste pakketverlies optreedt. Door het aanpassen van het takgewicht van enkel die link en het herberekenen van alle routes, zullen we trachten de maximale verlieskans die optreedt over het gehele netwerk te reduceren. Het spreekt voor zich dat na het

herrouteren de link met aangepast gewicht niet langer diegene hoeft te zijn waar het maximale verlies—dat nu lager ligt—optreedt. Het aanpassen van het takgewicht, wat origineel 1 bedraagt voor alle verbindingen, gebeurt door dit gewicht te verhogen met een zogenaamde strafmaat *P*. De waarde van *P* wordt zo bepaald dat een maximale reductie van de globale verlieskans gerealiseerd wordt. We herhalen het procédé van zoeken naar de link met maximale verlieskans en aanpassen van zijn gewicht, tot we er niet meer in slagen de maximaal optredende verlieskans nog te verlagen.

Na onderzoek voor een aantal test-scenario's bleek de reductie van het pakketverlies met een globale aanpak slechts beperkt. Dit ontspruit aan de gelimiteerde flexibiliteit die geboden wordt door routering van alle vragen met een zelfde kortstepad algoritme. Daarom gaan we vervolgens over tot een *lokale fase*, waarin niet langer alle vragen tegelijk geherrouteerd worden. We beschouwden twee algoritmische varianten hiervoor: SlechtsteLink en SlechtstePad.

In de SlechtsteLink-variant gaan we net als in de globale fase op zoek naar de verbinding met hoogste verlieskans. Maar nu zullen we de vragen die deze link doorkruisen één per één (in plaats van allemaal tesamen zoals in de globale stap), in een willekeurige volgorde, trachten te herrouteren. We doen dit opnieuw door het takgewicht te verhogen, maar nu enkel voor de op dat ogenblik beschouwde vraag.

Het SlechtstePad-algoritme gaat op zoek naar de verkeersstroom die het meeste pakketverlies ondervindt en sorteert de stromen daartoe volgens dalende PLR. Voor deze stroom wordt een ander pad gezocht door (enkel voor die stroom) aan linken van het originele pad hogere takgewichten toe te kennen. We beginnen hierbij met de link die op dat pad de hoogste verlieskans vertoont, om zo de linken te ontlasten die het ergst lijden aan pakketverlies. Wanneer we er na een vast aantal pogingen niet in slagen de maximale PLR optredend in het netwerk te verlagen, gaan we over tot de volgende verkeersstroom in de lijst.

We beëindigen[10] de zoektocht naar geschikte routes wanneer we er niet in slagen de maximaal optredende verlieskans nog te verlagen. Voor de SlechtsteLink-variant zal dit gebeuren wanneer, na het herrouteren van sommige vragen die de verbinding met hoogste verlieskans doorkruisen, alle andere verbindingen nog steeds een lagere verlieskans vertonen. In geval van SlechtstePad zal het algoritme eindigen wanneer alle stromen aan bod zijn gekomen.

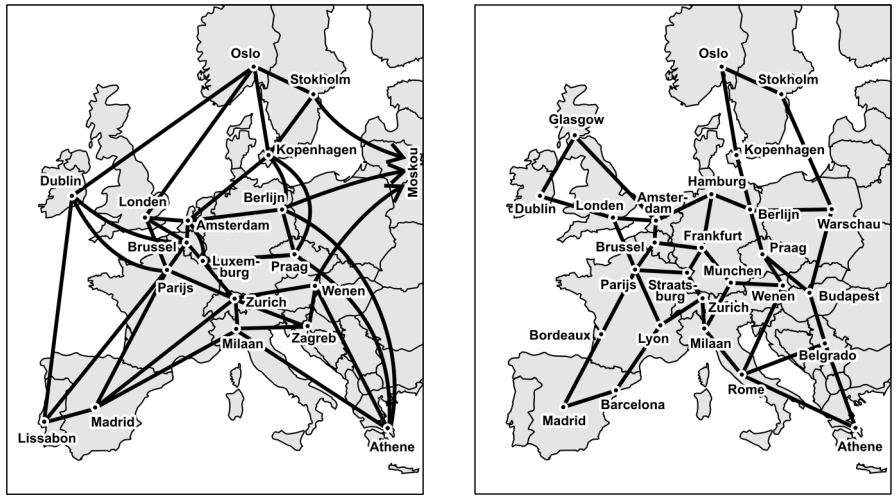
4.2.2 Netwerken en vraagmatrices

Om de prestatie van de algoritmes te evalueren zullen we de routes bepalen in twee pan-Europese netwerken. Het eerste, geschetst in Figuur 4.1a, verbindt negentien steden met veertig linken. Het tweede netwerk in Figuur 4.1b heeft een topologie ontleend aan een studie [212] kaderend in een samenwerking tussen twee Europese onderzoeksprojecten (Lion en Cost266) en gebruikt evenveel verbindingen, maar

[10] Bemerk dat het algoritme nooit in oneindige lussen verzeild zal geraken: we zullen nooit stromen verleggen van pad A naar pad B en dan terug van B naar A. Immers, wanneer we van A naar B overgaan, gebeurt dit enkel als daarmee de verlieskans strikt daalt.

tussen een groter aantal steden. De vraagmatrices die we beschouwen voor onze resultaten zijn:

- ***Strikt uniform (uni):*** Elke verbinding heeft dezelfde bandbreedte: alle $D_{b,e}$ waarden zijn dezelfde.
- ***Random uniform (rnd):*** We laten elke vraag $D_{b,e}$ genereren op basis van een uniforme distributie tussen 0 en een maximale waarde die afhangt van de gewenste netwerkbelasting.
- ***Random niet-uniform (rndG):*** We splitsen de vragen $D_{b,e}$ willekeurig op in twee groepen, waarbij we 96% van de vragen genereren op basis van een uniforme distributie over [0,7] en de overige 4% uniform kiezen uit [8,11]. Deze laatste vragen zijn de zogenaamde "grote vragen" (vandaar G). Om een bepaalde netwerkbelasting te bekomen vermenigvuldigen we de aldus bekomen waarden met een gepaste constante.
- ***Reëel (rl):*** Dit is de verkeersmatrix uit studies die kaderen in het Lion project, met name de waarden voor het jaar 2001 [212], die we met een constante vermenigvuldigen om een gewenste gemiddelde netwerkbelasting te verkrijgen.



(a) (b)

Figuur 4.1: Twee pan-Europese netwerken: (a) 19 knopen, (b) 27 knopen.

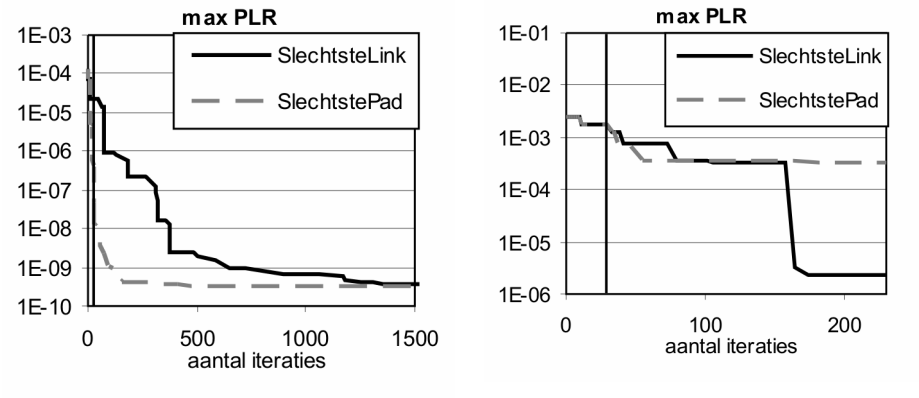
4.3 Evaluatie van de lokale fase algoritmes

Het spreekt voor zich dat we, vooraleer het hoofd te buigen over de voordelen van een PLR-gebaseerd routeringsalgoritme, hebben achterhaald welk van de twee varianten voor de lokale fase het efficiëntst bleek.

Een illustratie van de evolutie van de maximaal optredende verlieskans in het netwerk is in Figuur 4.2 gegeven. De verlieskans wordt er gedurende opeenvolgende

iteraties[11] uitgezet voor twee vraagmatrices van het type Random Niet-Uniform. We merken vooreerst op dat het aandeel in de PLR-reductie gerealiseerd door de globale fase beperkt is (aangegeven door een vertikale lijn en in beide gevallen slechts een paar tiental iteraties). Wat het verschil in evolutie tussen SlechtsteLink en SlechtstePad betreft, stellen we vast dat SlechtstePad sneller tot een lagere verlieskans komt. Dit vindt zijn oorsprong in het feit dat dit algoritme zijn aandacht richt op die stromen waarvoor de verlieskans het hoogst is. Voor SlechtsteLink is dit niet noodzakelijk zo: alle stromen die de verbinding passeren waar verlies het meest optreedt komen in aanmerking om geherrouteerd te worden.

Het finale resultaat, i.e. de maximale PLR die aan het einde bereikt wordt, is in de meerderheid van de gevallen gelijk voor beide varianten (bv. Figuur 4.2a). In een klein aantal gevallen evenwel slaagt SlechtsteLink erin een reductie van de verlieskans te bekomen die soms een grootte-orde lager ligt, zoals in Figuur 4.2b. De verklaring hiervoor is dat in het SlechtstePad-algoritme een verkeersstroom slechts één keer beschouwd wordt om er een andere route voor te vinden, namelijk wanneer de stroom de hoogste verlieskans ondervindt. Bij SlechtsteLink kan een zelfde stroom meermaals van route veranderen. Of SlechtstePad er al dan niet in slaagt dezelfde lage verlieskans te bereiken, hangt af van het beschouwde netwerk, de vraagmatrix en de willekeurige volgorde waarin de stromen ge(her)routeerd worden.



(a) (b)

Figuur 4.2: Evolutie van de maximale verlieskans waargenomen in opeenvolgende iteraties.

Het mag ook duidelijk zijn dat, zelfs wanneer de uiteindelijke maximale verlieskans die optreedt voor beide varianten gelijk is, de gevolgde route voor een gegeven vraag niet noodzakelijk in beide gevallen dezelfde is. Wanneer we de gebruikte capaciteit, zijnde de gebruikte bandbreedte gesommeerd over alle linken, berekenen, vinden we dat de SlechtstePad-variant er gemiddeld iets meer nodig heeft (tussen de twee en vijf procent).

Wanneer we het aantal iteraties beschouwen, zien we dat SlechtstePad er meer nodig heeft om tot een einde te komen. Het algoritme zal immers alle stromen een

[11] Met één iteratie bedoelen we één keer (pogen tot) herrouteren van een vraag.

aantal keer proberen herrouteren, terwijl het aantal iteraties in SlechtsteLink bepaald wordt door het aantal linken. Het is duidelijk dat het aantal linken doorgaans beduidend lager zal liggen dan het aantal verkeersstromen (dat van de orde $|N|^2$ is, met $|N|$ het aantal elementen in N). Anderzijds is het wel zo dat SlechtstePad binnen een relatief klein aantal iteraties al zijn laagste verlieskans bereikt, waar dit voor SlechtsteLink iets meer stappen vereist (cf. Figuur 4.2).

Uit deze vergelijking kunnen we besluiten dat de SlechtsteLink variant op basis van de kwaliteit van de gevonden oplossing te verkiezen is: we bereiken er de laagste verlieskans mee en de geconsumeerde bandbreedte in het netwerk ligt een tikkeltje lager.

4.4 Voordelen van het schatten van het pakketverlies

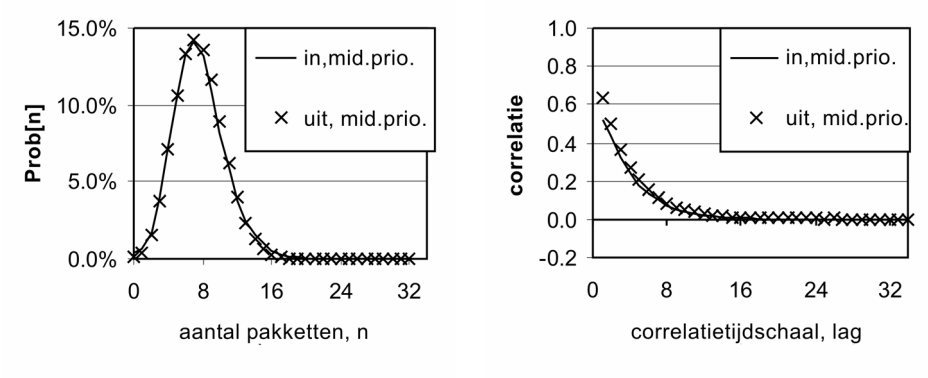
4.4.1 Schatting van pakketverlies

Zoals mag blijken uit de probleemstelling gaan we ervan uit dat we beschiken over een formule die ons het pakketverlies geeft dat optreedt op elk van de uitgaande linken van een optische pakketschakelaar, in functie van de belasting op elk van die linken. Aangezien die formule in elke iteratie van ons heuristisch algoritme opnieuw het pakketverlies op elke link moet geven, is een analytische gedaante gewenst. De performantie op netwerk-niveau in elke iteratie opnieuw bepalen door simulatie zou immers te tijdrovend zijn. Desalniettemin zijn de simulatieresultaten van een alleenstaande pakketschakelaar, zoals in vorig hoofdstuk, wel bruikbaar om tot zo'n formule te komen.

In de beschouwde netwerken is het uiteraard zo dat verkeer aan de ingang van zo'n pakketschakelaar minstens ten dele afkomstig is van de uitgaande vezels van een andere schakelaar. Opdat de resultaten van een enkelvoudige knoop nuttig zouden zijn voor de netwerk-scenario's, mag het verkeersprofiel van een stroom die de schakelaar doorkruist dus niet significant wijzigen. We hebben dan ook geverifiëerd of dit zo is. Voor de verschillende traffiektypes uit Hoofdstuk 3 hebben we per prioriteitsklasse het aantal pakketten bijgehouden dat in elk tijdslot de schakelaar binnenkomt en verlaat via elk van de vezels. We hebben dit aantal pakketten dat in een zelfde tijdslot de schakelaar binnenkomt n_T genoemd. Om het aankomstenproces te karakteriseren hebben we onze aandacht gevestigd op de probabiliteitsverdeling van n_T en de correlatie tussen het aantal pakketten in tijdsloten die *lag* uiteen liggen: tussen n_T en n_{T+lag}.

De schakelaar zal het profiel van een verkeersstroom op twee manieren kunnen beïnvloeden: door pakketten verloren te laten gaan en door pakketten te vertragen via de buffer. Pakketverlies zal voornamelijk impact hebben op de probabiliteitsdistributie van n_T , terwijl vertraging eerder de correlatiestructuur veranderen kan: buffering kan pakketten meer in de tijd spreiden. Laatstgenoemd effect zal het sterkst optreden voor de Balans-strategie met een Incr-buffer. Een voorbeeld van de opgetekende histogrammen voor n_T en de correlatie is geschetst in Figuur 4.3 voor het GeoAanUit-model in een schakelaar met Incr-buffer van B=64 bufferpoorten,

gebruik makend van de Balans-strategie. Onze intuïtieve verwachtingen worden er bevestigd: het pakketverlies zorgt voor een beperkte afname van de probabiliteiten voor hogere n_T waarden en de buffer zorgt voor een lichte wijziging van de correlatie tussen n_T en n_{T+lag}. De impact van de schakelaar is evenwel beperkt en we kunnen met voldoende vertrouwen stellen dat hetzelfde model gebruikt kan worden om zowel in- als uitgaand verkeer te beschrijven: noch de vorm van histogrammen, noch die van de correlatie worden grondig veranderd.



(a) (b)

Figuur 4.3: Illustratie van het aankomstenproces en de corresponderende karakteristieken van het uitgaande verkeer: (a) aantal pakketten n_T dat aankomt per tijdslot; (b) correlatie tussen n_T en n_{T+lag}.

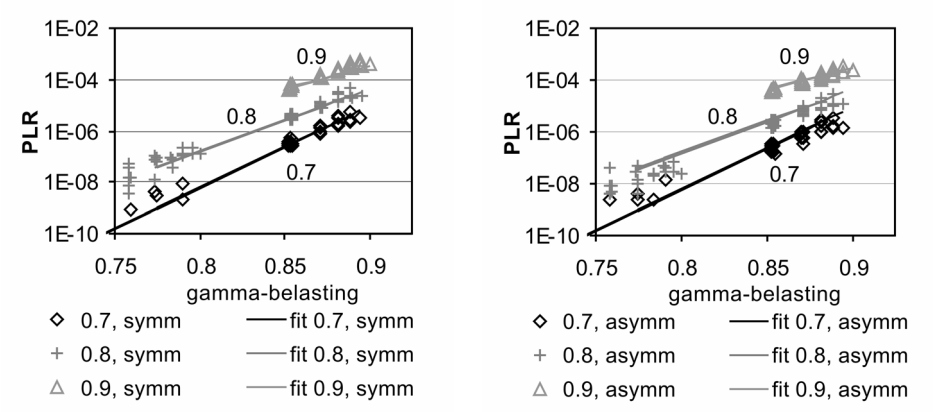
Nu de zin van simulaties voor een enkelvoudige knoop bevestigd is, willen we zulke simulaties gebruiken om tot een gesloten gedaante te komen voor een functie die het verlies geeft in functie van de belasting van de uitgaande linken. En dit voor een voldoende wijd bereik aan belastingen, die uiteraard niet voor alle uitgaande linken dezelfde hoeven te zijn. Het is duidelijk dat de belasting op een bepaalde uitgaande link van doorslaggevend belang zal zijn voor het pakketverlies dat op die link optreedt. Maar aangezien het verkeer bestemd voor die link de buffer zal moeten delen met andere verkeersstromen, zal er toch een zekere afhankelijkheid zijn van de belasting op andere linken. Om die wisselwerking te vatten hebben we de empirische gedaante van een verlieskansformule voorgesteld zoals gegeven in vergelijking 4.1, waar L_f staat voor de belasting op uitgaande vezel f. De maat L_γ is een indicator van de gemiddelde belasting van de schakelaar, waarbij meer gewicht gehecht wordt aan hoog belaste vezels. Het is immers het verkeer dat bestemd is voor die vezels dat intensief van de buffer gebruik zal maken en daardoor de verlieskans op andere vezels kan beïnvloeden.

$$\mathrm{PLR}(L_f \,|\, L_1 \ldots L_F) \approx \mathrm{g}(L_f, L_\gamma) \quad \text{met } L_\gamma = \sqrt[\gamma]{\sum_{i=1}^{F} L_i^{\,\gamma} \Big/ F} \qquad 4.1$$

Om in te schatten of een dergelijke gedaante voor de verlieskansfunctie zinvol is, en zo ja, een idee te krijgen over een mogelijke analytische vorm van de functie g

uit bovenstaande vergelijking (4.1), hebben we een reeks simulaties uitgevoerd voor elke mogelijke combinatie van lasten L_f genomen uit {0, 0.7, 0.8, 0.9}. We hebben de verlieskansen bepaald voor elk van die combinaties, gegenereerd door twee types traffiekmatrices. In de eerste, *symm* genaamd, draagt elke ingangsvezel *i* in dezelfde mate bij tot de belasting van een gegeven uitgangsvezel *f*: de fractie van de belasting L_f gegenereerd door ingangsvezel *i* bedraagt L_f / F. Het tweede type traffiekmatrix *asymm* wordt gekarakteriseerd door asymmetrische bijdragen van de onderscheiden ingangsvezels: het verkeer gegenereerd voor een bepaald koppel in- en uitgangsvezels (*i*,*f*) wordt gegeven door vergelijking 4.2.

$$L_{i,f} = \begin{cases} L_f \big/ 2^{1+(i-f) \bmod F} & , i \neq f \\ L_f \big/ 2^{F-1} & , i = f \end{cases} \qquad 4.2$$



(a) (b)

Figuur 4.4: Verlieskansen en gefitte curves in functie van L_γ: (a) voor een *symm* verkeerspatroon waar elke ingangsvezel in dezelfde mate bijdraagt tot de belasting op een uitgaande vezel; (b) voor een *asymm* verkeerspatroon waar dit niet zo is.

De resulterende grafieken die de verlieskans uitzetten in functie van de gewogen last L_γ zijn gegeven in Figuur 4.4 voor een schakelaar met F=6 in- en uitgangsvezels die elk W=32 golflengten dragen en een Fix-buffer met B=32 bufferpoorten. Door γ te variëren zagen we dat voor relatief hoge waarden (orde 20 en meer) de verlieskansen voor verschillende L_f waarden sterk uiteen lagen, terwijl die voor gelijke (L_γ , L_f) waarden sterk geconcentreerd waren. Vermits de curves voor constante L_f zich in een logaritmische grafiek quasi tot een rechte herleiden, zijn we gekomen tot de gedaante beschreven in vergelijkingen 4.3–4.5 voor de verlieskansfunctie. De waarden voor de onderscheiden parameters werden bekomen met de kleinste kwadratenmethode en zijn samengevat in Tabel 4.1. We gebruikten de resulterende verlieskansfunctie al voor de resultaten uit vorige sectie en zullen dat blijven doen in het resterende deel van dit hoofdstuk.

$$\mathrm{PLR}(L_f \,|\, L_1 \ldots L_F) \approx \mathrm{p}(L_f) \cdot 10^{\mathrm{r}(L_f)\cdot(L_\gamma - L_f)} \qquad 4.3$$

$$\mathrm{p}(L_f) = \mathrm{PLR}(L_f \mid \text{uniform}) = a \cdot {L_f}^{b} \cdot 10^{c - L_f} \qquad 4.4$$

$$\mathrm{r}(L_f) = u \cdot L_f + v \qquad 4.5$$

Tabel 4.1: Gefitte waarden van de parameters voor vergelijkingen 4.1, 4.3–4.5.

Parameter	*Waarde*	*Parameter*	*Waarde*
γ	32.97	a	62.59
u	71.58	b	189.15
v	-82.77	c	-63.81

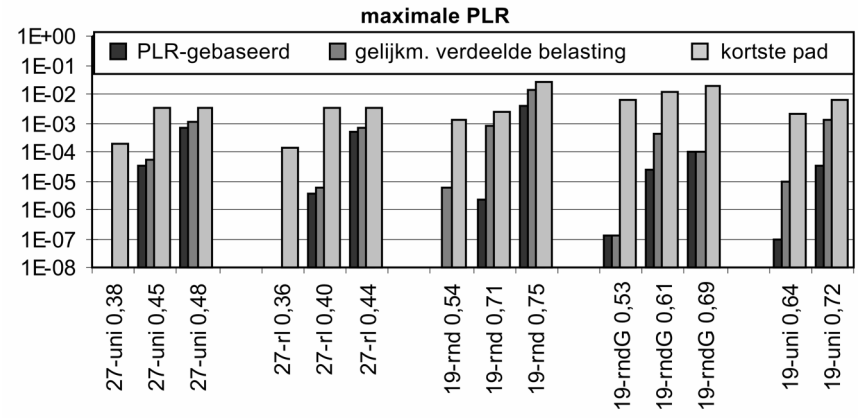
4.4.2 Vergelijking met kortste-pad routering en gelijkmatig verdeelde belasting

Om de zin of onzin van het beschouwde routeringsalgoritme te achterhalen hebben we de resulterende maximale verlieskans vergeleken met minder intelligente strategieën. Het ligt voor de hand de reductie van het pakketverlies te beschouwen ten opzichte van zuivere kortste-pad routering. Met de aldus gevonden routes starten we ons streven naar lager pakketverlies.

Zoals aangegeven bij het zoeken naar een analytische gedaante van het pakketverlies voor een gegeven link, is een doorslaggevende factor de belasting op die link zelf. Dit is uiteraard de reden waarom het gelijkmatig verdelen van het verkeer (Eng. load balancing) dikwijls wordt nagestreefd. Merk op dat de algoritmes opgesteld in Sectie 4.2.1 aangepast kunnen worden om zulke gelijkmatige verdeling te bekomen. Het volstaat in de beschrijving “verlieskans” te vervangen door “belasting”: we zullen dan met het algoritme het minimaliseren van de maximaal voorkomende linkbelasting nastreven. Vergelijking van de hiermee gerealiseerde verlieskansen met diegene die bereikt waren door het oorspronkelijke algoritme zal dan leren hoe belangrijk een nauwkeurige schatting van de verlieskans is om die zo laag mogelijk te houden. De linkbelasting kan immers gezien worden als benadering van eerste orde voor de verlieskans.

In Figuur 4.5 zijn de waarden geschetst van de maximale verlieskans die optreedt wanneer de routes respectievelijk met het PLR-gebaseerde algoritme, het gelijkmatig verdelen van de last en de kortste-pad routering bepaald worden. We stellen tot onze opluchting vast dat de reductie van de verlieskans door een intelligente routering tegenover kortste-pad routering aanzienlijk is. Het verschil tussen kortste-pad routering en ons PLR-gebaseerd algoritme kan tot meerdere grootteordes bedragen. Het is duidelijk dat het meeste voordeel behaald kan worden wanneer de gemiddelde netwerkbelasting eerder laag is: er zijn dan voldoende linken met nog vrije capaciteit om door herrouteren de links waar veel verkeer doorstroomt te ontlasten. Om dezelfde reden zien we ook dat bij niet-uniforme verkeerspatronen (Random Niet-Uniform, rndG) de reductie zelfs bij relatief hoge

gemiddelde netwerkbelasting nog aanzienlijk is. Vanzelfsprekend zal een intelligente routering nauwelijks baten wanneer het netwerk nagenoeg ten volle belast is.

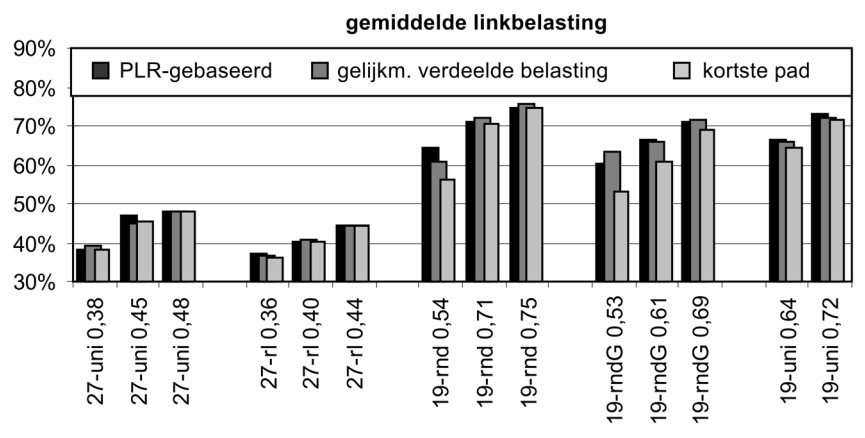


Figuur 4.5: Vergelijking van de maximaal optredende PLR voor de verschillende routeringsalgoritmes. De labels op de X-as geven het scenario aan: het aantal knopen in het netwerk, het verkeerspatroon, de gemiddelde netwerkbelasting.

De vergelijking met de resultaten die behaald werden door gelijkmatige verdeling van de belasting leert dat het wel degelijk zinvol is om van een nauwkeurigere schatting van de verlieskans gebruik te maken. Hoewel ook de verdeling van de belasting, vergeleken met kortste-pad routering, significante reducties van de verlieskans oplevert, stellen we vast dat de kloof met PLR-gebaseerde routering, hoewel vaak beperkt (factor 1.5), soms nog aanzienlijk kan zijn (tot twee grootteordes toe in de onderzochte gevallen). De verklaring ligt in het feit dat enkel de belasting van de link zelf als indicatie van pakketverlies niet volstaat: op die manier wordt interactie met andere stromen, veroorzaakt door het delen van een zelfde (beperkt) buffer, verwaarloosd. Wanneer de belasting laag is (cf. de gevallen met gemiddelde netwerkbelasting van de orde 0.35 à 0.5), is het verschil met een nauwkeurige PLR-schatting beperkt tot een factor 1.5 à 2: in dit geval wordt de buffer niet zwaar gebruikt, met een nauwelijks merkbare correlatie tussen de verlieskansen op verschillende uitgaande vezels tot gevolg. Wanneer de netwerkbelasting evenwel hoger oploopt, zal de buffer zo vaak benut worden dat de interactie niet langer verwaarloosbaar is. Zoals eerder gesuggereerd zal de discrepantie opnieuw verdwijnen voor uitermate hoge netwerkbelasting, vermits intelligente routering daar nauwelijks nog baat.

Bemerk dat het verschil in bereikte verlieskansen tussen PLR-gebaseerd routeren en gelijkmatig verdelen van de last vooral in het oog springt voor de verkeerstypes Random (rnd) en Uniform (uni). De verklaring is dat in deze gevallen de beschouwde netwerkbelasting betrekkelijk hoog ligt, waardoor verwaarlozing van de interactie tussen stromen op verschillende uitgaande vezels duidelijk nadeel oplevert. Voor het Random Niet-Uniforme (rndG) type ligt de belasting ook vrij

hoog, maar aangezien de individuele verkeersstromen meer bandbreedte vragen kunnen ze niet eenvoudig geherrouteerd worden naar lager belaste linken.



Figuur 4.6: Vergelijking van de gemiddelde linkbelasting voor de verschillende routeringsalgoritmes.

De prijs die betaald wordt door intelligente routering om lagere pakketverliezen te bereiken is uiteraard dat afgeweken wordt van kortste-pad routering en de totale gebruikte bandbreedte hoger komt te liggen. Om een indicatie te krijgen van die toename, hebben we in Figuur 4.6 de benutte fractie van de aanwezige capaciteit uitgezet—wat equivalent is met de gemiddelde linkbelasting. Het blijkt dat in de door ons beschouwde gevallen de toename in bandbreedtegebruik relatief beperkt blijft (een verhoging met maximaal 7% belasting). Het is niet verwonderlijk dat in het bijzonder voor het Random Niet-Uniform (rndG) vraagpatroon die toename hoog is, vermits daar de geherrouteerde stromen de meeste bandbreedte verbruiken. Of een PLR-gebaseerd algoritme dan wel een gelijkmatige lastverdeling leidt tot de hoogste toename, hangt af van het beschouwde geval, vermits de volgorde waarin vragen geherrouteerd worden kan verschillen.

4.5 Besluiten

We hebben in dit hoofdstuk ons werk omtrent het intelligent routeren in een OPS-netwerk uiteengezet. Gebaseerd op simulaties hebben we allereerst een methode ontwikkeld om het pakketverlies nauwkeurig te schatten voor verkeersstromen in zo'n netwerk. We toonden aan hoe een routeringsalgoritme van deze schattingen gebruik kan maken om routes te vinden voor een gegeven set verkeersstromen met het oog op het minimaliseren van de verlieskansen. De reductie van dat verlies ten opzichte van kortste-pad-routering was aanzienlijk (verschillende grootteordes). Vergelijking met een routeringsalgoritme om de belasting zo uniform mogelijk te spreiden over het netwerk leerde dat ook dit een beduidende verlaging van het verlies oplevert. Maar aangezien de linkbelasting slechts een indicator is van de

verlieskans en de wisselwerking—door het delen van een buffer in de schakelknopen—tussen verschillende linken er niet door in rekening wordt gebracht, blijkt een meer nauwkeurige PLR-schatting toch nog voordeliger, soms zelfs tot een paar grootteordes.

Bemerk dat we in dit hoofdstuk de vragen tussen knopenparen als ondeelbaar en allen van dezelfde prioriteit hebben beschouwd. Voor een bespreking van het geval met meerdere verkeersstromen tussen eenzelfde knopenpaar en meerdere prioriteiten, verwijzen we naar Appendix H.

Hoofdstuk 5
Interactie met het Transport Controle Protocol

5.1 Situering

In vorig hoofdstuk behandelden we algoritmes die bepaalden hoe verkeer best doorheen een netwerk gerouteerd wordt. Voor een relatief statische verkeerssituatie zoals we die er impliciet onderstelden zal die routering lange tijd onveranderd blijven. Een ogenblik waarop ze drastisch veranderen kan op korte tijdschaal, is wanneer er in het netwerk een fout optreedt, bijvoorbeeld doordat bij graafwerken een kabel met optische vezels gebroken wordt. Gezien de omvang van de hoeveelheid verkeer die hierdoor verloren kan gaan, heeft men verschillende herstelmechanismen bedacht [206,213,214]. Zo'n mechanisme zal ervoor zorgen dat verkeer de falende netwerkdelen omzeilt en er zo toch nog in slaagt z'n bestemming te bereiken.

Wat er precies gebeurt met verkeer dat moet omgeleid worden van een primair pad naar een herstelpad zal afhangen van de gebruikte technologie en in het bijzonder of die een circuit- dan wel een pakketgeschakeld concept implementeert. Bemerk dat we bij onze bespreking optische netwerken onderstellen, maar de resultaten kunnen even goed gebruikt worden om bv. elektrische pakketschakelende netwerken met circuitschakelende te vergelijken (zie Appendix I). Aan de fundamentele verschilpunten tussen die circuit- en pakketschakelende concepten zullen we hier bijzondere aandacht besteden vanuit het standpunt van herstel.

De herstelacties hebben hun weerslag op de performantie die gebruikers van het netwerk ervaren. Zoals eerder aangehaald vormt IP-verkeer het grootste deel van wat de gebruikers door het netwerk willen transporteren. Binnen IP domineert één subtype, met name het TCP-verkeer, wat volgens recente metingen verantwoordelijk is voor meer dan 95% van al het IP-verkeer gemeten in bytes [215]. Nu is het TCP-protocol reactief van aard, wat betekent dat het zich aanpast aan de beschikbare bandbreedte. Herstel- en andere acties die de routes voor zo'n TCP-verbindingen veranderen, kunnen dan reactie uitlokken van de adaptieve mechanismen eigen aan TCP. We zullen in dit hoofdstuk zulke effecten behandelen.

In de volgende Sectie 5.2 geven we een beknopt overzicht van de reactieve natuur van TCP, wat zich situeert op de zogenaamde transportlaag in het klassieke

gelaagde OSI-model [216]. Een netwerk zal instaan voor het verzorgen van de vereiste connectiviteit tussen eindpunten van communicatiepartners op de toepassingslaag. Twee fundamenteel verschillende schakeltechnologiën voor zo'n netwerk zullen we behandelen in Sectie 5.3, met bijzondere aandacht voor de gebruikelijke herstelmechanismen. De eigenlijke resultaten met betrekking tot de effecten van die mechanismen op het TCP-gedrag worden uiteengezet in 5.4. Besluiten staan samengevat in Sectie 5.5.

De eigen publicaties waarop dit hoofdstuk gestoeld is, zijn [4,16] (waarvan de eerste te lezen staat in Appendix I) en [1,3] voor enkele kanttekeningen.

5.2 TCP/IP

Het Transmissie Controle Protocol (TCP) is connectie-georiënteerd: vóór het versturen van de eigenlijke gegevens wordt een verbinding opgezet via een drievoudig proces van spreekwoordelijk handjes schudden (Eng. three-way handshake) [102,216]. Belangrijker voor de studie in dit hoofdstuk is het feit dat TCP betrouwbaar is, in de zin dat de zender vereist dat hij van de ontvanger een bevestiging (Eng. acknowledgement, ACK) krijgt. De zender zal daarom, na een bepaald aantal nieuwe segmenten te hebben doorgestuurd, wachten tot hij ofwel de correcte ontvangst door de andere partij bevestigd krijgt via een ACK, ofwel bij dat wachten op een ACK z'n geduld verliest (bv. er gaat een timer af). In het laatste geval zal hij aannemen dat de gegevens verloren gingen en opnieuw proberen hetzelfde segment te zenden. De hoeveelheid gegevens die de bron mag versturen—vooraleer er gepauseerd wordt tot ontvangst van een ACK—hangt af van het congestievenster *cwnd* (Eng. congestion window), dat nooit de door de ontvanger bepaalde waarde *rwnd* (Eng. receiver window) zal overschrijden.

Hoewel er nog steeds onderzoek naar en ontwikkeling van TCP-varianten is (zie bv. een recent overzicht in [217]), werden de fundamenten ervan geruime tijd geleden vastgelegd in RFC 793 [218]. Ze kunnen worden samengevat in vier componenten [217]: (i) het additief toenemen, multiplicatief afnemen (Eng. additive increase, multiplicative decrease, AIMD) dat het congestievenster met één segment ter grootte van SMSS (Eng. Sender Maximum Segment Size) verhoogt per zogenaamde RTT (Eng. Round Trip Time), maar het venster halveert wanneer een segment verloren blijkt te zijn gegaan; (ii) het gebruik van een timer om segmenten opnieuw te versturen wanneer hun ontvangst niet bevestigd wordt binnen een zogenaamd RTO-interval (Eng. Retransmission Timeout); (iii) het trage start mechanisme (Eng. slow start) om bij aanvang van de communicatie af te tasten hoeveel bandbreedte beschikbaar is, tot het *cwnd* een grens bereikt die aangeduid wordt als *sstresh* (Eng. slow start threshold); en (iv) het principe van ACK-klokken, waarmee verwezen wordt naar het zenden van (nieuwe) segmenten als respons op de ontvangst van een ACK. In onze studie zullen we gebruik maken van de TCP NewReno versie, die beschreven wordt in de meest recente RFC omtrent TCP [219].

Het bovenstaande summiere lijstje van de voornaamste karakteristieken van TCP bevat aanwijzingen over hoe het TCP-gedrag beïnvloed kan worden door het falen van verbindingen en de daarop volgende herstel-acties die de route gevolgd door TCP-stromen veranderen. Een eerste effect is dat door het omschakelen naar

een andere route de RTT, dit is de tijd nodig om de weg van bron naar bestemming en terug af te leggen, verandert. De omleiding rond het falende netwerkgedeelte kan die RTT plotsklaps verhogen, waardoor in TCP de RTO-timer kan afgaan met het nodeloos heruitzenden van verloren gewaande segmenten tot gevolg. De wachttijd RTO wordt bekomen op basis van een schatting van de RTT die SRTT genoemd wordt (Eng. smoothed round trip time) [220,221]. Om zich voldoende snel aan herroutering door herstel-acties te kunnen aanpassen zou die SRTT dus niet al te traag de werkelijke RTT moeten volgen.

Een tweede effect op TCP is uiteraard dat door de faling een reeks segmenten verloren kan gaan, tot de vereiste herstel-acties voltooid zijn. Het verder gebruikte NewReno algoritme zal bij detectie van de verliezen terugvallen op een mechanisme van zogenaamd snel heruitzenden/snel herstel (Eng. fast retransmit/fast recovery). Maar in tegenstelling tot wat de naam laat uitschijnen is dit, afhankelijk van de hoeveelheid verloren gegane segmenten, niet altijd de snelste methode om de verliezen te boven te komen. In sommige gevallen, met name wanneer relatief veel gegevens verloren gaan, kan het voordeliger zijn om de RTO-timer te laten afgaan en op het trage start mechanisme te vertrouwen om de verkeersuitwisseling te herstellen. Wanneer we één enkele TCP-verbinding beschouwen, kunnen we de voorwaarde gegeven in vergelijking 5.1 afleiden voor de hoeveelheid verloren gegane gegevens G opdat het "fast retransmit/fast recovery"-mechanisme sneller de verliezen te bovenkomt dan het afgaan van de RTO-timer. (Zie Appendix I voor details.)

$$\frac{\max(0, sstresh - G) + (sstresh - SMSS)}{2} \cdot \frac{\min(sstresh, G)}{SMSS} \geq 2^{G+1} \cdot SMSS \qquad 5.1$$

In de praktijk zal een falende verbinding uiteraard een bundeling van vele TCP-stromen dragen, zodat het bepalen van de hoeveelheid verloren gegevens per individuele stroom niet eenvoudig bepaald kan worden. De reactie van zo'n bundel op falingen en herstel-acties is dan ook niet langer te vatten in analytische formules. Daarom zullen we, zoals verder beschreven, simulaties gebruiken om toch het gedrag van een aggregaat van TCP-stromen te kunnen evalueren.

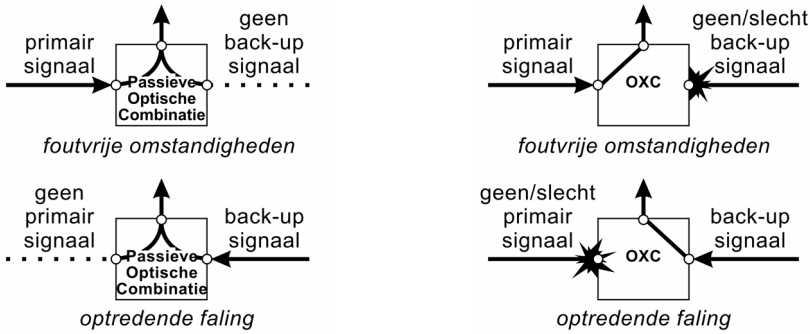
5.3 Herstel: circuit- versus pakketschakelen

Zoals uiteengezet in het inleidende Hoofdstuk 1 zijn voor het transportnetwerk twee concepten mogelijk: pakketschakelen ofwel circuitschakelen. Origineel werden de herstelmechanismen in het kader van MPLS ontwikkeld voor een (elektrisch) pakketgeschakeld netwerk. Terwijl de vertaling ervan naar optisch pakketschakelen evident is, zijn er voor het circuitgeschakelde alternatief van golflengteschakelen toch een aantal niet onbelangrijke verschilpunten [222].

Waar de routes in een pakketgeschakeld concept probleemloos kunnen overlappen en dezelfde golflengten kunnen gebruiken, is het samennemen van twee inkomende circuits in één enkel uitgaand signaal (van dezelfde bitrate) doorgaans niet mogelijk. Nochtans is dit wat typisch gebeuren moet in een schakelaar waar een omleiding voor herstel en een primair werkend pad samenkomen. In het geval van

optisch schakelen kan dit gerealiseerd worden door optische koppelaars zoals schematisch aangegeven in Figuur 5.1a. Een vereiste opdat dit werkbaar zou zijn is dat het werkende pad en het herstelpad nooit gelijktijdig een signaal dragen. Wanneer dit niet gegarandeerd kan worden is een volwaardige schakelaar nodig (Figuur 5.1b), of moeten voor werkende paden en herstelpaden afzonderlijke golflengtes gebruikt worden.

Dit brengt ons bij een tweede belangrijk onderscheid tussen pakketschakelende en circuitschakelende technologieën. Om snelle omschakeling in geval van falingen mogelijk te maken, kunnen herstelpaden op voorhand worden opgezet. Men spreekt dan van protectie (in tegenstelling tot restauratie [223]). In circuitschakelen impliceert dit het reserveren van bandbreedte, in het bijzonder een volledige golflengte, voor hersteldoeleinden. Voor pakketschakelen gebaseerd op MPLS verbruikt het op voorhand installeren van de herstelpaden evenwel geen bandbreedte: er worden in de schakelknopen enkel routeringstabellen ingevuld met de correcte afbeeldingen van inkomende labels op uitgaande poorten en eventueel nieuwe uitgaande labels. In [222] wordt getoond dat hierdoor in een circuitgeschakeld concept een grotere hoeveelheid bijkomende capaciteit voor protectie-doeleinden geïnstalleerd moet worden.



Figuur 5.1: Optisch samenvloeien van werkend en herstelpad: (a) met passieve koppelaars, of (b) een volwaardig schakel-element.

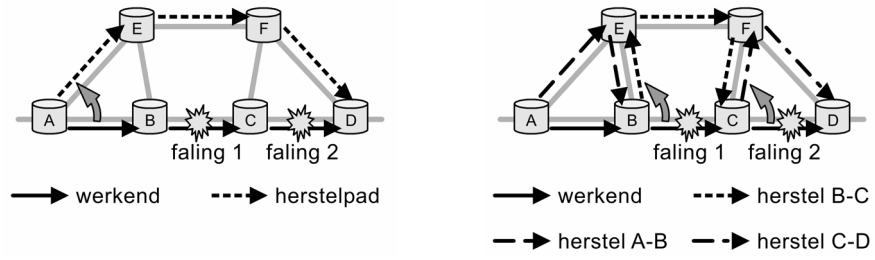
De GMPLS-protectiemechanismen die we verderop zullen beschouwen zijn allen gebaseerd op het vooraf opzetten van labelgeschakelde paden (waarbij voor circuitschakelen een label zal overeenkomen met een golflengte). Zo'n herstelpad zal ofwel een enkele link of knoop overspannen, ofwel de hele weg tussen bron en bestemming: in het eerste geval heeft men het over lokale protectie, terwijl het laatste padprotectie genoemd wordt [224,225]. Deze mechanismen zijn geïllustreerd in Figuur 5.2, waaruit blijkt dat in geval van lokale protectie een herstelpad voorzien moet worden per mogelijke link- of knoopfout[12] (de figuur beschouwt enkel linkfouten). Padprotectie vereist slechts één herstelpad per werkend pad, maar

[12] Merk op dat, wanneer zogenaamde labelstapeling (Eng. label stacking) mogelijk is, een herstelpad per beschouwde knoop- of linkfout kan gedeeld worden voor alle werkende paden die het falende netwerkelement doorkruisen [226].

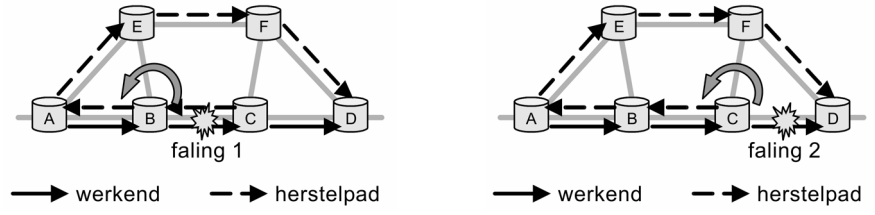
vertrouwt wel op signalisatie om de bronknoop van de verbindingen op de hoogte te brengen van de gewenste omschakeling naar het herstelpad. Lokale protectie-mechanismen laten toe dat de schakelacties genomen worden op de plaatsen waar de fout gedetecteerd wordt. Het vermijden van signalisatie kan tijd besparen en er zo voor zorgen dat minder gegevens verloren gaan omwille van een snellere reactietijd.

Een derde protectieschema, geïllustreerd in Figuur 5.3, werd gesuggereerd door Haskin en Krishnan [227] en tracht de voordelen van voorgaande schema's te verenigen: lokale teruglussing (Eng. local loop-back). De kerngedachte die aan de grondslag ligt is slechts één herstelpad per werkend pad te voorzien waarbij toch lokale omschakelacties volstaan (i.e. zonder signalisatie naar andere knopen te vereisen). Figuur 5.3 toont dat de omschakelactie zal verschillen naargelang van de optredende fout, maar steeds hezelfde herstelpad volgt: eerst terugkerend naar de bronknoop (in tegenovergestelde richting als het werkende pad) en vervolgens via een van het primaire pad gescheiden route naar de bestemming.

Ter afronding van dit korte overzicht van de beschouwde herstelmechanismen nog deze eindnoot: we zullen in onze studies enkel protectie beschouwen op één enkele laag, met name de optische transportlaag. Vanzelfsprekend kan er ook gedacht worden om de taak van het opvangen van netwerkfalingen te spreiden over meerdere lagen (bv. een elektrische IP/MPLS laag en een optische circuit-geschakelde laag), maar daar zullen we hier verder geen gewag van maken: we verwijzen met genoegen naar bv. [2] voor een discussie daaromtrent.



Figuur 5.2: Illustratie van (a) padprotectie en (b) lokale linkprotectie.



Figuur 5.3: Illustratie van lokaal teruglussen voor twee verschillende falingen.

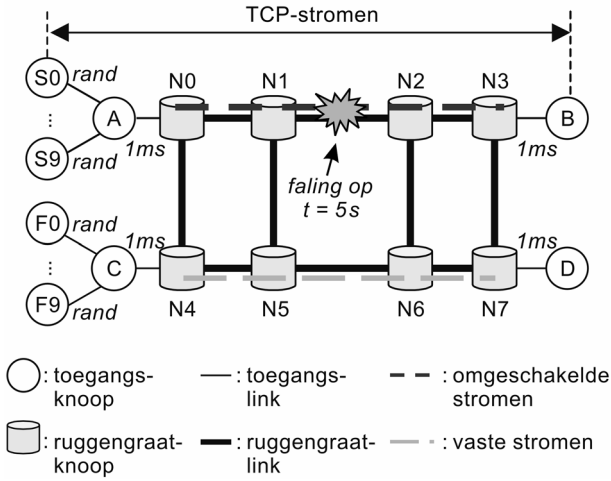
5.4 Effect van herstel-acties op TCP verkeer

5.4.1 Bestudeerde facetten

De hamvraag die we in dit hoofdstuk proberen te beantwoorden, is: hoe goed worden herstelacties, en de neveneffecten die ze teweeg brengen, verteerd door het belangrijkste protocol uit de transportlaag, TCP? We hebben al aangegeven dat het voornaamste effect van een faling en het daaropvolgende herstel is dat er een hoeveelheid verkeer verloren gaat en dat de gewijzigde route leidt tot een verandering van de RTT. We zullen het gedetailleerde onderzoek naar de invloed op een bundeling TCP-connecties opsplitsen in een aantal facetten: (i) de invloed van de omschakel-actie naar het herstelpad, (ii) de invloed van de veranderende RTT en (iii) het effect van het terugschakelen naar het oorspronkelijke pad eenmaal de faling hersteld is. Tot slot zullen we ook een gevalsstudie beschrijven die verifieert of de conclusies ook vertaald kunnen worden naar de protectiemechanismen beschreven in voorgaande sectie. Vooraleer de resultaten daarvan voor te stellen, zetten we nu eerst de gebruikte methodologie uiteen.

5.4.2 Methodologie

Het antwoord op de geformuleerde vragen en gevalsstudies zullen we zoeken door gebruik te maken van de behoorlijk populaire Netwerk Simulator, kortweg ns-2 [228]. Dit is een kosteloos beschikbare discrete-gebeurtenissen-simulator specifiek gericht op onderzoek van (computer)netwerken en ondersteunt onder meer TCP, IP-routering en multicast protocollen over zowel bedrade als draadloze netwerken.



Figuur 5.4: Schets van de simulatie-opstelling.

Om een bepaald herstelmechanisme, of één van de parameters die het TCP-gedrag zal bepalen, te bestuderen, zullen we steeds gebruik maken van dezelfde

generische topologie (of een deel daarvan), zoals geschetst in Figuur 5.4. De gebruikte parameters gemeenschappelijk aan alle simulaties staan opgesomd in Tabel 5.1. We hebben ervoor gekozen de flessenhals—die de verliezen veroorzaakt die de bandbreedte ingenomen door een TCP-verbinding beperkt—in het toegangsnetwerk te lokaliseren (tenminste wanneer geen falingen optreden). Daartoe gaven we toegangslinken een lagere bandbreedte. De propagatietijd voor de ruggengraatlinken werd genomen ergens tussen 1 en 60 ms, afhankelijk van het beschouwde scenario, zodat de totale RTT over de verschillende TCP-connecties variëerde van een paar tien- tot een paar honderdtal milliseconden.

De pakketten zullen vooraf bepaalde routes volgen, waarvoor we de MPLS-voorzieningen van de simulator gebruikt hebben—het spreekt voor zich dat de precieze route voor de herstelpaden zal afhangen van het beschouwde herstelmechanisme. Langs deze paden zullen we een aggregaat van TCP-stromen sturen. Vermits zulke TCP-verbindingen hun oorsprong vinden buiten het ruggengraatnetwerk en zo duidelijk verschillende eind-tot-eind vertragingen kunnen hebben, laten we ze vertrekken in verschillende toegangsknopen, verbonden met de rest van het netwerk met linken van verschillende lengte en dus propagatievertraging: we kozen die willekeurig volgens een uniforme distributie over [10ms,100ms]. Daarenboven lieten we de TCP-bronnen met zenden beginnen op willekeurige ogenblikken, om kunstmatige synchronisatie te vermijden. Zodoende vulden we het netwerk met een mix van TCP-bronnen met ongecorreleerde starttijdstippen en verschillende RTT's. De TCP-stromen zullen, gezien de relatie tussen RTT en zowel de RTO als het ACK-klokken, met verschillende snelheid reageren op de falingen en herstelacties.

Tabel 5.1: Gemeenschappelijke simulatieparameters.

Parameter	*Waarde*
TCP	TCP NewReno [219]
aantal bronknopen	10 voor elk pad
aantal TCP-stromen	10 voor elke bronknoop (dus 100 per pad)
bandbreedte toegangslink	80% van de bandbreedte van een link in het ruggengraatnetwerk (90% voor de studie van tijdseffecten in Sectie 5.4.3)
rand vertraging toegangslink	random gegenereerd o.b.v. een uniforme distributie in het interval [10ms, 100ms]

De twee belangrijkste parameters die we zullen variëren om de TCP-effecten te beoordelen zijn:

- ***Omschakeltijd (δ):*** Dit is de tijd die verstrijkt tussen het optreden van een linkfaling en de omschakeling van de getroffen verkeersstromen naar het herstelpad.
- ***RTT-verandering (ΔRTT):*** Deze is gerelateerd aan de extra propagatietijd die de TCP-stromen zullen ondervinden wanneer ze van het originele werkende pad naar

het herstelpad omgeschakeld worden. Voor de propagatietijden zullen we Latijnse letters gebruiken, typisch *L*. Het verschil in RTT tussen het originele en het herstelpad zal dan *ΔRTT* genoteerd worden.

Het scenario dat we simuleren om de invloed van deze parameters op het TCP-gedrag te onderzoeken zal bestaan uit een opeenvolging van drie periodes, elk vijf seconden lang, zoals aangegeven in Tabel 5.2. Tijdens de eerste periode zullen de TCP-bronnen het zenden beginnen en geleidelijk aan het netwerk vullen met verkeer. Op $t = 5s$ zal de link N1-N2 falen. Vervolgens, op $t = 5s + \delta$, zal de herstelschakeling uitgevoerd worden en zullen de getroffen stromen voortaan het herstelpad volgen. Om δ exact te regelen zullen we de schakeling als het ware handmatig instellen door de MPLS-routering op precies het gewenste tijdstip aan te passen. Bemerk dat dit betekent dat zowel de gegevens in doorgang op link N1-N2 als diegene die in N1 (of N2 voor de ACK's) aankomen in [5s, 5s + δ]. De laatste vijf seconden van het simulatie-scenario zal de link N1-N2 terug werken en het verkeer wordt teruggeschakeld naar het originele werkende pad. Elk scenario hebben we 150 keer herhaald met andere willekeurig gegenereerde waarden voor de starttijden van de TCP-bronnen en de propagatietijden van de toeganslinken.

Tabel 5.2: Gemeenschappelijke simulatiescenario.

Tijd	*Gebeurtenis*
t = 0s	Begin van de simulatie
0.5s < t < 1 s	TCP-bronnen beginnen verkeer te genereren
t = 5s	de link N1-N2 faalt
t = 5s + δ	de herstelschakeling wordt uitgevoerd
t = 10s	de faling van link N1-N2 wordt opgeheven
t = 15s	einde van de simulatie

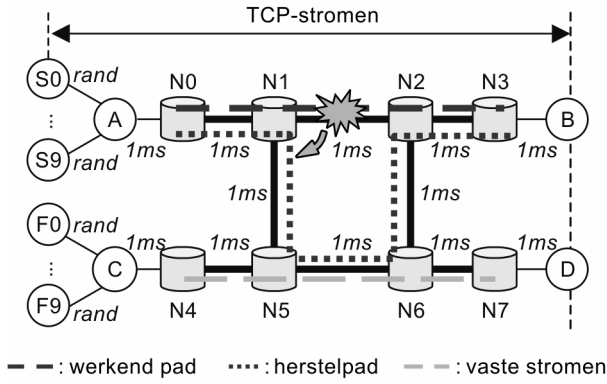
Het criterium dat we hanteren om de herstelmechanismen op hun performantie te evalueren is TCP-goodput. Dit is het aantal unieke bytes dat succesvol van bron naar bestemming werd overgebracht binnen een bepaald tijdsinterval. Het mag klaar wezen dat het deze goodput is, uitgedrukt in bv. bytes per seconde, waar de eind-gebruikers belang aan hechten. In de verder getoonde grafieken zullen we goodput relatief uitzetten tegenover de linkbandbreedte die beschikbaar is voor het aggregaat van TCP-stromen. We zullen dus het totale aantal succesvol overgebrachte bytes nemen in intervallen van duur *T* en dit delen door het maximale aantal dat we hadden kunnen ontvangen (gegeven door *T* vermenigvuldigd met de bandbreedte).

5.4.3 Invloed van de omschakelsnelheid

Het eerste aspect dat we behandelen is de impact van de snelheid waarmee na de faling wordt overgeschakeld naar het herstelpad. Het hiertoe bestudeerde scenario is geschetst in Figuur 5.5. We hebben twee sets van TCP-stromen beschouwd: (i) de omgeschakelde stromen die vertrekken uit knopen S0-S9 en die omwille van een

link-faling zullen worden omgeschakeld naar een herstelpad en (ii) een verzameling vaste stromen die ontspringen aan knopen F0-F9 en de falende link niet doorkruisen. Deze laatste zullen dus voor de volledige duur van de simulatie dezelfde route volgen. Het interval waarop we onze aandacht richten is de periode waarin de link faalt, het interval [5s,10s].

In een *pakketgeschakeld* netwerk kan zoals aangegeven een herstelpad opgezet worden zonder daarvoor bandbreedte te reserveren. In dit geval kan het zo zijn dat de omgeschakelde stromen langs hun omweg een link moeten delen met andere verkeersstromen. Op het ogenblik van de omschakeling zullen de geschakelde stromen de strijd om bandbreedte (die TCP zo volledig mogelijk tracht te vullen) aangaan. Hoe die strijd verloopt, zal afhangen van de omschakeltijd δ. Wanneer de omschakeling ogenblikkelijk plaatsgrijpt, zullen de omgeschakelde TCP-stromen nog aan een vrij hoog tempo aan het zenden zijn, omdat het verlies van de pakketten in doorgang op de falende link nog niet gedetecteerd zal zijn en TCP het zenden nog niet vertraagd zal hebben. Wanneer ze dan de knoop N5 bereiken, waar ze samenkomen met andere TCP-stromen die ook veel bandbreedte innemen, kan dit in die knoop tot verliezen leiden door een tekort aan bufferruimte om de plotse toevloed aan gegevens op te vangen. Zo kunnen heel wat segmenten verloren gaan voor beide categorieën van stromen, zowel de vaste als de omgeschakelde. Als reactie daarop zullen alle TCP-bronnen hun zend-tempo behoorlijk drastisch reduceren. Het ligt dus niet voor de hand dat de snelste omschakeltijd in termen van goodput ook de beste resultaten zal opleveren.

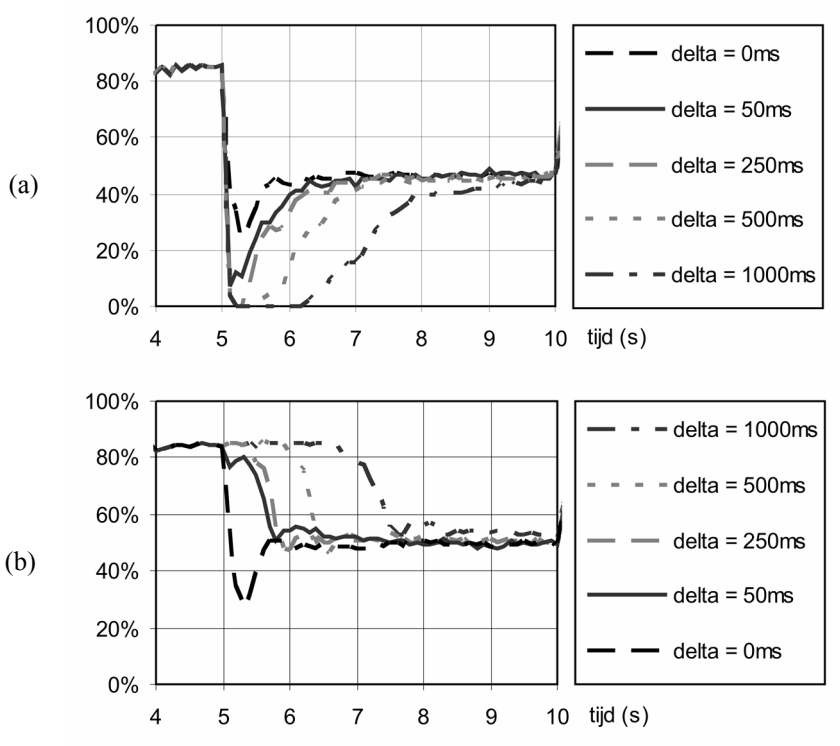


Figuur 5.5: Invloed van de omschakelsnelheid: simulatie-opstelling.

In Figuur 5.6 hebben we de evolutie van die goodput uitgezet voor zowel de vaste als omgeschakelde stromen in [4s, 10s], voor het geval van pakketschakelen zonder reservatie van bandbreedte voor herstel. Het beschreven effect van een zeer snelle omschakeltijd (δ=0) is er duidelijk uit af te lezen. Merk ook op dat het een aantal seconden kan duren eer de TCP-effecten tot een nieuwe min of meer stabiele verdeling van de beschikbare bandbreedte hebben geleid.

Om in te schatten welke δ het meest voordelig is in termen van totale goodput, vergeleken we de verhouding f(δ)=*Good*(δ)/*Good*(0), waarbij *Good*(δ) de goodput

is die bereikt wordt met omschakeltijd δ voor het geheel van omgeschakelde en vaste stromen, gedurende de eerste anderhalve seconde na het optreden van de linkfaling. Het beschouwde interval[13] van (5s,6.5s] werd gekozen omdat dit de relevante periode is voor de kleinere δ-waarden. Voor de 150 beschouwde experimenten per δ-waarde tonen we in Figuur 5.7 het histogram en gefitte normale verdelingen van de f(δ)-waarden. Uit die gegevens blijkt dat alle waarden δ>0 door de band genomen een betere goodput realiseren dan ogenblikkelijk herstel (δ=0): de gemiddelde waarde voor f(δ) ligt boven de 100%.

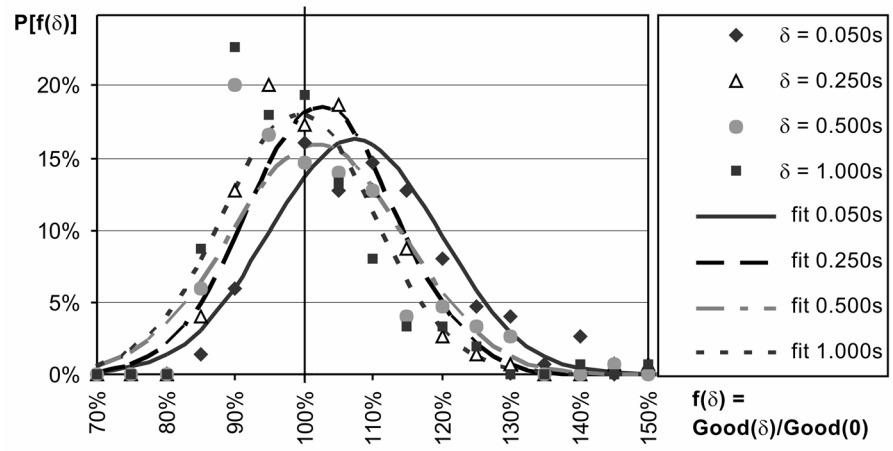


Figuur 5.6: Invloed van de omschakelsnelheid: goodput evolutie voor (a) de omgeschakelde stromen tussen S*x* en B, (b) de vaste stromen tussen F*x* en D.

Zo snel mogelijk herstellen blijkt dus niet noodzakelijk de beste oplossing: het introduceren van een kleine vertraging zorgt ervoor dat de omgeschakelde stromen hun verbruikte bandbreedte al getemperd hebben vooraleer ze de vaste stromen beginnen te vervoegen in knoop N5. We zien dan een geleidelijkere aanpassing van beide stromen aan de nieuwe netwerksituatie. Bepalen welke δ-waarde het beste

[13] Voor de grenzen van een interval zal een rond haakje gebruikt worden wanneer het een open randpunt betreft en een recht wanneer het gesloten is: (*a*,*b*] betekent dus het interval van *a* tot en met *b*, exclusief *a*, maar inclusief *b*.

netto resultaat oplevert is echter niet eenvoudig (zie Appendix I voor details). Dit zal afhangen van onder meer de linkbelasting (in het voorgestelde geval was die begrensd tot 90% door de bandbreedte op de toegangslinken, zodat de last op verbinding N5-N6 tot 180% kon oplopen), de RTT die de TCP-bronnen ervaren (grotere RTT impliceert tragere reactie op veranderingen in netwerktopologie) en het aantal TCP-stromen dat een link deelt (een groter aantal zal tot op zekere hoogte tot snellere stabilisatie leiden omdat het absolute gegevensverlies per stroom zal dalen).



Figuur 5.7: Invloed van de omschakelsnelheid: histogram van de waargenomen goodput in (5s,6.5s], relatief ten opzichte van die voor δ=0.

Vanuit een pragmatisch oogpunt kunnen we voorzichtig besluiten dat snel herstel (orde van tiental milliseconden) niet zo slecht is. Deze conclusie zal wellicht des te meer van toepassing zijn op gevallen waar linken massaal veel TCP-stromen dragen (meer dan de 100 of 200 uit het bestudeerde geval), en/of lichter belast zijn. Zo bijvoorbeeld vonden we voor toegangslinken die slechts 60% van de bandbreedte in het ruggengraatnetwerk bedroegen, dat de δ-waarde met beste goodput resultaten duidelijk verschoof richting δ=0.

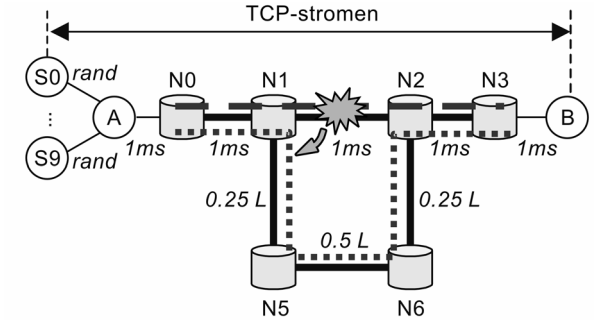
Wanneer we een *circuitgeschakeld* onderliggend netwerk beschouwen zal er geen interactie optreden tussen omgeschakelde en vaste stromen: voor elk apart zal er een circuit, i.e. een golflengte voorbehouden zijn. Het is duidelijk dat dynamische TCP-effecten, geliëerd aan het verlies van pakketten op de falende link en de veranderende RTT, wel zullen blijven optreden. In het circuitgeschakelde geval besluiten we op basis van simulaties—gelijklopend aan degene voor het pakket-geschakelde geval—dat des te sneller de omschakeling naar het herstelpad gemaakt wordt, des te betere performantie in termen van goodput. Voor een set van 140 simulaties (met verschillende waarden voor de willekeurig gegenereerde starttijden van TCP-bronnen en vertragingen van toeganslinken) voor elk van de δ-waarden uit {0, 5, 10, 20, 30, 40, 50, 250, 500, 1000ms} vonden we dat in 94% van de gevallen

δ=0 tot de beste goodput leidde. Het voordeel van circuitschakelen met betrekking tot het TCP-gedrag is duidelijk dat door reservatie van bandbreedte de interactie tussen vaste en omgeschakelde stromen vermeden wordt, waardoor TCP-stromen minder sterk verstoord worden (aangezien ze zich niet moeten aanpassen aan een gewijzigde hoeveelheid beschikbare bandbreedte). Bemerk dat men hiervoor wel een prijs betaalt in termen van te installeren netwerkcapaciteit [1].

5.4.4 Invloed van de veranderende RTT

In vorige sectie hebben we geanalyseerd wat de impact is van tragere of snellere herstel-acties, waarbij we ook aandacht besteedden aan mogelijke interacties tussen verschillende TCP-stromen. In deze sectie ligt de focus op het effect van de verandering in RTT die kan optreden wanneer het herstelpad niet dezelfde lengte heeft (in hops en/of vezellengte en dus propagatietijd). Om de invloed van die ΔRTT te isoleren zullen we hier de voornoemde interacties achterwege laten en voor het vergelijken van verschillende ΔRTT-waarden de herstelsnelheid δ constant houden.

Het beschouwde scenario wordt in Figuur 5.8 verduidelijkt. De totale propagatietijd van de overspanning N1-N5-N6-N2 werd gelijkgesteld aan L en gevariëerd over {1ms, 2ms, 10ms, 50ms, 100ms}. De falende verbinding N1-N2 heeft een vertraging van 1ms, zodat de beschouwde variaties in RTT gegeven zijn door $\Delta RTT = 2\cdot(L - 1\text{ms})$. Voor L = 100ms betekent dit quasi een verdrievoudiging van de RTT die, gelet op de waarden aangegeven in Figuur 5.8 (en Tabel 5.1), voor de stromen tussen bronknopen Sx en eindknoop B gemiddeld 120ms bedraagt. Dit is een situatie die zou kunnen optreden wanneer lokale teruglussing als herstelmechanisme wordt gebruikt bij een faling dicht bij een begin- of eindpunt van een verbinding (in het bijzonder in een netwerk met lage connectiviteit): in dat geval zal een herstelpad inderdaad beduidend langer zijn dan het originele werkende pad, vermits dit zal bestaan uit bijna het volledige werkende pad, diezelfde route in tegenovergestelde richting en een disjunct alternatief pad tussen bron- en eindknoop (herinner u Figuur 5.3).



Figuur 5.8: Invloed van veranderende RTT: simulatie-opstelling.

Een voorbeeld van de goodput-evolutie na de faling, meer bepaald in (5s,10s], is voor het geval van een omschakeltijd van δ = 5ms gegeven in Figuur 5.9. Het bevestigt onze verwachtingen op basis van de karakteristieken van TCP: een toe-

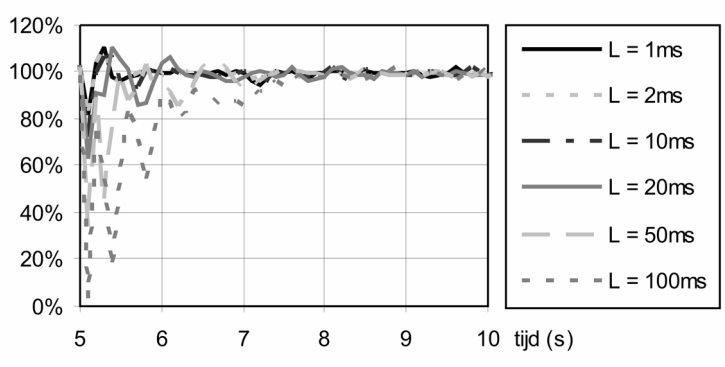
name in RTT, veroorzaakt door omschakeling naar het herstelpad, resulteert in een tijdelijke afname van de goodput. Het langere pad kan ervoor zorgen dat de timer voor heruitzenden (RTO) afgaat, zelfs al wordt snel omgeschakeld (δ = 5ms). Bovendien impliceert een langere RTT ook een trager herstel van het door de faling veroorzaakte verlies aan pakketten: de snelheid waarmee het zend-venster in TCP verhoogd wordt is gerelateerd aan de RTT, zodat de stromen meer tijd nodig hebben om de beschikbare bandbreedte opnieuw op te vullen.

Om een min of meer rigoureuze vergelijking mogelijk te maken, hebben we, net als in vorige sectie, de goodput over het interval kort na de faling beschouwd, meer bepaald tijdens (5s,6.5s]. Opnieuw hebben we het histogram van de goodput voor een gegeven *L*-waarde opgesteld, gedeeld door die voor *L* = 1ms. De vergelijking van de gevonden waarden, opnieuw over een set van 150 simulaties per waarde van *L*, staat samengevat in Tabel 5.3. Uit die numerieke gegevens kunnen we besluiten dat voor de beschouwde herstelsnelheid van δ = 5ms een langer herstelpad, dat leidt tot een toename van RTT met ongeveer 50%, wellicht te aanvaarden is: we vinden een resulterende goodput die niet meer dan 10% lager ligt dan wanneer de RTT onveranderd zou blijven.

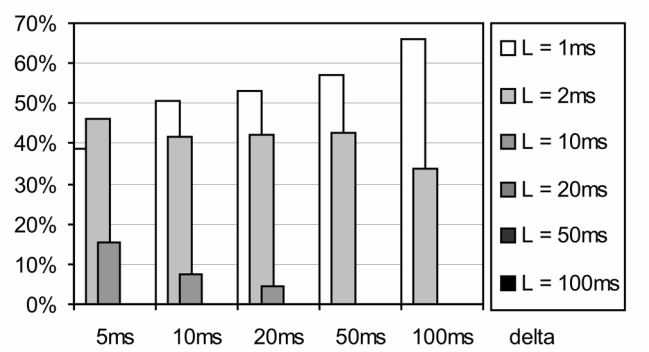
De vraag rijst uiteraard of we geen grotere verschillen in RTT kunnen accepteren voor tragere herstelmechanismen. Om daarop te antwoorden hebben we onze experimenten herhaald voor δ-waarden genomen uit {5ms, 10ms, 20ms, 100ms}. Wanneer de tijd δ tussen faling en omschakeling naar het herstelpad voldoende groot wordt, verwachten we dat de goodput tot nul terugvalt, ongeacht het RTT-verschil. Wat zoals eerder uitgelegd wél altijd zal afhangen van die RTT is de snelheid waarmee de bandbreedte opnieuw geleidelijk aan wordt gevuld (i.e. de aangroei van goodput richting 100%). De evolutie van de goodput (met curves gelijkaardig aan Figuur 5.9, zie Appendix I) heeft deze kwalitatieve voorspellingen bevestigd: voor grotere waarden van *L* vinden we een trager herstel van het goodput-niveau.

Tabel 5.3: Invloed van veranderende RTT: vergelijking van de goodput in het interval (5s,6.5s].

L	*ΔRTT*	*relatief verschil in goodput tegenover L=1ms*	*fractie van de resultaten waar L lagere goodput realiseert dan L=1ms*	*fractie van de resultaten waar L tot de hoogste goodput leidt*
1 ms	0.00%	0.00%	0.00%	38.67%
2 ms	+1.67%	+0.13%	42.00%	46.00%
10 ms	+15.00%	–1.03%	75.33%	15.33%
20 ms	+31.67%	–3.13%	94.00%	0.00%
50 ms	+81.67%	–13.66%	100.00%	0.00%
100 ms	+165.00%	–31.48%	100.00%	0.00%



Figuur 5.9: Invloed van veranderende RTT: goodput evolutie voor δ=5ms.



Figuur 5.10: Invloed van veranderende RTT: fractie van de simulatieresultaten waar een bepaalde L-waarde tot de hoogste goodput leidde in (5s,6.5s].

Om een accurate vergelijking te maken tussen de onderscheiden veranderingen in RTT (evenredig zijnde met L), hebben we ook hier de behaalde goodput berekend over het interval[14] (5s,6.5s]. Figuur 5.10 toont aan dat wanneer de omschakeltijd δ verhoogt, de "beste" L-waarde verschuift naar lagere waarden. Dit wijst erop dat hoe langer we wachten om naar het herstelpad om te schakelen, met hoe meer zekerheid we kunnen stellen dat een kleinere verandering in RTT tot betere goodput zal leiden. Een analyse van de goodput gegevens bevestigt dit (zie Tabel I.5 in Appendix I), alsook Figuur 5.10. Figuur I.14 uit de appendix leert trouwens ook dat een verhoging van de padlengte (en dus RTT) met L – 1ms ongeveer even nefast is voor de goodput als een verhoging van de omschakeltijd δ van dezelfde grootteorde (maar met behoud van RTT): vergelijk bijvoorbeeld de goodput-waarden voor (L, δ) = (10ms, 100ms) en (100ms, 10ms). Dit suggereert dat het niet eenvoudig is te

[14] Bemerk dat dit voor sommige situaties korter is dan de tijd nodig om opnieuw tot een stabiel goodput niveau te komen, maar een langer interval verandert de relatieve posities voor de verschillende L-waarden niet.

beslissen wat beter is: een snelle omschakeling maken naar een langer herstelpad (denken we bv. aan teruglussen), of eerder een tragere omschakeling naar een pad van vergelijkbare lengte (wat o.a. bij padprotectie zou kunnen gebeuren).

5.4.5 Invloed van terugschakelen naar werkend pad

De vorige analyses spitsten zich toe op wat er gebeurt wanneer een faling optreedt. Wanneer evenwel de faling wordt opgelost (bv. herstel van gebroken vezels), zal in het geval van omkerende herstelmechanismen het verkeer teruggeschakeld worden naar het originele pad. Hier zullen we de effecten van dit terugschakelen onderzoeken in twee gevallen, kortweg Pakket en Circuit genoemd, en hun performantie in termen van goodput vergelijken met een niet-omkerend mechanisme dat ook na het opheffen van de faling de stromen het herstelpad laat volgen. Het geval Pakket verwijst naar pakketschakelen, waar het samenvloeien van herstel- en werkend pad geen probleem vormt, terwijl het Circuit-geval een circuitschakelende laag onderstelt die niet toelaat dat in een schakelaar twee inkomende circuits worden samengenomen in één uitgaand circuit.

De beschouwde topologie is dezelfde als in voorgaande secties. Waar we voorheen het gebeuren tijdens de faling onderzochten, kijken we nu wat er in [10s,15s] gebeurt eenmaal de falende link hersteld is. De mogelijke acties die daar ondernomen worden zijn:

- ***Pakket:*** De routes die gevolgd worden door de pakketten worden teruggeschakeld naar het originele werkende pad op tijdstip t = 10s. In dit pakketgeschakelde scenario zal de resterende stroom pakketten, die nog onderweg zijn langs het (langere) herstelpad, in knoop N2 samenvloeien met pakketten die opnieuw de link N1-N2 volgen. Hetzelfde gebeurt in knoop N1 voor de ACK-pakketten die in de omgekeerde richting bewegen.
- ***Circuit:*** Hier is samenvloeien van de stroom langs het opnieuw werkende pad en de resterende pakketten langs het herstelpad niet mogelijk. Zodra het eerste pakket de gerepareerde link doorkruist en in knoop N2 aankomt, zullen pakketten die nog langs het herstelpad van N5 komen, verloren gaan. Hetzelfde geldt, mutatis mutandis, voor de ACK's in N1.
- ***GeenActie:*** In dit geval blijven pakketten na het opnieuw functioneren van link N1-N2 het herstelpad volgen.

De enige parameter die hier nog van belang is, is het verschil in padlengte (dus RTT). Net als in voorgaande sectie hebben we om het effect ervan te bestuderen L gevariëerd over {1ms, 2ms, 10ms, 20ms, 50ms, 100ms}.

Vooraleer de resultaten in meer detail te bekijken, kunnen we al een kwalitatieve voorspelling maken van de verschillen die zullen optreden:

- ***Pakket:*** Na de faling krijgt knoop N2 twee pakketstromen te verwerken, die samen dezelfde uitgaande link naar N3 moeten volgen. Wanneer L groot is betekent dit dat de duur van dit samenvloeien relatief lang kan worden (want er zijn dan nog veel pakketten onderweg langs het alternatieve pad, dat $L - 1$ms langer is). In zo'n geval verwachten we dus dat verliezen kunnen optreden door gebrek aan bufferruimte in N2 om de tijdelijke overbelasting op te vangen. Een tweede effect, dat ook optreedt voor kleinere L-waarden, is het feit dat pakketten

niet langer in de correcte volgorde worden afgeleverd op de bestemming B. Wanneer een pakket langs het opnieuw werkende pad een voorganger (nog onderweg langs het herstelpad) voorbijgestoken heeft, zal TCP reageren als was er pakketverlies opgetreden en de verloren gewaande pakketten opnieuw verzenden. We verwachten dan ook voor lage *L*-waarden een zekere terugval in goodput.

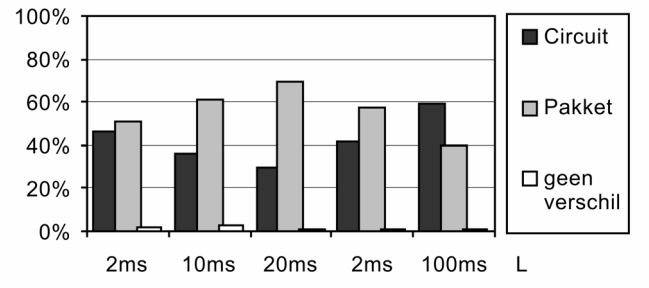
- ***Circuit:*** In dit geval zullen steeds pakketverliezen optreden zodra het herstelpad in lengte verschilt van het originele: eenmaal de faling opgelost is, gaan de pakketten die nog onderweg zijn op het herstelpad verloren. Hoe groter *L*, des te meer pakketten er verloren gaan. Opnieuw verwachten we daarom een terugval in goodput. Maar in tegenstelling tot het pakketgeschakelde concept zullen pakketten niet in een verkeerde volgorde bij de bestemming toekomen, waardoor de terugval mogelijks ietwat sneller kan worden goedgemaakt.
- ***GeenActie:*** Vermits er hier niets verandert aan de netwerksituatie zoals de TCP-stromen die ervaren, zal de goodput hetzelfde niveau behouden.

Uit deze korte analyse blijkt dat GeenActie de beste goodput resultaten zal kunnen voorleggen, maar of bij een terugschakeling naar het originele pad een Circuit-geschakeld dan wel een Pakketgeschakeld concept beter is, valt moeilijk a priori in te schatten.

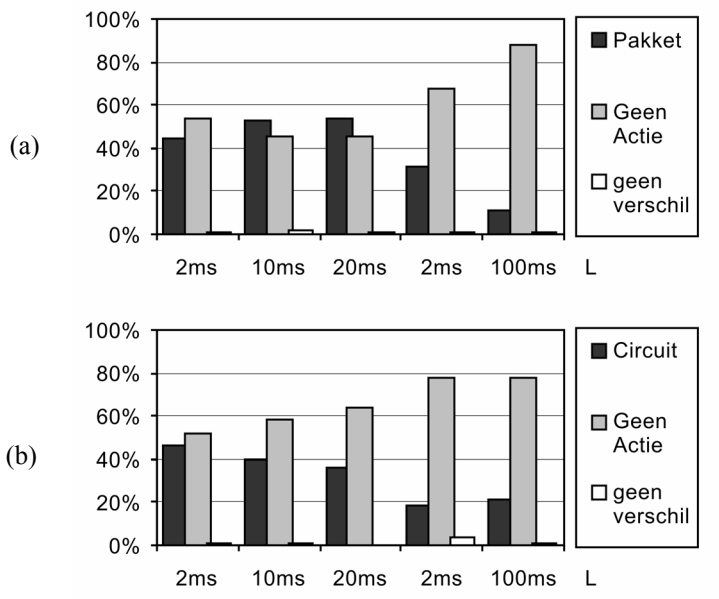
Een studie van de goodput-evolutie in het relevante interval na t = 10s leert dat de drie beschouwde gevallen zich voor kleine RTT-verschillen (*L* = 1ms, 2ms) nauwelijks van elkaar onderscheiden in termen van TCP-goodput: voor het Pakket-geschakelde geval zijn er geen verliezen door het samenvloeien en slechts een uiterst beperkt aantal herordeningen van pakketten; voor Circuitschakelen is het aantal verloren pakketten miniem. Maar wanneer we *L* verhogen tot 10 à 20ms vinden we wel een duidelijk waarneembaar verschil tussen Pakket- en Circuitschakelen. Het Pakketgeschakeld geval vertoont een kleine terugval in goodput (ca. –5%) doordat een beperkt aantal TCP-connecties terugvalt naar zogenaamd snel heruitzenden/snel herstel (zie hoger, Sectie 5.2). Omdat niet pakketverlies, maar wel een herordening daarvoor aan de basis ligt, kan het TCP-mechanisme zeer snel het oorspronkelijke goodput-niveau herstellen. Voor circuitschakelen daarentegen treden zulke verliezen wel op en observeren we een grotere terugval in goodput. Voor voldoend grote waarden van *L* zullen zoals gezegd verliezen optreden zelfs in het pakket-geschakelde geval en er de goodput-degradatie, veroorzaakt door het uit volgorde afleveren van pakketten, versterken. Voor het bestudeerde geval *L* = 100ms bij-voorbeeld is het gezamenlijke effect van beide fenomenen dat het globale goodput-niveau slechter is dan voor Circuitschakelen.

Om een meer nauwkeurige vergelijking te maken hebben we alweer de goodput over een bescheiden interval (10s, 11.5s] na de schakelactie beschouwd. Figuur 5.11 toont in hoeveel van de 150 beschouwde simulaties per *L*-waarde elk van de onder-scheiden strategieën (Pakket, Circuit) de hoogste goodput realiseert. Op basis van deze goodput-gegevens de keuze te maken tussen een Pakket- en een Circuit-geschakelde aanpak is niet zo evident. De resultaten wijzen uit dat voor beperkte verschillen in RTT het Pakketschakelen beter presteert, terwijl voor grotere verschillen het Circuitschakelen hoger scoort. Het exacte punt waar Pakketschakelen slechter wordt dan Circuit-schakelen zal in het algemeen afhangen van de

hoeveelheid bufferruimte[15] in de schakelknopen, in het bijzonder die waar herstelpad en origineel pad samenvloeien.



Figuur 5.11: Invloed van terugschakelen na opheffing van de faling: circuit- vs pakketschakelen. We zetten de fractie uit van de simulatieresultaten waar een bepaalde terugschakel-strategie de hoogste goodput opleverde in (10s,11.5s].



Figuur 5.12: Invloed van terugschakelen na opheffing van de faling: al dan niet terugschakelen? We zetten de fractie uit van de simulatieresultaten waarbij terugschakelen of het onverandered laten van de routes (GeenActie) de hoogste goodput opleverde tijdens (10s,11.5s], in (a) voor pakketschakelen en (b) voor circuitschakelen.

[15] In de simulaties werd een output-buffer gebruikt met SFQ-strategie, ter grootte van 50 pakketten.

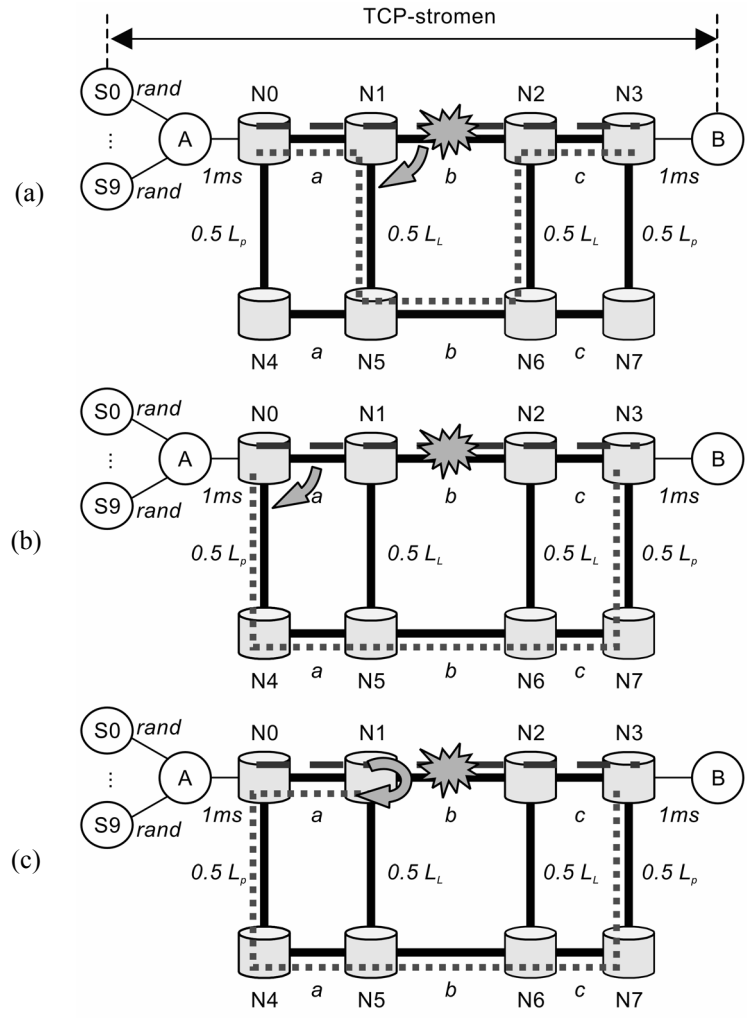
In Figuur 5.12 beogen we de terugval in goodput, veroorzaakt door de herstelacties, in te schatten. We hebben hiervoor voor Pakket- en Circuitschakelen apart gekeken in hoeveel van de simulatiegevallen het beter was Geen Actie te ondernemen bij herstel van het primaire pad. Voor kleine *L*-waarden is er geen overtuigende winnaar aan te duiden en vinden we dat er nauwelijks enige terugval in goodput is. Voor grote *L* is de trend duidelijk: GeenActie is ontegensprekeijk beter. We kunnen besluiten dat alleen wanneer het verschil in padlengte groot wordt er een noemenswaardige performantiedaling is, in vergelijking met het onaangeroerd laten van de gevolgde route.

5.4.6 Gevalsstudie: MPLS-protectiemechanismen

Met voorgaande simulatiestudies hebben we inzicht verworven in de invloeden van omschakelsnelheden en veranderende RTT op het gedrag van TCP-stromen. In de nu volgende gevalsstudie zullen we de vergaarde kennis toepassen op de belangrijkste protectiemechanismen in GMPLS voor een pakketschakelende technologie. De topologie die we zullen beschouwen is erg gelijkaardig aan voorgaande situaties en wordt geschetst in Figuur 5.13. We zullen in totaal vier scenario's beschouwen, overeenkomend met de drie protectiemechanismen uitgelegd in Sectie 5.3 en een variant van het teruglussen. De onderscheiden herstelacties, die gevallen na het falen van de link tussen N1 en N2 ondernomen zullen worden, zijn:

- ***Lokale protectie (Lokaal):*** Hier wordt lokale linkprotectie toegepast. Na het falen van de link, wat opnieuw na vijf seconden gebeurt, zal zoals aangegeven in het bovenste deel van Figuur 5.13 op $t = 5s + \delta$ een omschakeling gebeuren in knoop N1 naar het herstelpad dat L_L langer is dan het originele. Na het opheffen van de faling op t=10s zullen pakketten door schakelaar N1 (resp. N2) pakketten opnieuw langs het originele pad gestuurd worden en zo in N2 (resp. N1) samenvloeien met de pakketten die nog langs het herstelpad vertrokken waren.
- ***Padprotectie (Pad):*** Wanneer we padprotectie toepassen is signalisatie nodig om knoop N0 van het falen van verbinding N1-N2 op de hoogte te brengen. Hierdoor zal de omschakeling, nu in N0 (resp. N3), op een later tijdstip gebeuren dan voor Lokale protectie, met name op $t = 5s + \delta + a$ (resp. $t = 5s + \delta + c$). Opnieuw zal het herstelpad langer zijn dan het originele, met nu een verhoging ter grootte van L_P en zal, na opheffen van de faling, verkeer langsheen het originele pad en het herstelpad samenvloeien, nu in knoop N3 (resp. N0). Ook die terugschakeling naar de oorspronkelijke route vereist signalisatie, maar dat de schakelactie later optreedt dan voor bv. lokale protectie is hier minder erg: verkeer dat even na t=10s nog het herstelpad volgt gaat immers niet verloren, in tegenstelling tot verkeer dat we even na t=5s nog langs het werkende pad sturen.
- ***Lokale teruglussing (Teruglus):*** Dit is het Lokaal Teruglussen zoals origineel voorgesteld in [227] en eerder beschreven in Sectie 5.3. Hier is, zoals in lokale protectie, geen signalisatie vereist en kan de omschakeling naar het herstelpad gebeuren op $t = 5s + \delta$, in een knoop die een eindpunt vormt van de falende link. Het herstelpad van N0 naar N3 is nu $L_P + 2 \cdot a$ langer dan het originele; de terugweg (gevolgd door de ACK's) is $L_P + 2 \cdot c$ langer. Nadat de faling is opgeheven, schakelen N1 en N2 opnieuw over op de originele route.

- ***Variant op lokale teruglussing (TeruglusVar):*** Het nadeel van lokaal teruglussen is dat het verkeer nodeloos een rondje maakt tussen knopen N0 en N1 (tussen N2 en N3 voor de ACK's). In deze variant van teruglussen zullen we het originele teruglussen mengen met padprotectie: op $t = 5s + \delta$ schakelen we over naar het terugluspad en zendt schakelaar N1 een controlesignaal naar knoop N0 (ook N2 naar N3). Bij ontvangst ervan op $t = 5s + \delta + a$ schakelt N0 pakketten niet meer langs het onnodige lusje, maar onmiddellijk naar N4 zoals in padprotectie (idem voor N3 en N7).



Figuur 5.13: Gevalsstudie van MPLS-protectiemechanismen: schets van de simulatie-opstelling: (a) lokale protectie, (b) padprotectie, (c) lokale teruglussing. Merk op dat voor de eenvoud enkel de routes lopend van N0 naar N3 getoond zijn en niet die in omgekeerde richting.

Tabel 5.4: Gevalsstudie van MPLS protectiemechanismen: vier netwerkscenario's en de corresponderende parameters.

Scenario	L_L	L_P	a	b	c
I	1ms	3ms	2ms	1ms	48ms
II	5ms	15ms	10ms	1ms	40ms
III	10ms	30ms	20ms	1ms	30ms
IV	20ms	60ms	40ms	1ms	10ms

Tabel 5.5: Gevalsstudie van MPLS-protectiemechanismen: de tijdsparameters die bepalend zijn voor het TCP-gedrag.

Protectie-mechanisme	*Omschakeltijd*	*ΔRTT*
Lokaal	δ	$2 \cdot L_L$
Padprotectie	δ + a	$2 \cdot L_P$
Teruglussen	δ	$2 \cdot L_P + 2 \cdot a + 2 \cdot c$

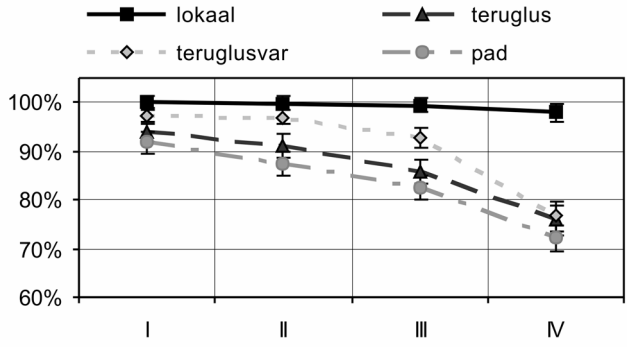
Voor elk van deze vier scenario's hebben we simulaties uitgevoerd voor vier sets topologie-parameters, zoals opgesomd in Tabel 5.4. Met $a + c$ = 50ms en b = 1ms leidt dit tot een gemiddelde RTT van 216ms. De overige parameterwaarden gaven we dezelfde waarden als besproken in 5.4.2.

Uit de gebruikte parameterwaarden en het discours gevoerd in voorgaande secties, kunnen we meteen afleiden dat lokale protectie het best zal scoren in termen van goodput onmiddellijk na de faling: de omschakeltijd naar het herstelpad is het kortst (namelijk δ) en ook het verschil in RTT is het kleinst. De waarden van deze twee tijdsparameters voor de onderscheiden scenario's staan samengevat in Tabel 5.5. Voor padprotectie is de toename in RTT hoger dan lokale protectie. Daarenboven zal de omschakeling ook langer op zich laten wachten, met hoger pakketverlies tot gevolg: beide effecten zullen tot een reductie van goodput leiden kort na de faling. Lokale teruglussing vermijdt de relatief trage omschakeltijd, maar verhoogt de toename in RTT tegenover padprotectie. Aangezien we in 5.4.4 hebben vastgesteld dat een stijging van RTT even schadelijk is voor de goodput als een verhoging van de omschakeltijd van vergelijkbare grootteorde, verwachten we dat het netto voordeel van lokale teruglussing tegenover Padprotectie niet aanzienlijk zal zijn. De variant die de schakelsnelheid behoudt, maar de onnodige lus in de routering nadien verwijdert, zou hierdoor een hogere goodput kunnen realiseren. Maar de omschakeling van de route met lus naar die zonder heeft een gelijkaardig effect als het in 5.4.5 beschouwde omschakelen van herstel- naar werkend pad: een plotse afname van de RTT en het niet in volgorde afleveren van de pakketten. Of er ook een netto voordeel zal zijn is dus zeer de vraag.

De bovenstaande kwalitatieve analyse wordt bevestigd door de geobserveerde goodput-evolutie, waarvoor we verwijzen naar Appendix I. Een vergelijking van de

bereikte goodput over het interval (5s,7.5s], waarin de verschillende mechanismen duidelijk verschillen in goodput gedrag, is samengevat in Figuur 5.14. We kunnen eruit besluiten dat Lokale Protectie tot de beste goodput-resultaten leidt voor de beschouwde netwerkparameters. Omdat kiezen tussen Padprotectie en Lokale Teruglussing neerkomt op het inruilen van langere omschakeltijd voor grotere RTT-toename, vinden we dat ze niet al te veel van elkaar verschillen qua netto goodput. De variant van het lokaal teruglussen blijkt enkel duidelijk voordeel op te leveren wanneer de lengte van de lus niet al te groot is (scenario's I, II en III).

Voor het gedrag na de faling, met name wanneer terug overgeschakeld wordt naar het originele pad, is zoals uiteengezet in 5.4.5 enkel het verschil in RTT van belang. We vinden dan ook (zie Appendix I) dat lokale protectie het beste goodput-resultaat oplevert, gevolgd door padprotectie en pas dan lokale teruglussing. De variant van teruglussing valt in de periode na het opheffen van de faling uiteraard samen met padprotectie.



Figuur 5.14: Gevalsstudie van MPLS-protectiemechanismen: goodput in (5s,7.5s] voor elk van de verschillende scenario's.

5.5 Besluiten

In dit hoofdstuk zijn we opgeklommen tot het niveau van de gebruikers van een (al dan niet optisch pakketgeschakeld) netwerk, meer bepaald de transportlaag TCP, om er de invloed van plotse omschakelacties in de routering naar aanleiding van netwerkfalingen te bestuderen. Voor zover we hebben kunnen vaststellen uit het doorsnuisteren van de literatuur, mag dit gerust als een originele bijdrage bestempeld worden[16].

We hebben in onze studie de interactie beschouwd tussen omgeschakelde TCP-stromen en andere die al aanwezig waren in (delen van) de herstelroute en de impact van de omschakeltijd daarop geëvalueerd. Uit die analyse leidden we af dat het tot

[16] Wat we ook wel kunnen afleiden uit het geciteerd worden in de rubriek "Scanning the literature" van Optical Networks Magazine in het nummer van november–december 2002.

het uiterste opdrijven van de herstelsnelheid (tot ruim onder de 50ms) niet noodzakelijk voordelig is: vanuit het standpunt van TCP-goodput kan een iets langere vertraging betere resultaten opleveren.

Verder hebben we ook het effect van het verschil in RTT tussen werkend en herstelpad bestudeerd voor verschillende omschakelsnelheden. Niet algeheel tot onze verrassing hebben we geconstateerd dat die verhoging in RTT een negatieve invloed heeft op de goodput en des te sterker voor grotere RTT-toename. Onze resultaten toonden ook dat geen duidelijk uitsluitsel kan worden gegeven over de vraag of het beter is snel naar een duidelijk langer herstelpad om te schakelen dan trager naar een pad van ongeveer dezelfde lengte als het origineel werkende: de verschillen in goodput zijn klein en hangen af van de precieze tijds- en netwerk-parameters.

Ook hebben we de reactie van TCP onderzocht op een terugschakeling naar het werkende pad eenmaal de faling opgelost is. Hierbij hebben we gewezen op het verschillende gedrag voor circuit- en pakketschakelende technologieën en hun uiteenlopende performantie in termen van goodput. Het feit dat in circuitschakelen stromen niet kunnen samenvloeien—wat wenselijk zou zijn waar herstelpad en een origineel werkend pad samenkomen—blijkt enkel in mindere prestaties te resulteren voor erg kleine verschillen in padlengte (klein ten opzichte van de aanwezige buffercapaciteit voor pakketschakelen). In de andere gevallen vinden we evenwel geen sterke discrepanties in bereikte goodput voor circuit- of pakketschakelen. Vanzelfsprekend heeft een circuitgeschakeld concept wel andere nadelen, zoals het feit dat meer extra bandbreedte moet worden voorzien om het netwerk resistent te maken tegen falingen. Voor een gedetailleerde studie daarvan verwijzen we naar [222].

We hebben het hoofdstuk afgerond met een gevalsstudie omtrent de invloed van herstelacties op TCP-gedrag voor vier protectiemechanismen zoals voorgesteld in het GMPLS-raamwerk. Die analyse leerde dat lokale protectie beter presteert in termen van TCP-goodput. Lokale teruglussing levert niet veel voordeel op tegenover padprotectie: een kleine winst aan goodput bij de omschakeling na de faling, maar licht verlies bij terugschakeling eenmaal de faling opgelost is. Het vermijden van de lus in het herstelpad bij lokale teruglussing levert in termen van goodput enkel voordeel op wanneer die lus betrekkelijk kort is in vergelijking met de totale padlengte. Opnieuw dienen we hier de kanttekening te maken dat deze rangschikking in termen van TCP-performantie genuanceerd moet worden met andere criteria, zoals in [1], waar we de duurte—in termen van vereiste band-breedte—aantonen van lokale protectie in vergelijking met andere herstel-mechanismen.

Hoofdstuk 6
Samenvattend besluit

Het opzet van dit werk was een zo breed mogelijk spectrum te belichten van de conceptuele vraagstukken die optreden bij het beschouwen van optisch pakketgeschakelde netwerken. We zijn begonnen met het scheppen van een referentiekader op basis van een literatuurstudie, om zo tot een overzicht te komen van wat een optisch pakketgeschakeld netwerk precies is, welke keuzes gemaakt kunnen worden op vlak van technologie, pakketformaat en architectuur van de schakelknopen.

Qua architecturen hebben we vastgesteld dat er typisch ingespeeld wordt op de geografische schaal van het netwerk. Zo wordt voor regionale, zogenaamde metronetwerken in veel gevallen geopteerd voor een ringnetwerk. Onze bijdrage heeft twee varianten van schakelarchitecturen voor zo'n metro-omgeving met elkaar vergeleken en geëvalueerd in termen van de hoeveelheid apparatuur (componenten) die moet worden geïnstalleerd om te voldoen aan een gegeven vraag naar bandbreedte binnen een verzameling verkeersknooppunten, waar de optische pakketschakelaars gelokaliseerd zijn. We bespraken een zogenaamde passieve knooparchitectuur, die omwille van zijn zeer eenvoudige structuur op basis van vandaag commercieel beschikbare technologieën zou kunnen worden geconstrueerd. We hebben getoond dat die nauwelijks moet onderdoen voor een complexere actieve architectuur: in termen van te installeren zend- en ontvangstcapaciteit brengt de passieve architectuur geen meerkost met zich mee. Wel is het zo dat een actieve architectuur minder golflengten nodig heeft, vooral wanneer er voldoende vraag is naar bandbreedte voor verbindingen die binnen één metro-ring blijven en deze ruimtelijk niet al te veel overlappen. Dit laatste geval vereist wel dat het controleprotocol dat de toegang tot het metro-netwerk regelt zogenaamd ruimtelijk hergebruik (Eng. spatial reuse) toelaat. Daarnaast hebben we ook aandacht besteed aan het al dan niet voordelig zijn van bidirectionele ringen en een zogenaamd golflengtebandconcept.

Voor schakelarchitecturen voor vermaasde ruggengraatnetwerken hebben we aandacht besteed aan het schalen van schakelmatrices naar grote dimensies. Geïnspireerd op het baanbrekende werk van Clos hebben we geïllustreerd dat hiervoor een architectuur met meerdere stadia zijn nut kan bewijzen. Specifiek voor optische pakketschakelaars hebben we aangetoond dat het derde schakelstadium uit Clos' originele werk geëlimineerd kan worden en vervangen kan worden door golflengte-conversie. Voor een "broadcast-and-select"-schakelmatrix gebaseerd op SOA-technologie hebben we in een gevalsstudie aangetoond dat dit een sterkere

reductie oplevert van het aantal vereiste schakelende SOA's dan een Clos-gebaseerde matrix met drie volwaardige schakelstadia.

We hebben onze studie vervolgd met onderzoek naar schakel-algoritmen voor een optische pakketschakelaar met een gedeelde recirculerende buffer opgebouwd uit vertragingslijnen. Voor synchroon, slot-gebaseerd schakelen van pakketten van vaste lengte hebben we de Balans-strategie ontworpen die de aanwezige bufferuimte zo efficiënt mogelijk benut. Die nieuwe strategie slaagt erin duidelijk lager pakketverlies te bereiken voor dezelfde hoeveelheid vertagingslijnen.

Voor het schakelprobleem zonder differentiatie vonden we in de literatuur duchtig uitgekiende oplossingen. In geval van pakketten van variabele lengte hebben we daarom bijzondere aandacht besteed aan het realiseren van kwaliteitsdifferentiatie. Waar voor vaste pakketlengte een eenvoudige aanduiding van prioriteiten in pakketten volstaat (zoals aangetoond met eigen werk), moeten voor variabele lengte bijkomende technieken gebruikt worden. De uit OBS bekende strategie van offset-gebaseerde differentiatie—waar men de pakkethoofding van prioritair verkeer eerder voorop stuurt—werd vergeleken met twee zelf bedachte strategieën: vooruitkijken en slot-gebaseerde controle. Hoewel deze laatste techniek er ook wel in slaagt differentiatie te realiseren, is ze beduidend minder uitgesproken dan de overige mechanismen. Het voordeel ervan is dat, door het hanteren van een slot-concept, de complexiteit van de controle-eenheid gereduceerd kan worden. Het vooruitkijken bleek op vlak van pakketverlies even goed of een tikkeltje beter te scoren dan offset-gebaseerde differentiatie, maar zorgt voor sterkere discriminatie van langere pakketten. De klassieke offset-differentiatie blijkt dus moeilijk te overtreffen.

Ander vernieuwend werk werd geleverd door het vergelijken van de performantie van twee mechanismen om in slot-gebaseerde schakelmatrices toch pakketten van variabele lengte te behandelen. Een trein-concept, waarbij alle slots die samen één pakket vormen steeds als een geheel worden geschakeld, vergeleken we met een zogenaamd wagon-concept dat voor elk van die slots onafhankelijk beslissingen neemt. Het trein-concept kwam hierbij als winnaar uit de bus: de performantie in termen van gegevensverlies ligt in de meerderheid van de gevallen lager. Enkel voor gemiddelde belasting en wanneer de pakketlengte gemeten in slots beperkt blijft, kan een wagon-concept voordeel opleveren doordat de bufferruimte dan efficiënter benut wordt. Wel blijkt een wagon-concept erin te slagen behoorlijk sterke kwaliteitsdifferentiatie te realiseren louter op basis van prioriteitsaanduidingen in de pakkethoofding, waar dit voor het trein-concept slechts in beperkte differentiatie resulteert.

Ook op netwerkniveau hebben we een onderzoeksbijdrage geleverd: we hebben een routeringsalgoritme bedacht om het pakketverlies (PLR) te minimaliseren. Daartoe gebruikten we een vrij nauwkeurige schatting van het pakketverlies in de optische pakketschakelaars met gedeelde recirculerende buffer. Routering op basis van de schatting van het verlies bleek uiteraard beduidend lagere PLR te bereiken dan pure kortste-pad routering. Belangrijker in deze is dat ook ten opzichte van een algoritme dat zich enkel baseert op het spreiden van de linkbelasting (Eng. load balancing) duidelijke voordelen geboekt werden.

Nog dichter bij de gebruiker die zijn verkeer over een (al dan niet optisch pakketgeschakeld) netwerk stuurt, situeert zich onze uitgebreide studie van TCP-gedrag. Hier hebben we de invloed bestudeerd van het omschakelen van de routes gevolgd door een aggregaat van zulke TCP-stromen (die het gros van het IP-verkeer vormen, dat tegenwoordig qua volume het meer traditionele verkeer zoals klassieke telefonie overtreft). In het bijzonder hebben we de route-veranderingen bestudeerd die optreden naar aanleiding van falingen. Het criterium dat we hanteerden was de zogenaamde goodput, zijnde de overdracht van unieke gegevens tijdens een bepaalde periode. We hebben de individuele invloeden van de omschakelsnelheid naar een herstelpad geanalyseerd, alsook de toename in lengte van de gevolgde route.

In een pakketgeschakelde netwerktechnologie kan interactie ontstaan tussen omgeschakelde TCP-stromen en andere die al aanwezig zijn langsheen (delen van) het herstelpad. Er blijkt dat in zo'n geval het tot een absoluut minimum herleiden van de omschakelsnelheid niet noodzakelijk beter is. Wanneer die interactie niet kan optreden, blijken de circuit- en pakketgeschakelde technologieën vergelijkbare goodput-prestaties te leveren. Enkel bij het terugschakelen naar het originele werkende pad na opheffen van de faling is er dan onderscheid, waarbij het afhangt van de verandering in padlengte of het pleit wordt beslecht in het voordeel van circuit- dan wel pakketschakelen. Voor zeer kleine verschillen lijkt pakketschakelen (met voldoende buffer) iets beter te scoren, terwijl circuit-schakelen doorgaans hogere goodput bereikt voor grotere afwijkingen tussen de lengte van werkend- en herstelpad.

Met een gevalsstudie voor de bekende GMPLS-protectiemechanismen toetsten we onze besluiten. De hoogste goodput-resultaten werden er behaald met lokale linkprotectie. Lokale teruglussing presteerde niet veel beter dan padprotectie: we vonden een klein voordeel qua goodput bij omschakeling naar het herstelpad, maar een licht verlies bij terugschakelen na het opheffen van de faling. De lus wegwerken die eigen is aan—jawel, u raadt het al—lokale teruglussing blijkt niet echt de moeite te lonen: enkel wanneer die lus betrekkelijk kort is met betrekking tot de totale padlengte, hebben we een verbetering van de goodput vastgesteld tegenover het laten bestaan van het nodeloze rondje.

Bij deze TCP-studies hebben we erop gewezen dat de evaluatie van circuit- versus pakketschakelen, en ook die van de GMPLS-protectiemechanismen, moet worden gebalanceerd met studies van de bandbreedtevereisten zoals uiteengezet in [1,222]: circuitschakelen vereist meer capaciteit om het netwerk robuust te maken tegen mogelijke falingen dan pakketschakelen. Lokale protectie blijkt daar de slechtst scorende stategie te zijn (tegenover de betere op vlak van TCP-goodput).

We mogen besluiten dat de collectie van deze studies heel wat vragen oplost die rijzen bij het nadenken over optisch pakketschakelen. Maar even duidelijk is het dat er minstens nog even zoveel niet beantwoord werden. Voor de knoop-architectuur hebben we bijvoorbeeld geen gedetailleerde analyse gemaakt van de technologische haalbaarheid en het nagaan van bijvoorbeeld fysische beperkingen van de voorgestelde Clos-architectuur op basis van SOA's. Maar daarvoor zullen we

in aansluitend onderzoek beroep doen op de expertise van collegae[17] uit Athene. Wat de realisatie van de bestudeerde metro-architectuur betreft, zijn intussen in de literatuur al onderzoeksresultaten te vinden [93,118]. Omtrent de AWG-gebaseerde schakelmatrix, die we hier slechts uiterst beknopt bespraken, zal u ongetwijfeld veel meer vernemen in het nakende doctoraatsproefschrift van ir. J. Cheyns.

Meer geavanceerde routeringsalgoritmes dan degene die we in Hoofdstuk 4 bedachten zal u kunnen vinden in de doctoraatsthesis van lic. S. Vandenberghe, in het bijzonder voor IP/MPLS netwerken met kwaliteitsdifferentiatie. Voor de routering in meerlaagse netwerken en in het bijzonder het zogenaamde "grooming"-probleem verwijzen we u naar de recente proefschriften [229,230].

Een aspect dat we in dit proefschrift bewust niet hebben behandeld is het netwerkbeheer. Maar daarin onderscheidt een OPS-netwerk zich niet wezenlijk van actuele netwerktechnologieën, waaraan in onze vakgroep al meer dan één volledig doctoraat werd gewijd [231,232]. Ook de controle-aspecten eigen aan optische netwerken genoten binnen INTEC onderzoeksaandacht (bv. het vierde hoofdstuk in [230]).

[17] Met wie we trouwens al een uitbreiding van de hier voorgestelde resultaten voorbereid hebben [11].

Bibliografie

[1] D. Colle, P. Van Heuven, C. Develder, S. Van den Berghe, I. Lievens, M. Pickavet, P. Demeester, "MPLS recovery mechanisms for IP-over-WDM networks", Photonic Network Commun. (PNET), Kluwer Academic Publishers, vol. 3, no. 1/2, Jan. 2001, pp. 23–40.

[2] S. De Maesschalck, D. Colle, A. Groebbens, C. Develder, I. Lievens, P. Lagasse, M. Pickavet, P. Demeester, F. Saluta, M. Quagliatti, "Intelligent optical networking for multilayer survivability", IEEE Commun. Mag., vol. 40, no. 1, Jan. 2002, pp. 42–49.

[3] D. Colle, S. Demaesschalck, C. Develder, P. Van Heuven, A. Groebbens, J. Cheyns, I. Lievens, M. Pickavet, P. Lagasse, P. Demeester, "Data-centric optical networks and their survivability", IEEE J. Selected Areas in Commun., vol. 20, no. 1, Jan. 2002, pp. 6–20.

[4] C. Develder, D. Colle, S. Demaesschalck, M. Pickavet, P. Demeester, "Influence of GMPLS recovery mechanisms on TCP performance", Photonic Network Commun. (PNET), vol. 4, no. 3/4, Jul.-Dec. 2002, pp. 321–345.

[5] J. Cheyns, C. Develder, E. Van Breusegem, E. Baert, A. Ackaert, M. Pickavet, P. Demeester, "Routing in an AWG based optical packet switch", Phot. Netw. Commun. (PNET), vol. 5, no. 1, Jan. 2003, pp. 69–80.

[6] L. Dittmann (ed.), C. Develder, D. Chiaroni, F. Neri, F. Callegati, W. Koerber, A. Stavdas, M. Renaud, A. Rafel, J. Solé-Pareta, W.Cerroni, N. Leligou, L. Dembeck, B. Mortensen, M. Pickavet, N. Le Sauze, M. Mahony, B. Berde, G. Eilenberger, "The European IST project DAVID: a viable approach towards optical packet switching", IEEE J. Selected Areas in Commun., vol. 21, no. 7, Sep. 2003, pp. 1026–1040.

[7] E. Baert, C. Develder, M. Pickavet, P. Demeester, "Routing strategies to minimize packet loss in an optical packet switched network with recirculating FDL buffers", zal verschijnen in Phot. Netw. Commun. (PNET), vol 7, no. 2, Mar. 2004.

[8] C. Develder, B. Van Houdt, C. Blondia, M. Pickavet, P. Demeester, "Analytical MMAP-based bounds for packet loss in optical packet switching with recirculating FDL buffers", submitted to PNET.

[9] J. Cheyns, C. Develder, E. Van Breusegem, E. Baert, D. Colle, M. Pickavet, P. Demeester, "Clos lives on in Optical Packet Switching", aanvaard voor publicatie in IEEE Commun. Mag.

[10] F. Callegati, C. Develder, W. Cerroni, M. Pickavet, P. Demeester,

"Scheduling algorithms for a slotted packet switch with either fixed or variable length packets", aanvaard voor publicatie in Photonic Network Commun. (PNET), Kluwer Academic Publishers.

[11] C. Matrakidis, C. Develder, A. Stavdas, M. Pickavet, P. Demeester, "Multistage switches for optical packet switching based on SOA technology", to be submitted to Optical Networks Magazine

[12] P. Demeester, D. Colle, S. Demaesschalck, C. Develder, M. Pickavet, P. Van Heuven, "Resilience in IP over WDM based multilayer networks", (Invited) The 26st European Conference on Optical Communication (ECOC 2000), Munich, Germany, 3-7 Sep. 2000.

[13] D. Colle, C. Develder, P. Van Heuven, M. Pickavet, P. Demeester, L. Raptis, G. Chatzilias, C. Mas, Y.I. Manolessos, J. Comellas, A. Rafel, J. Prat, J. Solé-Pareta, J. Moyano, S. Brunazzi, S. Rotolo, R. Stankiewicz, A. Gladish, "Recovery techniques for IP-over-WDM networks", (Invited) IP over DWDM Conf., Paris, France, 27-30 Nov. 2000.

[14] D. Colle, P. Van Heuven, A. Groebbens, C. Develder, M. Pickavet, P. Demeester, "Dimensioning reliable IP-over-WDM networks", IP over DWDM Conf., Paris, France, 27-30 Nov. 2000.

[15] P. Demeester, D. Colle, S. Demaesschalck, C. Develder, M. Pickavet, P. Van Heuven, "Resilience in IP over WDM based multilayer networks", (Invited) Optical Networks 2000, Göteborg, Sweden, 7 Dec. 2000.

[16] C. Develder, Didier Colle, Pim Van Heuven, Steven Van den Berghe, Mario Pickavet, Piet Demeester, "Influence of recovery time on TCP behaviour", MPLS World Congress, Paris, France, 6-9 Feb. 2001.

[17] D. Colle, C. Develder, P. Van Heuven, S. Demaesschalck, A. Groebbens, M. Pickavet, P. Demeester, "Resilience in IP-over-WDM networks", (Invited) Proc. 5th Working Conf. on Optical Netw. Design and Modelling (ONDM 2001) , Vienna, Austria, 5-7 Feb. 2001.

[18] A. Groebbens, D. Colle, C. Develder, S. De Maesschalck, M. Pickavet, P. Demeester, "Use of backup trees to improve resource efficiency of MPLambdaS protection mechanisms", Proc. 3rd Intl. Workshop on Design of Reliable Commun. Netw. (DRCN 2001), Budapest, Hungary, 7-10 Oct. 2001, pp. 152–159.

[19] D. Colle, S. De Maesschalck, C. Develder, A. Groebbens, M. Pickavet, P. Demeester, "Design and dimensioning issues in survivable data-centric optical networks", (Invited) Proc. 1st COST270 Workshop on Reliability of Optical Netw., Systems and Components, Dubendorf, Switzerland, 12-13 Dec. 2001.

[20] C. Develder, M. Pickavet, P. Demeester, "Assessment of packet loss for an optical packet router with recirculating buffer", Proc. 6th IFIP Working Conf. on Optical Netw. Design and Modelling (ONDM2002), Torino, Italy, 4-6 Feb. 2002, pp. 247–261.

[21] M. Pickavet, A. Ackaert, E. Baert, J. Cheyns, D. Colle, S. De Maesschalck, P. Demeester, C. Develder, A. Groebbens, I. Lievens, E. Van Breusegem, S. Verbrugge, Q. Yan, "Design of communication networks using heuristics", (Invited) Book of abstracts of "Arbeitskreis Mathematik in Forschung und

Praxis", 23rd Symp. on Mathematik in der Telekommunikation, Duisburg, Germany, 18-19 Mar. 2002.

[22] M. Pickavet, C. Develder, E. Baert, P. Demeester, "A.I. Techniques for planning telecommunication networks", Proc. Intl. Conf. on Artificial Intelligence (IC-AI'02), Las Vegas, NV, 24-27 Jun. 2002.

[23] C. Develder, J. Cheyns, E. Van Breusegem, E. Baert, A. Ackaert, M. Pickavet, P. Demeester, "Node architectures for optical packet and burst switching", Proc. Int. Topical Meeting on Photonics in Switching (PS2002), (invited) paper PS.WeA1, Cheju Island, Korea, 21–25 Jul. 2002, pp. 104–106.

[24] C. Develder, M. Pickavet, P. Demeester, "Strategies for an FDL based feed-back buffer for an optical packet switch with QoS differentiation", Proc. Conf. on Optical Internet (COIN2002), paper COIN.TuD1, Cheju Island, Korea, 21–25 Jul. 2002, pp. 114–116.

[25] C. Develder, M. Pickavet, P. Demeester, "Choosing an appropriate buffer strategy for an optical packet switch with a feed-back FDL buffer", Proc. 28th European Conference on Optical Communication (ECOC2002), Copenhague, Denmark, 8-12 Sept. 2002.

[26] C. Develder, J. Cheyns, M. Pickavet, P. Demeester, "Multistage architectures for optical packet switching using SOA-based broadcast-and-select switches", Tech. Digest Optical Fibre Conference (OFC 2003), paper FS4, Atlanta, GA, USA, 23-28 Feb. 2003, pp. 794–795.

[27] J. Cheyns, E. Van Breusegem, C. Develder, A. Ackaert, M. Pickavet, P. Demeester, "Performance improvement of an internally-blocking optical packet/ burst switch", Proc. Intl. Conf. on Commun. (ICC2003), Anchorage, AK, 11-15 May 2003, vol. 2 , pp. 1304–1308.

[28] D. Colle, J. Cheyns, C. Develder, E. Van Breusegem, A. Ackaert, M. Pickavet, P. Demeester, B. Feng, H. Lønsethagen, E. Zouganeli, J. Fernández-Palacios, J.F. Lobo, "GMPLS extensions for supporting advanced optical networking technologies", Proc. Intl. Conf. on Transparent Optical Networks (ICTON 2003), Warschau, Poland, 29 Jun. – 3 Jul. 2003.

[29] C. Develder, M. Pickavet, P. Demeester, "On trains and wagons: switching variable length packets in a slotted OPS network", Proc. Conference on Optical Internet / Australian Conference on Optical Fibre Technology (COIN/ACOFT 2003), paper TuA2-6, Melbourne, Australia, 13-16 Jul. 2003, pp. 233–236.

[30] C. Develder, R. Van Caenegem, E. Baert, M. Pickavet, P. Demeester, "Active versus passive OPS architectures for metro rings: a network dimensioning point of view", Proc. 29th European Conf. on Optical Commun. (ECOC 2003), paper We.1.4.4, Rimini, Italy, 21-25 Sep. 2003, vol. 3, pp. 376–377.

[31] C. Develder, J. Cheyns, E. Van Breusegem, E. Baert, D. Colle, M. Pickavet, P. Demeester, "Architectures for optical packet and burst switches" (invited), Proc. 29th European Conf. on Optical Commun. (ECOC 2003), paper Mo.4.4.1, Rimini, Italy, 21-25 Sep. 2003, vol. 1, pp. 100–103.

[32] C. Develder, J. Cheyns, M. Pickavet, P. Demeester, "Service differentiation

mechanisms for variable length packets in an optical switch with recirculating FDL buffer", Tech. Digest Photonics in Switching (PS2003), paper PS.Mo.C8,Versailles, France, 28 Sep.-2 Oct. 2003, pp. 92–94.

[33] R. Van Caenegem, C. Develder, E. Baert, D. Colle, M. Pickavet, P. Demeester, "Architectures for OPS metro rings: comparing active versus passive nodes / A dimensioning point of view", aanvaard voor 8th IFIP Working Conf. on Optical Network Design and Modelling (ONDM 2004).

[34] J. Cheyns, E. Van Breusegem, C. Develder, D. Colle, M. Pickavet, P. Demeester, "Evaluating cost functions for OPS node architectures / A packaging driven approach", aanvaard voor 8th IFIP Working Conf. on Optical Network Design and Modelling (ONDM 2004).

[35] D. Colle, C. Develder, S. De Maesschalck, A. Groebbens, M. Pickavet, P. Demeester, "Design and planning of reliable communication networks", 1st FTW PHD Symposium, Gent, Belgium, 5 Dec. 2000.

[36] C. Develder, J. Cheyns, E. Van Breusegem, E. Baert, M. Pickavet, P. Demeester, "Optical Packet/Burst Switching", 2nd FTW PHD Symposium, paper 100 (Proceedings available on CD-Rom), Interactive poster session, , Gent, Belgium, 11 Dec. 2001.

[37] C. Develder, E. Baert, M. Pickavet, P. Demeester, "Optical Packet Switched Networks with Recirculating FDL Buffers", 3rd FTW PHD Symposium, Gent, Belgium, 11 Dec. 2002.

[38] A. Groebbens, C. Develder, D. Colle, S. Demaesschalck, B. Lannoo, M. Pickavet, P. Demeester, "Optimization of backup tree structures, reducing spare capacity in optical networks, while retaining protection speed", 3rd FTW PhD Symposium, Gent, Belgium, 11 Dec. 2002.

[39] R. Inkret, et al., "Advanced Infrastructure for photonic networks / Extended final report of Cost 266 action", 2003.

[40] R.W. Wagner, "Communications Network Trends and Opportunities", Proc. Conf. On the Optical Internet/ Australian Conf. On Optical Fibre Technology (COIN/ACOFT 2003), Melbourne, Australia, 13-16 Jul. 2003.

[41] K.G. Coffman, A.M. Odlyzko, "Growth of the internet",
in: I.P. Kaminow, T. Li (eds.), "Optical fiber telecommunications IV B: systems and impairments", Academic Press, 4th ed., April 2002, pp. 17–56.

[42] C.K. Kao, G.A. Hockham, "Dielectric-fibre surface waveguides for optical frequencies", Proc. IEE, vol. 113, no.7, Jul. 1966, pp. 1151–1158.

[43] ITU Recommendation G.652, "Characteristics of a single mode optical fibre cable", 1997.

[44] J. Kim, O. Boyraz, M.N. Islam, "150+ channel ultra-DWDM source with N×10 GHz spacing utilizing longitudinal mode slicing of supercontinuum", Tech. Digest Optical Fibre Commun. Conf. and Exhibition (OFC 2000), Baltimore, MD, 7-9 Mar. 2000, vol. 3, pp. 5–7.

[45] Y. Hida, et al., "400-channel 25-GHz spacing arrayed-waveguide grating covering a full range of C- and L-bands", Tech. Digest Optical Fibre Commun. Conf. and Exhibition (OFC 2001), Anaheim, CA, 17-22 Mar. 2001, paper WB2.

[46] A. Neukermans, R. Ramaswami, "MEMS technology for optical networking

applications", IEEE Commun. Mag., vol. 39, no. 1, Jan. 2001, pp. 62–69.
[47] L. Garret, et al., "The MONET New Jersey Network Demonstration", IEEE J. on Selected Areas in Commun., vol. 16, no. 7, Sep. 1998, pp. 1199–1219.
[48] K. Shimano, C. Cavazzoni, A. D'Allessandro, A. Manzalini, "Architecture and functional requirements of control planes for Automatic Switched Optical Networks: experience of the IST project LION", Proc. Workshop on High Performance Switching and Routing (HPSR 2002), Kobe, Japan, 26-29 May 2002.
[49] G. Bernstein, E. Mannie, V. Sharma, "Framework for MPLS-Based control of optical SDH/SONET networks", IEEE Network, vol. 15, no. 4, Jul./Aug. 2001, pp. 20–26.
[50] K. Sato, N. Yamanaka, Y. Takigawa, M. Koga, S. Okamoto, K. Shiomoto, E. Oki, W. Imajuku, "GMPLS-based photonic multilayer router (Hikari router) architecture: an overview of traffic engineering and signaling technology", IEEE Commun. Mag., vol. 40, no. 3, Mar. 2002, pp. 96–101.
[51] L. Haché, L. Li, "Unified control infrastructure for carrier network evolution", IEEE Commun. Mag., vol. 38, no. 11, Nov. 2000, pp. 74–77
[52] S. Yao, B. Mukherjee, S. Dixit, "Advances in photonic packet switching: an overview", IEEE Commun. Mag., vol. 38, no. 2, Feb. 2000, pp. 84–94.
[53] D.K. Hunter, I. Andonovic, "Approaches to Optical Internet Packet Switching", IEEE Commun. Mag., Sep. 2000, pp. 116–122.
[54] C. Qiao, M. Yoo, "Optical Burst Switching (OBS) – a New Paradigm for an Optical Internet", J. High Speed Networks, vol. 8, no. 1, Jan. 1999.
[55] E. Van Breusegem, J. Cheyns, D. Colle, M. Pickavet, P. Demeester, "Overspill Routing In Optical Networks: a new architecture for future proof IP-over-WDM networks", Proc. Conf. on Optical Netw. and Communications (OptiComm 2003), Dallas, TX, 15–48 Oct. 2003.
[56] D. Chiaroni, et al., "New 10 Gbit/s 3R NRZ optical regenerative interface based on semiconductor optical amplifiers for all-optical networks", Proc. Int. Conf. Integrated Optics and Optical Fiber Commun. / European Conf. on Optical Commun. (IOOC/ECOC 97), Edinburgh, UK, 22-25 Sept. 1997, pp. 41–45.
[57] B. Lavigne, et. al., "Full validation of an optical 3R regenerator at 20 Gbit/s", Tech. Digest Optical Fibre Commun. Conf. and Exhibition (OFC 2000), Baltimore, MD, 7-9 Mar. 2000, vol. 3, pp. 93–95.
[58] B. Sartorius, "3R all-optical signal regeneration", Proc. European Conf. on Optical Commun. (ECOC 2001), Amsterdam, The Netherlands, 30 Sep. – 4 Oct. 2001, pp. 98–125.
[59] D. Chiaroni, B. Lavigne, A. Jourdan, L. Hamon, C. Janz, M. Renaud, "Feasibility assessment of a synchronization interface for photonic packet-switching systems", Proc. Int. Conf. Integrated Optics and Optical Fiber Commun. / European Conf. on Optical Commun. (IOOC/ECOC 97), Edinburgh, UK, 22-25 Sept. 1997, vol. 3, pp. 148–151.
[60] M. Burzio, P. Cinato, R. Finotti, P. Gambini, M. Puleo, E. Vezzoni, L. Zucchelli, "Optical cell synchronization in an ATM optical switch", Proc. European Conf. on Optical Commun (ECOC 94), Firenze, Italy, 25-29 Sep.

1994, pp. 581–584.
[61] T. Sakamoto, et al., “Optical packet synchronization for optical packet switches”, Proc. Conf. on Optical Internet (COIN2002), paper COIN.TuA3, Cheju Island, Korea, 21–25 Jul. 2002, pp. 78–80
[62] M. C. Cardakli, S. Lee, A. E. Willner, V. Grubsky, D. Starodubov, J. Feinberg, “Reconfigurable optical packet header recognition and routing using time-to-wavelength mapping and tunable fiber Bragg gratings for correlation decoding”, IEEE Photon. Technol. Lett., vol. 12, no. 5, May 2000, pp. 552–554.
[63] T. Fjelde, et al., “Novel scheme for simple label-swapping employing xor logic in an integrated interferometric wavelength converter,” IEEE Photon. Technol. Lett., vol. 13, no. 7, Jul. 2001, pp. 750–752.
[64] K. Kitayama. K. Onohara, M. Murata, “Capability of optical code-based MPLS (OC-MPLS)”, Proc. 6th IFIP Working Conf. on Optical Netw. Design and Modelling (ONDM2002), Torino, Italy, 4-6 Feb. 2002, pp. 263–278.
[65] H.J.S. Dorren, et al., “Optical packet switching and buffering by using all-optical signal processing methods”, IEEE J. Lightwave Techn., vol. 21, no. 1, Jan. 2003, pp. 2–12.
[66] M.T. Hill, H. De Waardt, G.D. Khoe, H.J.S. Dorren, “All-optical flip-flop based on coupled laser diodes”, IEEE J. Quantum Electron., vol. 37, Mar. 2001, pp. 157–159.
[67] C. Liu, Z. Dutton, C.H. Behroozi, L.V. Hau, “Observation of coherent optical information storage in an atomic medium using halted light pulses”, Nature, vol. 409, Jan. 2001, pp. 490–493 .
[68] D. F. Phillips, A. Fleischhauer, A. Mair, R.L.Walsworth, “Storage of Light in Atomic Vapor”, Physical Review Letters, vol. 86, Jan. 2001, pp 783–786.
[69] A. V. Turukhin, et. al., “Observation of ultraslow and stored light pulses in a solid”, Physical Review Letters, vol. 88, Dec. 2001.
[70] M.S. Bigelow, N.N. Lepeshkin, R.W. Boyd, “Observation of Ultraslow Light Propagation in a Ruby Crystal at Room Temperature”, Physical Review Letters, vol. 90, March 2003.
[71] A. Bianco, G. Galante, E. Leonardi, F. Neri, M. Rundo, “Access Control Protocols for Interconnected WDM Rings in the DAVID Metro Network”, Proc. 2001 Tyrrhenian Intern. Workshop on Digital Communications (IWDC 2001), Taormina, Italy, Sept. 2001.
[72] S. De Maesschalck, P. Demeester, “A functional model for the SONATA switchless optical network”, Proc. 4th IFIP Working Conf. on Optical Netw. Design and Modeling (ONDM 2000), Athens, Greece, 7-8 Feb. 2000, pp. 151–162.
[73] A. Bianco, E. Leonardi, M. Mellia, F. Neri, “Network controller procedures in SONATA, a large scale all-optical WDM network”, Proc. 4th IFIP Working Conf. on Optical Netw. Design and Modeling (ONDM 2000), Athens, Greece, 7-8 Feb. 2000, pp. 247–264.
[74] M.Pousa, C. Steenbergen, T. Almeida, J. Wellen, B. Vermeulen, “Harmonics – Hybrid Access Reconfigurable Multi-wavelength Optical Networks for IP-based Communication Services”, Proc. 5th World Multiconf. on Systemics,

Cybernetics and Informatics (SCI 2001), Orlando, FL, 22-25 Jul. 2001.

[75] G. Kramer, G. Pesavento, "Ethernet Passive Optical Network (EPON): building a next-generation optical access network", IEEE Commun. Mag., vol. 40, no. 2, Feb. 2002, pp. 66–73.

[76] P. Arijs, "Planning of ring-based telecommunication networks", PhD. Thesis, Ghent University, 2000-2001.

[77] A. Carena, V. Ferrero, R. Gaudino, V. De Feo, F. Neri, P. Poggiolini, "RINGO: A demonstator of WDM optical packet network on a ring topology", Proc. 6th IFIP Working Conf. on Optical Netw. Design and Modeling (ONDM 2002), Torino, Italy, 4-6 Feb. 2002, pp. 183–197.

[78] D. Dey, A. van Bochove, A. Koonen, D. Geuzebroek, M. Salvador, "FLAMINGO: a packet-switched IP-over-WDM all-optical MAN", Proc. European Conf. on Optical Commun. (ECOC 2001), Amsterdam, The Netherlands, 30 Sep. – 4 Oct. 2001, pp. 480–481.

[79] K. Shrikhande, et al., "HORNET: A packet-over-WDM multiple access metropolitan area ring network", IEEE J. Selected Areas in Commun., vol 18, no. 10, Oct. 2000, pp. 2004–2016.

[80] M.A. Marsan, A. Bianco, E. Leonardi, M. Meo, F.Neri, "MAC-protocols and fairness control in WDM multi-rings with tunable transmitters and fixed receivers", IEEE J. Selected Areas in Commun., vol. 14, no. 6, June 1996, pp. 1230–1244.

[81] I. Cidon, Y. Ofek, "MetaRing – A full-duplex ring with fairness and spatial reuse", IEEE Trans. on Commun., vol. 40, no. 1, Jan. 1993, pp. 110–120.

[82] A. Marsan, A. Bianco, E. Leonardi, A. Morabito, F. Neri, "All-optical WDM multi-rings with differentiated QoS", IEEE Commun. Mag., vol. 37, no. 2, Feb. 1999, pp. 58–66.

[83] K. Vlachos, et al., "An optical IM/FSK coding technique for the implementation of a label-controlled arrayed waveguide packet router", IEEE J. Lightwave Techn., vol. , no., Nov. 2003, pp. xx–xx.

[84] D.K. Hunter, et al., "WASPNET: A wavelength switched packet network", IEEE Commun. Mag., vol. 37, no. 3, Mar. 1999, pp. 120–129.

[85] Y. Lin, "Micromachined free space switches with sub millisecond switching times", Tech. Digest Optical Fiber Commun. Conf. and Exhibition (OFC 1998), 5-10 Mar. 1998, San Jose, CA, pp. 147–148.

[86] R. Ryf, et al., "1296-port MEMS transparent optical crossconnect with 2.07 Petabit/s switch capacity", Tech. Digest Optical Fiber Commun. Conf. and Exhibition (OFC 2001), 17-22 Mar. 2001, Anaheim, CA, postdeadline paper PD28.

[87] J.F. Fouquet, "Compact optical cross-connect switch based on total internalreflection in a fluid-containing planar lightwave circuit", Tech. Digest Optical Fiber Commun. Conf. and Exhibition (OFC 2000), 5-10 Mar. 2000, Baltimore, MA, pp. 204–206.

[88] L. Eldada, R. Norwood, R. Blomquist, L.W. Shacklette, M.J. McFarland, "Thermo-optically active polymeric photonic components", Tech. Digest Optical Fiber Commun. Conf. and Exhibition (OFC 2000), 5-10 Mar. 2000, Baltimore, MA, pp. 124–126.

[89] D. Enguang, W. Deming, X. Anshi, "High speed integrated acousto-optic switch with high extinction ratio", Tech. Digest Optical Fiber Commun. Conf. and Exhibition (OFC 2000), 5-10 Mar. 2000, Baltimore, MA

[90] A. Dugan, J.-C. Chiao, "The Optical Switching Spectrum: A Primer on Wavelength Switching Technologies", Telecommunication Magazine, May 2001.

[91] S. J. B. Yoo, et al., "Rapidly switching all-optical packet routing system with optical-label swapping incorporating tunable wavelength conversion and a uniform-loss cyclic frequency AWGR", Photonics Technology, vol. 14, no. 8, Aug. 2002, pp. 1211–1213.

[92] C. Guillemot, et al., "Transparent optical packet switching: the European ACTS KEOPS project approach", IEEE J. Lightwave Tech., vol. 16, no. 12, Dec. 1998, pp. 2117–2134.

[93] D. Chiaroni, et al., "First demonstration of an asynchronous optical packet switching matrix prototype for muti-terabit class routers/switches", Proc. European Conf. on Optical Commun. (ECOC 2001), Amsterdam, The Netherlands, 30 Sep. – 4 Oct. 2001, postdeadline paper ThA4.11.

[94] S.L. Danielsen, C. Joergensen, B. Mikkelsen, K.E. Stubkjaer, "Analysis of a WDM packet switch with improved performance under bursty traffic conditions due to tuneable wavelength converters", IEEE J. Lightwave Technology, vol. 16, no. 5 , May 1998, pp. 729–735.

[95] V. Eramo, M. Listanti, "Packet loss in a bufferless optical WDM switch employing shared tunable wavelength converters", IEEE J. Lightwave Techn., vol. 18, no. 12 , Dec. 2000, pp. 1818–1833.

[96] D.K. Hunter, M.C. Chia, I. Andonovic, "Buffering in optical packet switches", J. Lightwave Techn., vol. 16, no. 12, Dec. 1998, pp. 2081–2094.

[97] Z. Haas, "The 'staggering switch': an electronically controlled optical packet switch", J. Lightwave Techn., vol. 11, no. 5, May-Jun 1993, pp. 925–936.

[98] S. Yao, B. Mukherjee, S. J. Ben Yoo, S. Dixit, "All-Optical Packet-Switched Networks: A Study of Contention-Resolution Schemes in an Irregular Mesh Network with Variable-Sized Packets", Proc. Conf. on Optical Netw. and Commun. (OptiComm 2000), 22-26 Oct. 2000, Plano, TX.

[99] D.K. Hunter, et al., "SLOB: a switch with large optical buffers for packet switching", J. Lightwave Techn., vol. 16, no. 10, Oct. 1998, pp. 1725–1736.

[100] S. Bjornstad, D.R. Hjelme, N. Stol, "An optical packet switch design with shared electronic buffering and low bit rate add/drop inputs", Proc. of Int. Conf. on Transparent Optical Netw. (ICTON 2002), Warzaw, Poland, April 2002, pp. 69–72.

[101] J. Masip, J. Solé-Pareta, "Providing differentiated service categories in optical packet networks", Proc. 16th Intl. Teletraffic Congress (ITC 1999), Edinburg, UK, 7-11 Jun. 1999, vol. 3, pp. 1115-1126.

[102] R.W. Stevens, "TCP/IP Illustrated, Volume 1: The Protocols, Addison-Wesley, 1994.

[103] M. De Prycker, "Asynchronous Transfer Mode: solution for broadband ISDN", Prentice Hall Intl. Ltd., UK, 1995.

[104] P.B. Hansen, S.L. Danielsen, K.E. Stubkjaer, "Optical packet switching

without packet alignment", Proc. European Conf. on Optical Commun. (ECOC 1998), Madrid, Spain, 22-24 Sep. 1998, pp. 591–592.

[105] B. Mukherjee, A.E. Kamal, "Scheduling variable-length messages on slotted, high-speed fiber optic LANs/MANs using the continuation-bit approach", Proc. 10th IEEE Conf. on Computer Commun. (Infocom 1991), Bal Harbour, FL, 7-11 Apr. 1991, vol. 2, pp. 678–687

[106] F. Callegati, W. Cerroni, "Time-wavelength exploitation in optical feedback buffer with trains of packets", Proc. Conf. on Optical Netw. and Communications (OptiComm 2002), Boston, MA, 29 Jul. – 2 Aug. 2002, pp. 274–285.

[107] M. Nord, S. Bjornstad, C. Gauger, "OPS or OBS in the core network?", Proc. 7th IFIP Working Conf. on Optical Netw. Design and Modeling (ONDM 2003), Budapest, Hungary, 3-5 Feb. 2003.

[108] A. Ge, F. Callegati, L.S. Tamil, "On optical burst switching and self-similar traffic", IEEE Commun. Letters., vol. 4, no. 3, Mar. 2000.

[109] Y. Xiong, M. Vandenhoute, H.C. Cankaya, "Control architecture in optical burst-switched WDM networks", IEEE J. Selected Areas in Commun., vol. 18, no. 10, Oct. 2000, pp. 1838–1851

[110] S.F. Su, R. Olshansky, "Performance of multiple access WDM networks with subcarrier multiplexed control channels", IEEE J. Lightwave Techn., vol. 11, no. 5, May-June 1993, pp. 1028–1033.

[111] C. L. Lu, D. J. M. Sabido, P. Poggiolini, R. T. Hofmeister, L.G. Kazovsky, "CORD—A WDMA optical network: Subcarrier-based signaling and control scheme", IEEE Photon. Technol. Lett., vol. 7, no. 5, May 1995, pp. 555–557.

[112] N. Chi, et al., "Optical label swapping and packet transmission based on ASK/DPSK orthogonal modulation format in IP-over-WDM networks", Tech. Digest Optical Fiber Commun. Conf. and Exhibition (OFC 2003), 23-28 Mar. 2003, Atlanta, GA, pp. 792–794.

[113] X. Xiao, L.M. Ni, "Internet QoS: the big picture", IEEE Netw. Mag., vol. 13, no. 2, March 1999, pp. 8–18.

[114] D. Kettler, H.Kafka, D. Spears, "Driving fiber to the home", IEEE Commun. Mag., no. 38, no. 11, Nov. 2000, pp. 106–110.

[115] J. Kawataka, et al., "Novel optical fiber cable for feeder and distribution sections in access network", IEEE J. Lightwave Techn., vol. 21, no. 3, Mar. 2003, pp. 789–796.

[116] N. Le Sauze, ed., "Network concepts validation and benchmarking", Project Deliverable D101 of the IST-project DAVID, Sep. 2003.

[117] S. Yao, S. J. Ben Yoo, B. Mukherjee, S. Dixit, "All-optical packet switching for metropolitan area networks: Opportunities and challenges", IEEE Commun. Mag., vol. 39, no. 3, March 2001, pp. 142–148.

[118] A. Stavdas, et al., "IST-DAVID: concept presentation and physical layer modeling of the metropolitan area network", IEEE J. Lightwave Techn., vol. 21, no. 2 , Feb. 2003, pp. 372–383.

[119] T.K. Woodward, A.L. Lentine, J.D. Fields, G. Giaretta, R. Limacher, "First demonstration of native Ethernet optical transport system prototype at 10 Gb/s based on multiplexing of gigabit Ethernet signals", IEEE Photonics

Techn. Lett.,vol. 12, no. 8, Aug. 2000, pp. 1100–1102.
[120] D.G. Cunningham, "The status of the 10-Gigabit Ethernet standard", Proc. European Conf. on Optical Commun. (ECOC 2001), Amsterdam, The Netherlands, 30 Sep. – 4 Oct. 2001, pp. 364–367.
[121] ——, "An Introduction to Resilient Packet Ring Technology", white paper by the RPR Alliance, online: http://www.rpralliance.org/articles/Whitepaper10-01.pdf
[122] A. Jourdan, D. Chiaroni, E. Dotaro, G.J. Eilenberger, F. Masetti, M. Renaud, "The perspective of optical packet switching in IP dominant backbone and metropolitan networks", IEEE Commun. Mag., vol. 39, no. 3, Mar. 2001, pp. 136–141.
[123] A. Takada, J.H. Park, "Architecture of ultrafast optical packet switching ring network", IEEE J. Lightwave Techn., vol. 20, no. 12, Dec. 2002, pp. 2306–2315.
[124] N. Le Sauze, et al., "A novel, low cost optical packet metropolitan ring architecture", Proc. European Conf. on Optical Commun. (ECOC 2001), Amsterdam, The Netherlands, 30 Sep. – 4 Oct. 2001, vol.6, pp. 66–67.
[125] N. Le Sauze, et al., "DBORN: a shared WDM Ethernet bus architecture for optical packet metropolitan networks", Tech. Digest Int. Topical Meeting on Photonics in Switching (PS2002), paper PS.TuC2, Cheju Island, Korea, 21-25 Jul. 2002, pp. 89–91.
[126] H. Skouffis, ed., "Results of prospective studies and modelling support", Project Deliverable D203 of the IST-project DAVID, Sep. 2003.
[127] R. Van Caenegem, "Planning van een OPS metro-netwerk", Master Thesis, Ghent University, 2002–2003.
[128] P. Arijs, P. Demeester, "The merit of shared and dedicated protection WDM rings in a hybrid network design", Tech. Digest Optical Fibre Commun. Conf. and Exhibition (OFC 2000), Baltimore, MD, 7-9 Mar. 2000, vol. 4, pp. 93–95.
[129] J.-K. Rhee, I. Tomkos, M.-J. Li, "A broadcast-and-select OADM optical network with dedicated optical-channel protection", IEEE J. Lightwave Techn., vol. 21, no. 1 , Jan. 2003, pp. 25–31.
[130] D. Chiaroni, N. Le Sauze, T. Zami, J.-Y. Emery, "Semiconductor optical amplifiers: a key technology to control the packet power variation", Proc. European Conf. on Optical Commun. (ECOC 2001), Amsterdam, The Netherlands, 30 Sep. – 4 Oct. 2001, vol. 3, pp. 314–315.
[131] D. Wolfson, T. Fjelde, A. Kloch, "Technologies for all-optical wavelength conversion in DWDM networks", Proc. Conf on Lasers and Electro-Optics (CLEO/Pacific Rim 2001), 15-19 Jul. 2001, Chiba, Japan, pp. II-574 – II-575.
[132] C. Clos, "A study of non-blocking switching networks, Bell System Technical Journal", vol. 32, 1953, pp. 406–424.
[133] R. A. Thompson, D. K. Hunter, "Elementary photonic switching modules in three divisions", IEEE J. of Selected Areas in Commun., vol. 14, no. 2, Feb. 1996, pp. 362–373.
[134] R. I. MacDonald, "Large modular expandable optical switching matrices",

Photonic Techn. Lett., vol. 11, no. 6, Jun. 1999, pp. 668–670.
[135] I. Lelić, "Large multi-stage OXC", Proc. European Conf. on Optical Commun. (ECOC 2001), Amsterdam, The Netherlands, 30 Sep. – 4 Oct. 2001, vol.4, pp. 540–541.
[136] L. Wosinska, L. Thylen, R. Holmstrom, "Large-capacity strictly nonblocking optical cross-connects based on microelectrooptomechanical systems (MEOMS) switch matrices: reliability performance analysis", IEEE J. Ligthwave Techn., vol. 19, no. 8, Aug. 2001, pp. 1065–1075.
[137] G. Shen, T. H. Cheng, S. K. Bose, C. Lu, T. Y. Chai, "Architectural design for multistage 2-D MEMS optical switches", IEEE J. Lightwave Techn., vol. 20, no. 2, Feb. 2002, pp. 178–187.
[138] O. Gerstel, P. Lin, "Efficient architecture for a duplex multi-stage optical non-blocking switch", Tech. Digest Optical Fibre Commun. Conf. and Exhibition (OFC 2000), Baltimore, MD, 7-9 Mar. 2000, vol. 2, pp. 350–352.
[139] Y.-K. Chen, C.-C. Lee, "Fiber-Bragg Grating-based large nonblocking multiwavelength cross-connects", IEEE J. Ligthwave Techn., vol. 16, no. 10, Oct. 1998, pp. 1746–1756.
[140] H.J. Chao, C.H. Lam, E. Oki, "Broadband packet switching technologies", John Wiley & Sons, Inc., 2001.
[141] F.T. Leighton, "Introduction to parallel algorithms and architectures: Arrays – Trees – Hypercubes", Morgan Kaufmann, 1992.
[142] M. Collier, T. Curran, "The strictly nonblocking condition in three-stage networks", Proc. 14th International Teletraffic Congress (ITC 14), Antibes, 6-10 Jun. 1994.
[143] A. M. Hill1, D. Careglio, J. Solé-Pareta, A. Rafel, "Relative costs of WDM rings and PONs for metro optical packet networks", Proc.
[144] B.B. Mortensen, M.S. Berger, "Optical Packet Switching Demonstrator", Proc. European Conf. on Optical Commun. (ECOC 2002), Copenhagen, Denmark, 8-12 Sep. 2002.
[145] S. Yao, F. Xue, B. Mukerjee, S.J.B Yoo, S. Dixit, "Electrical ingress buffering and traffic aggregation for optical packet switching and their effect on TCP-level performance in optical mesh networks", IEEE Commun. Mag., vol. 40, no. 9, Sep. 2002, pp. 66–72.
[146] B.B. Mortensen, M.S. Berger, "Estimating timeout parameters for packet aggregation", Proc. Conf. on the Optical Internet / Australian Conf. on Optical Fibre Techn. (COIN/ACOFT 2003), Melbourne, Australia, 13-16 Jul. 2003, pp. 225–228.
[147] M. Düser, P. Bayvel, "Performance of a dynamically wavelength-routed optical burst switched network", IEEE Photonic Techn. Lett., vol. 14, no. 2, Feb. 2002, pp. 239–241
[148] W. Willinger, M. Taqqu, R. Sherman, D. Wilson, "Self-similarity through high-variability: statis-tical analysis of Ethernet LAN traffic at the source level", IEEE/ACM Trans. Netw., vol. 5, no. 1, Jan. 1997, pp. 71–86.
[149] M.E. Crovella and A. Bestravos, "Self-similarity in World Wide Web traffic: evidence and possible causes", IEEE/ACM Trans. Netw., vol. 5, no. 6, Dec. 1997, pp. 835–846.

[150] K. Park, G. Kim, M. Crovella, “On the effect of traffic self-similarity on network performance”, Proc. SPIE Int. Conf. on Perf. and Control of Netw. Sys., Nov. 1997, pp. 296–310.
[151] Y. Zhou, H. Sethu, “A simulation study of the impact of switching systems on self-similar properties of traffic”, Proc. Workshop on Statistical Signal and Array Processing, 14-16 Aug. 2000, pp. 500–504.
[152] F. Xue, S. Yao, B. Mukherjee, S.J.B. Yoo, “The performance improvement in optical packet-switched networks by traffic shaping of self-similar traffic”, Tech. Digest Optical Fiber Commun. Conf. and Exhibit (OFC 2002), Anaheim, CA, 17-22 Mar. 2002.
[153] H. Christiansen, “Using OPNET to compare and analyse different traffic bundling schems”, Proc. OPNETWork 2001, Washington, DC, 27-31 Aug. 2001.
[154] M. Izal, J. Aracil, “On the influence of self-similarity on optical burst switching traffic”, Proc. Global Telecommun. Conf. (Globecom 2002), Taipei, Taiwan, 17-21 Nov. 2002, vol. 3, pp. 2308–2312.
[155] J.D. Angelopolous, N. Leligou, H. Linardakis, A. Stavdas, “A QoS-sensitive MAV for slotted WDM metropolitan rings”, Proc. 6th IFIP Working Conf. on Optical Netw. Design and Modeling (ONDM 2002), Torino, Italy, 4-6 Feb. 2002, pp. 3–15.
[156] A. Bianco, G. Galante, E. Leonardi, F. Neri, “Measurement based resource allocation for interconnected WDM rings”, Photonic Netw. Commun. (PNET), vol. 5, no. 1, Jan. 2003, pp. 5–22.
[157] T. Inukai, “An Efficient SS/TDMA Time Slot Assignment Algorithm”, IEEE Trans. on Commun., vol. 27, no. 10 , Oct. 1979. pp. 1449–1455.
[158] F. Callegati, D. Careglio, W. Cerroni, J. Sole-Pareta, “Assessment of packet loss for an optical feedback buffer node using slotted variable-length packets and heavy-tailed traffic”, Proc. Int. Conf. on Transparent Optical Networks (ICTON 2002), Warschau, Poland, 21-25 Apr. 2002, pp. 51-56.
[159] G. Bianchi, J.S. Turner, “Improved queueing analysis of shared buffer switching networks”, IEEE/ACM Transactions on Networking, no. 4, August 1993, pp. 482-490.
[160] A. Monterosso, A. Pattavina, “Performance analysis of multistage interconnection networks with shared-buffered switching elements for ATM switching”, Proc. 11th IEEE Conf. on Computer Commun. (Infocom 1992), Florence, Italy, May 1992, pp. 124-131.
[161] M.G. Hluchyj, M.J. Karol, “Queueing in High-Performance Packet Switching”, IEEE J. Selected Areas in Commun., no. 9, December 1988, pp. 1587–1597
[162] G. Shen, S.K. Bose, T. Hiang Cheng, C. Lu and T. Yoong Chai, “Performance study on a WDM packet switch with limited-range wavelength converters”, IEEE Commun. Letters, vol. 5, no. 10, October 2001, pp. 432–434.
[163] F. Callegati, G. Corazza, C. Rafaelli, “Exploitation of DWDM for optical packet switching with quality of service guarantees”, IEEE J. Selected Areas in Commun., vol. 20, no. 1, Jan. 2002, pp. 190–201.

[164] H. Zang, J.P. Jue and B. Mukherjee, "Photonic slot routing in all-optical WDM mesh networks", Proc. IEEE Global Telecommun. Conf. (Globecom 1999), vol. 1, Rio de Janeiro, Brazil, 5-9 Dec. 1999, pp. 1449–1453.

[165] A. Kushwaha, Sanjay K. Bose and Y. N. Singh, "Analytical modeling for performance studies of an FLBM-based all-optical packet switch", IEEE Commun. Letters, vol. 5, no. 5, May 2001, pp. 227–229.

[166] P.D. Bergstrom Jr., M.A. Ingram, A. J. Vernon, J.L.A. Hughes, P. Tetali, "A Markov Chain Model for an Optical Shared-Memory Packet Switch", IEEE Trans. on Commun., vol. , no. 10, October 1999, pp. 1593-1603

[167] C.A. Brackett, et al., "A scalable multiwavelength multihop optical network: a proposal for research on all-optical networks", IEEE J. Lightwave Techn., vol. 11, no. 5, May/Jun. 1993, pp. 736–753.

[168] C. Gauger, "Dimensioning of FDL buffers for optical burst switching nodes", Proc. 6th IFIP Working Conf. on Optical Netw. Design and Modeling (ONDM 2002), Torino, Italy, 4-6 Feb. 2002, pp. 117–132.

[169] Q. He, M.F. Neuts, "Markov chains with marked transitions", Stochastic Processes and their Applications, vol. 74, no. 1, Oct. 1998, pp. 37–52.

[170] K. Pawlikowski, H.-D.J. Jeong, J.-S.R. Lee, "On credibility of simulation studies of telecommunication studies", IEEE Commun. Mag., vol. 40, no. 1, Jan. 2002, pp. 132–139.

[171] M. Matsumoto, T. Nishimura, "Mersenne-twister: a 623-dimensionally equidistributed uniform pseudo-random number generator", ACM Trans. Modeling and Comp. Sim., vol. 8, no. 1, Jan. 1998, pp. 3–30

[172] A.M. Law, W.D. Kelton, "Simulation modeling and analysis", 3rd ed., McGraw Hill, Singapore, 2000.

[173] W. Cerroni, ed., "Traffic models for optical packet networks with quality of service differentiation", Project Deliverable D121 of the IST-project DAVID, Jul. 2001.

[174] C.J. Bovy, H.T. Mertodimedjo, G. Hooghiemstra, H. Uijterwaal, P. Van Mieghem, "Analysis of end-to-end delay measurements in Internet", Proc. Passive and Active Measurement Workshop (PAM2002), Fort Collins, CO, 25-26 Mar. 2002.

[175] S. Blake, et al., "An architecture for differentiated services", Internet RFC 2475, Informational, Dec. 1998.

[176] R. Braden, D. Clark, S. Shenker, "Integrated services in the Internet architecture: an overview", Internet RFC 1633, Informational, Jun. 1994.

[177] Y. Xu, R. Guérin, "Individual QoS versus aggregate QoS: A loss performance study", Proc. 21th IEEE Conf. on Computer Commun. (Infocom 2002), New York, NY, 23-27 Jun. 2002.

[178] D.E. Knuth, "The art of computer programming, volume 1: fundamental algorithms", Third Edition, Addison-Wesley, 1997.

[179] F. Callegati, "Approximate modeling of optical buffers for variable length packets", Photonic Netw. Commun. (PNET), vol. 3, no. 4, Oct. 2001, pp. 383–390.

[180] L. Tancevski, S. Yegnanarayanan, G. Castanon, L. Tamil, F. Masetti, T. McDermott, "Optical routing of asynchronous, variable length packets",

IEEE J. Selected Areas in Commun., vol. 18, no. 10, Oct. 2000, pp. 2084-2093.

[181] J.S. Turner, "Terabit burst switching", J. High Speed Netw., vol. 8, no. 1, Jan. 1999, pp. 3–16.

[182] B. Lannoo, J. Cheyns, E. Van Breusegem, A. Ackaert, M. Pickavet, P. Demeester, "A performance study of different OBS scheduler implementations", Proc. 7th Annual Symp. of the IEEE/LEOS Benelux Chapter, Amsterdam, The Netherlands, 9 Dec. 2002, pp. 191–194.

[183] B. Lannoo, "Studie van optical burst switching (OBS) aan de hand van simulaties", Master Thesis, Ghent University, 2001-2002.

[184] M. Yoo, C. Qiao, "Supporting multiple classes of services in IP over WDM networks", Proc. IEEE Global Telecommun. Conf. (Globecom 1999), Rio de Janeiro, Brazil, 5-9 Dec. 1999, vol. 1b, pp. 1023-1027.

[185] S. Yao, B. Mukherjee, S. J. B. Yoo, S. Dixit, "A unified study of contention-resolution schemes in optical packet-switched networks", IEEE J. Lightwave Techn, vol. 21, no. 3, Mar. 2003, pp. 672–683.

[186] S. Yao, S. J. B. Yoo, B. Mukherjee, "A comparison study between slotted and unslotted all-optical packet-switched network with priority-based routing", Tech. Digest Optical Fibre Commun. Conf. and Exhibition (OFC 2001), Anaheim, CA, 17-22 Mar. 2001.

[187] R. Jain, S. Routhier, "Packet Trains—Measurements and a New Model for Computer Network Traffic", IEEE J. Selected Areas in Commun., vol.4, no. 6, Sep. 1986, pp. 986–995.

[188] K. Dolzer, C. Gauger, J. Späth, S. Bodamer, "Evaluation of reservation mechanisms for optical burst switching", AEÜ Intl. J. Electron. Commun., vol. 55, no. 1, Jan. 2001.

[189] A. Kaheel, T. Khattab, A. Mohamed, H. Alnuweiri, "Quality-of-service mechanisms in IP-over-WDM networks", IEEE Commun. Mag., vol. 40, no. 12, Dec. 2002, pp. 38–43.

[190] J. Liu, N.Ansari, "Class-based dynamic buffer allocation for optical burst switching networks", Proc. Workshop on High Performance Switching and Routing (HPSR 2002), Kobe, Japan, 26-29 May 2002.

[191] A. Detti, V. Eramo, M. Listanti, "Performance evaluation of a new technique for IP support in a WDM optical network: optical composite burst switching (OCBS)", IEEE J. Lightwave Techn., vol. 20, no. 2, Feb. 2002, pp. 154–165

[192] A. Maach, G.V. Bochmann, "Segmented burst switching: enhancement of optical burst switching to decrease loss rate and support quality of service", Proc. 6th IFIP Working Conf. on Optical Netw. Design and Modelling (ONDM2002), Torino, Italy, 4-6 Feb. 2002, pp. 69–84.

[193] Y. Chen, M. Hamdi, D.H.K. Tsang, "Proportional QoS over OBS networks", Proc. IEEE Global Telecommun. Conf. (Globecom 2001), San Antonio, TX, 25-29 Nov. 2001, vol.3, pp. 1510–1514.

[194] C. Loi, W. Liao, D. Yang, "Service differentiation in optical burst switched networks", Proc. Globecom 2002.

[195] R. Ramaswami, K. Sivarajan, "Routing and wavelength assignment in all-optical networks", IEEE/ACM Trans. on Netw., vol. 3, no. 5, Oct. 1995, pp.

489–500.
[196] I. Chlamtac, A. Ganz, G. Karmi, “Lightpath communications: an approach to high bandwidth optical WANs”, IEEE/ACM Trans. on Commun., vol. 40, no. 7, Jul. 1992, pp. 1171–1182.
[197] N. Wauters, P. Demeester, “Design of the optical path layer in multiwavelength cross-connected networks”, IEEE J. Selected Areas in Commun., vol. 14, no. 5, Jun. 1996, pp. 881–892.
[198] N. Nagatsu, S. Okamoto, K. Sato, “Optical path cross-connect system scale evaluation using path accommodation design for restricted wavelength multiplexing”, IEEE J. Selected Areas in Commun., vol. 14, no. 5, Jun. 1996, pp. 893–902.
[199] S. Baroni, P. Bayvel, “Wavelength requirements in arbitrarily connected wavelength-routed optical networks”, IEEE J. Lightwave Techn., vol. 15, no. 2, Feb. 1997, pp. 242–251.
[200] B. Van Caenegem, W. Van Parys, F. De Turck, P. Demeester, “Dimensioning of survivable WDM networks”, J. Selected Areas in Commun., vol. 16, no. 7, Sep. 1998, pp. 1146–1157.
[201] J. Moy, “OSPF version 2”, Internet RFC 2328, Standards Track, Apr. 1998.
[202] D. Oran, “OSI IS-IS Intra-domain routing protocol”, Internet RFC 1142, Feb. 1990.
[203] B. Fortz, M. Thorup, “Internet traffic engineering by optimizing OSPF weights”, Proc. 19th IEEE Conf. on Computer Commun. (Infocom 2000), Tel Aviv, Israel, 26-30 Mar. 2000, pp. 519–528.
[204] A. Elwalid, Low C. Jin, I. Widjaja, “MATE: MPLS adaptive traffic engineering”, Proc. 20th IEEE Conf. on Computer Commun. (Infocom 2001), Anchorage, AK, 22-26 Apr. 2001, pp. 1300–1309.
[205] D. Awduche, “MPLS and traffic engineering in IP networks”, IEEE Commun. Mag., vol. 37, no. 12, Dec. 1999, pp. 42–47.
[206] G. Swallow, “MPLS advantages for traffic engineering”, IEEE Commun. Mag., vol. 37, no. 12, Dec. 1999, pp. 54–57.
[207] A. Lardies, R. Gupta, R. Patterson, “Traffic grooming in a multi-layr network”, Optical Netw. Mag., vol. 2, no. 3, May/Jun. 2001, pp. 91–99.
[208] M. Gryseels, P. Demeester, “A multi-layer planning approach for hybrid SDH-based ATM networks”, Proc. 6th Intl. Conf. on Telecommun. Systems (ICTS), Nashville, TN, 5-8 Mar. 1998, pp. 461–474.
[209] S. De Maesschalck, M. Pickavet, D. Colle, P. Demeester, “Multi-layer traffic grooming in networks consisting of an IP/MPLS Layer on top of a meshed optical network layer”, accepted for IEEE Global Telecommun. Conf. (Globecom 2003), 1-5 Dec. 2003, San Francisco, CA.
[210] A. Groebbens, D. Colle, S. De Maesschalck, M. Pickavet, P. Demeester, “Logical topology optimisation for dynamic multi-layer recovery schemes”, accepted for 4th Intl. Workshop on Design of Reliable Commun. Netw. (DRCN 2003), Alberta, Canada, 19-22 Oct. 2003.
[211] S. De Maesschalck, M. Pickavet, D. Colle, P. Demeester, “Influence of the IP traffic asymmetry on the cost of the optical network layer”, Photonic Netw. Commun. (PNET), vol. 4, no. 2, May 2002, pp. 133–150.

[212] S. De Maesschalck, et al., “Pan-European optical transport networks: an availability-based comparison”, Photonic Netw. Commun. (PNET), vol. 5, no. 3, May 2003, pp. 203–225.
[213] N. Ghani, “Lambda-labeling: a framework for IP-over-WDM using MPLS”, Optical Netw. Mag., vol. 1, no. 2, April 2000, pp. 45–48.
[214] Y. Ye, S. Dixit, M. Ali, “On joint protection/restoration in IP-centric DWDM based optical transport networks”, IEEE Commun. Mag., vol. 38, no. 6, Jun. 2000, pp. 174–183.
[215] S. Leinen, cited on *http://www.cs.columbia.edu/~hgs/internet/traffic.html*, Feb. 2001.
[216] A.S. Tanenbaum, “Computer networks”, 4th ed., Prentice Hall, Aug. 2002.
[217] S. Floyd, “A report on recent developments in TCP congestion control”, IEEE Commun. Mag., vol. 39, no. 4, Apr. 2001, pp. 84–90.
[218] J. Postel, “Transmission control protocol”, Internet RFC 793, Standards Track, Sep. 1981.
[219] S. Floyd, T. Henderson, “The NewReno modification to TCP's fast recovery algorithm”, Internet RFC 2582, Experimental,. Apr. 1999.
[220] R. Braden (ed.), “Requirements for internet hosts – communication layers”, Internet RFC 1122, Standards Track,. Oct. 1989.
[221] V. Paxson, M. Allman, “Computing TCP's retransmission timer”, Internet RFC 2988, Standards Track, Nov. 2000.
[222] D. Colle, A. Groebbens, P. Van Heuven, S. De Maesschalck, M. Pickavet, P. Demeester, “Porting MPLS-recovery techniques to the MPlS paradigm”, (Invited), Optical Netw. Mag., vol. 2, no. 4, Jul./Aug. 2001, pp. 29–47.
[223] P. Demeester, et al., “Resilience in multilayer networks”, IEEE Commun. Mag., vol. 37, no. 8, Aug. 1999, pp. 70–76.
[224] V. Sharma, F. Hellstrand (eds.), “Framework for MPLS-based recovery”, Internet RFC 3469, Informational, Feb. 2003.
[225] C. Huang, V. Sharma, K. Owens, S. Makam, “Building reliable MPLS networks using a path protection mechanism”, IEEE Commun. Mag., vol. 40, no. 3, Mar. 2002, pp. 156–162.
[226] R. Goguen, G. Swallow, “RSVP label allocation for backup tunnels”, Internet Draft, *draft-swallow-rsvp-bypass-label-01.txt*, Work in Progress, Nov. 2000.
[227] D. Haskin, R. Krishnan, “A method for aetting an alternative label switched paths to handle fast reroute”, Internet Draft, *draft-haskin-mpls-fast-reroute-05.txt*, Work in Progress, Nov. 2000.
[228] L. Breslau, et al., “Advances in network simulation”, IEEE Computer, vol. 33, no. 5, May 2000, pp.59–67.
[229] S. Demaesschalck, “Planning and evaluation of multi-layer optical networks”, PhD. Thesis, Ghent University, 2003.
[230] D. Colle, “Design and evolution of data-centric optical networks”, PhD. Thesis, Ghent University, Feb. 2002.
[231] F. De Turck, “Design of a generic software architecture for efficient managgement of integrated broadband networks and scheduled multimedia services”, PhD. Thesis, Ghent University, May 2002.

[232] F. Vandermeulen, “On the design of a generic, scalable, and open distributed software architecture for control and management of integrated broadband networks and multimedia services”, PhD. Thesis, Ghent University, May 2001.

Appendices

[A] C. Develder, J. Cheyns, E. Van Breusegem, E. Baert, D. Colle, M. Pickavet, P. Demeester, “Architectures for optical packet and burst switches”, invited for 29th European Conference on Optical Communications (ECOC 2003), Rimini, Italy, 21-25 Sep. 2003

[B] C. Develder, R. Van Caenegem, E. Baert, M. Pickavet, P. Demeester, “Active versus passive OPS architectures for metro rings: a network dimensioning point of view”, accepted for 29th European Conference on Optical Communications (ECOC 2003), Rimini, Italy, 21-25 Sep. 2003

[C] C. Develder, J. Cheyns, M. Pickavet, P. Demeester, “Multistage architectures for optical packet switching using SOA-based broadcast-and-select switches”, Tech. Digest Optical Fibre Conference (OFC 2003), paper FS4, Atlanta, GA, USA, 23-28 Feb. 2003, pp. 794–795.

[D] C. Develder, M. Pickavet, P. Demeester, “Assessment of packet loss for an optical packet router with recirculating buffer”, Proc. 6th IFIP Working Conference on Optical Network Design and Modelling (ONDM2002), Torino, Italy, 4 6 Feb. 2002, pp. 247–261.

[E] C. Develder, M. Pickavet, P. Demeester, “Strategies for an FDL based feed-back buffer for an optical packet switch with QoS differentiation”, Proc. Int. Conf. on Optical Internet (COIN2002), paper COIN.TuD1, Cheju Island, Korea, 21–25 Jul. 2002, pp. 114–116.

[F] C. Develder, J. Cheyns, M. Pickavet, P. Demeester, “Service differentiation mechanisms for variable length packets in an optical switch with recirculating FDL buffer”, accepted for PS 2003, Paris, France, 28 Sep.-2 Oct. 2003

[G] C. Develder, M. Pickavet, P. Demeester, “On trains and wagons: switching variable length packets in a slotted OPS network”, Proc. Conference on Optical Internet / Australian Conference on Optical Fibre Technology (COIN/ACOFT 2003), paper TuA2-6, Melbourne, Australia, 13-16 Jul. 2003, pp. 233–236.

[H] E. Baert, C. Develder, M. Pickavet, P. Demeester, “Routing strategies to minimize packet loss in an optical packet switched network with recirculating FDL buffers”, accepted for PNET.

[I] C. Develder, D. Colle, S. Demaesschalck, M. Pickavet, P. Demeester, “Influence of GMPLS recovery mechanisms on TCP performance”, Photonic Network Commun. (PNET), Vol. 4, No. 3/4, Jul.-Dec. 2002, pp. 321–345.

Appendix A

Architectures for optical packet and burst switches*

C. Develder, J. Cheyns, E. Van Breusegem, E. Baert, D. Colle, M. Pickavet, P. Demeester

Invited paper for 29th European Conference on Optical Communications (ECOC2003), Rimini, Italy, 21-25 Sep. 2003.

Abstract: Optical Packet Switching (OPS) and Optical Burst Switching (OBS) are optical networking concepts based on fully exploiting fast dynamic allocation of (D)WDM channels. Compared to more static approaches focusing on providing end-to-end optical channels, OPS and OBS aim at providing greater bandwidth efficiency, granularity and flexibility. In this paper we give an overview of the issues arising when designing an optical switch for either OPS or OBS, including packet formats, contention resolution techniques, and switching fabrics.

A.1 Introduction

The deployment of (D)WDM is generally adopted to satisfy the ever-increasing demand for bandwidth. Current standardisation efforts (GMPLS [1] in the IETF, ASON related work in ITU) address the move from the currently predominant point-to-point systems, to real optical networking supporting circuit-switched optical

* This work has been supported by the European Commission through the IST-projects DAVID (IST-1999-11387) and STOLAS (IST-2000-28557), and by the Flemish Government through the IWT GBOU-project "Optical Networking and Node Architec-tures". C. Develder and Jan Cheyns are supported as a Research Assistant of the Fund for Scientific Research – Flanders (F.W.O.–Vl.), Belgium. Erik Van Breusegem and Didier Colle acknowledge the IWT for their PhD resp. post-doctoral grants.

paths. Longer term strategies for optical networking that exploit the full potential of optical switching technologies, and thus the fast dynamic optical channel allocation, are Optical Packet Switching (OPS) and Optical Burst Switching (OBS).

Note that also hybrid architectures for optical switching can be devised, in order to combine the best of both worlds: limit resources needed for packet processing, and still adhere to efficient use of the available bandwidth. A possible approach is ORION: Overspill Routing in Optical Networks [2].

This paper discusses the issues in either OPS or OBS architectures. In the next section, we discuss alternatives for the packet format. We continue with a high-level view of the switch architecture, focusing on the different phases in packet processing. The subsequent section deals with the actual switching fabric. Next, we briefly discuss contention resolution. Before concluding, we present approaches to service differentiation.

A.2 Packet Format

The crux of any packet switching concept, including OPS and OBS, is to take full advantage of the available resources by only occupying bandwidth when there effectively is data to be sent. To this end, data is packetized: a chunk of data is assembled as payload, and a preceding header is added, containing at least information on the payload's destination. Transporting the header can be done in various ways: it can be (i) sent in-band just before the payload, using the same wavelength, (ii) transmitted on an orthogonal channel, e.g. through FSK-based modulation, or (iii) provided on a separate control channel. The main advantage of the latter is that control information and payload are physically separated, thus facilitating independent processing (e.g. electronic vs. optical). Yet, it calls for precise synchronization between data (payload) and control (header) channels.

Towards the implementation of packet switching in optics, two fundamentally different approaches exist: one can either opt for fixed length optical packets (necessitates fragmentation and reassembly functions interfacing to the client layers), or for variable length packets. The network can be operated in either a time-slotted manner, or rather an asynchronous mode. Usually the slotted approach is taken for fixed length packets, whereas the asynchronous operation is adopted in case of variable length packets. These different approaches are illustrated in Figure A.1 for in-band headers and single-wavelength packets. Note that some approaches spread the header info over multiple wavelengths, and jointly switch a whole waveband.

OPS usually denotes the use of fixed length packets, in a slotted mode of operation, whereas OBS [3] uses variable length packets. In addition, the granularity of OBS is taken to be coarser than OPS. Compared to OBS, the advantages of OPS are that due to its finer granularity it allows a more efficient bandwidth usage, and that logical performance (e.g. Packet Loss Rate, PLR) of switches operating in slotted mode is better than unslotted ones. Clearly, there is a reverse side of the medal: the finer granularity implies a larger amount of overhead (cf. less data per header) and the need for faster switches, while the slotted operation requires synchronization of the different inputs of the switch.

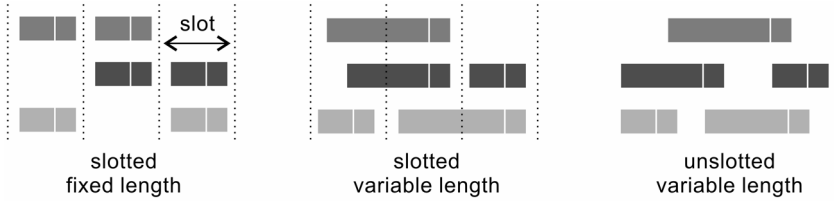


Figure A.1: Illustration of the various options for packet switching: fixed length packets, variable length packets, either in slotted or unslotted mode of operation.

Note that even with a slotted architecture, variable length packets can be dealt with (see middle part of Figure A.1). The same architecture as used for fixed, slotted packet switching can be used when segmenting the packet into a train of slots and either (i) adopt a wagon approach by treating the wagons individually, or (ii) treat the train as a whole and make switching decisions for the complete train upon arrival of its "locomotive". In [4], we showed that the wagon approach may lead to lower data loss rates if there is a buffer and trains are relatively short. However, we found this to hold only for a limited range of loads.

A.3 Node Architecture

A generic view of the architecture for an optical packet switch comprises three stages [5]: an input interface, a switching core, and an output interface. Alternatives for the switching fabric are discussed in the next section.

The input interface will at least have to provide extraction of the packet headers for appropriate processing, as the header will dictate the routing of the payload. In case of in-band headers, this implies that the different wavelengths will need to be demultiplexed, since the headers of different packets need to be isolated. To date, most architectures proposed by various projects dealing with optical packet switching foresee O/E/O conversion of the headers: they will be processed electronically to set and control the optical devices, esp. those in the switching stage. This control and routing in the resulting network can be based on Generalized Multi-Protocol Label Switching (GMPLS), e.g. in [6]. To avoid elaborate O/E/O header conversion, all-optical header processing techniques recently have been proposed, e.g. [7]. The payload, which does not need processing, can be kept in the optical domain, and thus be transparently transported from in- to outputs. Also note that the header's bitrate can be different (much lower) than that of the payload.

In the final output stage, packets destined for a same outgoing fibre will be multiplexed. Also, in this stage the packet headers may be (re-)written. Indeed, depending on the routing mechanism and accompanying header info, the header could need to be updated. In this case, the input stage could strip the original header and the output stage could add the new one. If the switch stage does not affect the channel on which the headers are transported, the new header clearly could already be inserted at the end of the input stage. When the switching stage would degrade

the signal too much (e.g. due to crosstalk), the output stage needs to include regeneration, possibly 3R, preferably all-optical.

A.4 Switching Matrix

The core functionality of an optical packet switch is to selectively transmit packets from a particular input port to a particular output port. Here, a "port" implies a certain wavelength on a certain fibre. Three well-known optical switching approaches are: micro-electromechanical systems (MEMS, [6]), a broadcast-and-select architecture, and an Arrayed Waveguide Grating (AWG). Since MEMS suffer from slow switching times (ms), they are not suitable for packet switching.

The broadcast-and-select architecture (B&S) has been proposed e.g. in the European research projects KEOPS [8] and DAVID [9]. A simplified view of the switching fabric proposed by the latter is depicted in Figure A.2 for 16 input fibres with each 16 wavelengths [10]. The first stage multiplexes different wavelengths into a single fibre, and jointly amplifies them to compensate for the subsequent power splitting stage. For each output wavelength, two switching stages are foreseen: the first selects one of the 16 input fibres, and thereof the second selects a single wavelength. Advantages of such a B&S architecture are that it's non-blocking, and that it can perform multicasting.

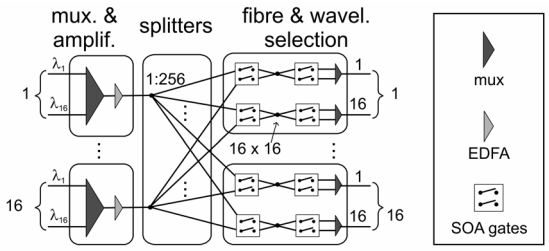


Figure A.2: A broadcast-and-select architecture as proposed in DAVID.

Another optical switching technique is based on a passive component: the AWG. This is an approach taken by e.g. the WASPNET [11] project, and the more recent STOLAS [12]. The wavelength of a signal offered at one of the AWG's input ports determines via what output port it will leave the AWG. Thus, through using tuneable wavelength converters at the inputs, an AWG can be used as a switching fabric. If the AWG is used for F fibres, each carrying W wavelengths, then in principle we can operate the switch with wavelength converters ranging only over the W wavelengths in use (assuming F<W). Unfortunately, the resulting switch then is a blocking one: there is no guarantee that all packets can be forwarded to a certain output fibre, even if we have only W (or less) packets to switch to each of the output fibres. To minimize the blocking probability, ingenious combination of the AWG's outputs into single fibres is needed [13], or at least intelligent scheduling needs to be applied to limit loss rates [14,15]. The node can be made non-blocking by using converters tuneable over F·W wavelengths. In this case, additional wavelength

converters at the outputs are needed (or F·W wavelengths will be present on a single fibre, of which only max. W will carry a packet).

With the advent of DWDM, the number of wavelengths on a single fibre has significantly increased. This means that the dimensions of the switching fabrics in the core of optical packet switches need to be huge as well. Therefore, multi-stage switching fabrics will need to be devised. This problem has been solved for circuit-switching by e.g. Clos. A three-stage Clos-network is depicted in Figure A.3. The number of intermediate stages required differs whether the structure is intended for slotted, fixed length OPS, or rather unslotted OBS: in the latter case one needs $k \geq 2n-1$ intermediate stage switches, whereas $k=n$ suffice for OPS. For an AWG switch, using multiple stages also reduces the tuneability range needed for wavelength converters.

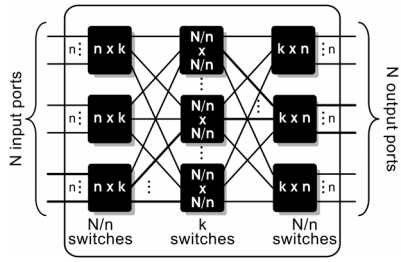


Figure A.3: A multi-stage Clos network for large switch dimensions.

By optimizing the choices for k and n, cost savings in terms of switching components needed (eg. SOAs when using the switch matrix depicted in Fig. 2) can be quite impressive. If wavelength conversion is adopted as a means of contention resolution (see further), the third switching stage can be eliminated and replaced by a set of wavelength converters. A study of the reduction in number of switching elements for the SOA-based architecture showed that for slotted OPS, the multistage architectures need less than half the number of SOAs for reasonable switch dimensions [16].

A.5 Contention Resolution

A major issue that needs to be resolved in any form of packet switching is contention resolution. In the case of optical packet switching, three approaches (or any combination thereof) can be adopted: (i) use deflection routing, (ii) provide buffering, and/or (iii) exploit the wavelength domain and use wavelength conversion. Note that providing buffering in the optical domain requires Fibre Delay Lines (FDLs), unless we can afford to convert it to and from electronics [17] (which may be the case in exceptional cases, e.g. for low priority traffic). From a performance point of view, exploiting wavelength conversion greatly lowers the PLR, as shown in e.g. KEOPS work [8]. Deflection is only effective when the network is not too highly loaded, such that there is enough free bandwidth available

along deflected routes. A logical performance comparison of the approaches can be found e.g. in [18].

The use of buffering has a major impact on the switch structure. From an architectural viewpoint, FDL buffers can be classified into either feed-forward or feed-back. Feed-forward buffers comprise input- and output-buffering schemes, whereas feed-back refers to a recirculating buffer: some of the switching fabric's output ports are connected through a FDL back to the input ports. Also, one can distinguish between single-stage and multi-stage FDL architectures. In feed-back buffers, usually a single buffering stage is used: the multi-stage approach, using multiple switching stages, is limited to feed-forward schemes. For feed-back configurations, the parameters are the number of switching fabric ports sacrificed to buffering, and the length of the recirculating FDL(s). When using multiple FDLs with different lengths, it is clear that lower PLRs are reached due to the larger buffer capacity [19]. The downside of different FDL lengths is that it is no longer possible to guarantee that (without sophisticated scheduling) packet reordering will not occur.

A.6 Service differentiation

Next generation optical networks need to fulfil the task of today's networks and continue carrying an amalgam of traffic types. Providing service differentiation will help to satisfy the various QoS requirements.

Multiple approaches to providing QoS in an IP-over-WDM scenario have been proposed [20]. Resource reservation for instance is another way to successfully achieve different service performance levels: wavelength converters or buffers can be dedicated to a single class of service, where the higher priority class gets more allocated resources. This can be done in either a static or dynamic way.

However, QoS differentiation can also be achieved without any resource reservation mechanism. In case of slotted OPS, we have demonstrated that a quite straightforward priority mechanism can achieve strong differentiation in terms of loss rates [19].

For OBS, a well-known technique to achieve differentiation is to give high priority data bursts a longer offset [21]. We have compared this approach with two alternatives in [22]: a look-ahead approach where packets are delayed at the inputs allowing low priority packets to be pre-empted by later arriving high priority ones, and a slotted control approach. The look-ahead approach performed similarly to OBS with differentiated offsets, and attained strong separation.

In a slotted approach with variable length packets however, the simple priority mechanism proved to be far less efficient in terms of differentiating the loss rates [22]. Still, by resorting to the "wagon" approach discussed earlier, and segment a variable length packet into slotted wagons, differentiation in terms of data loss rates can be significantly improved [4], as illustrated in Figure A.4 (note the much lower loss rates for high priority traffic for the "wagon" approach).

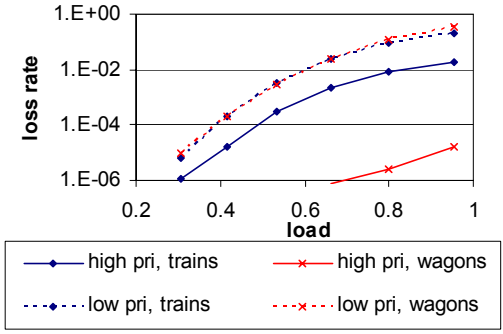


Figure A.4: Service differentiation for variable length packets in slotted OPS switches: segmenting packets into fixed size slots and treating them individually ("wagons") results in much stronger service differentiation than treating the packets as a whole ("trains").

A.7 Conclusion

OPS and OBS are packet switched approaches to optical networking, requiring fast switches. In this paper we have discussed various possible architectures for such switches, tackling packet formats, switching fabrics, contention resolution schemes and mechanisms to attain service differentiation. We summarised the pros and cons of the approaches in a qualitative manner, but also referred to results of performance studies.

References

[1]. D. Colle, et al., "GMPLS Extensions for Supporting Advanced Optical Networking Technologies", accepted for ICTON 2003, Warsaw, Poland, 29 Jun. – 3 Jul. 2003.

[2]. E. Van Breusegem, J. Cheyns, D. Colle, M. Pickavet, P. Demeester, "Overspill Routing In Optical Networks: a new architecture for future-proof IP over WDM networks", accepted for OptiComm 2003, Dallas, TX, 15-18 Oct. 2003.

[3]. C. Qiao, M. Yoo, "Optical Burst Switching (OBS) – a New Paradigm for an Optical Internet", J. High Speed Networks, vol. 8, no. 1, Jan. 1999.

[4]. C. Develder, M. Pickavet, P. Demeester, "On trains and wagons: switching variable length packets in a slotted OPS network", accepted for COIN/ACOFT 2003.

[5]. D.K. Hunter, I. Andonovic, "Approaches to Optical Internet Packet Switching", IEEE Comm. Mag., vol. 38, no. 9, Sep. 2000, pp. 116-120.

[6]. G.S. Kuo, ed., "Optical Switching", IEEE Comm. Mag., vol. 40, no. 3, March 2002.

[7]. K. Kitayama, K. Onohara, M.Murata, "Capability of optical code-based MPLS (OC-MPLS)", Proc. ONDM 2002, Torino, Italy, Feb. 2002.

[8]. C. Guillemot et al., "Transparent Optical Packet Switching: The European ACTS KEOPS Project Approach", J. of Lightwave Techn., vol. 16, no. 12, Dec. 1998.
[9]. L. Dittman (ed.), et al., "The European IST Project DAVID: a Viable Approach towards Optical Packet Switch-ing", accepted for JSAC.
[10]. D. Chiaroni, et al., "First demonstration of an asynchro-nous optical packet switching matrix prototype for MultiTera-bitclass routers/switches", Proc. of ECOC 2001, Amsterdam, The Netherlands, Oct. 2001.
[11]. D. Hunter, et al., "WASPNET: A Wavelength Switched Packet Network", IEEE Comm. Mag., vol. 37, no. 3, March 1999.
[12]. K. Vlachos, et al., "An Optical IM/FSK Coding Technique for the Implementation of a Label-Controlled, Arrayed Waveguide Packet Router", to appear in J. of Lightw. Techn., Nov. 2003.
[13]. J. Ramamirtham, J. Turner, "Design of Wavelength Converting Switches for Optical Burst Switching", Proc. Infocom 2002, New York City, NY, 23-27 Jun. 2002.
[14]. J. Cheyns, et al., "Routing in an AWG-based optical packet switch", Photonic Netw. Commun., vol. 5, no. 1, Jan. 2003.
[15]. J. Cheyns, et al., "Performance Improvement of an Internally-Blocking Optical Packet/Burst Switch", Proc. ICC 2003, Anchorage, AK, 11-15 May 2003.
[16]. J. Cheyns, J. Jennen, E. Van Breusegem, M. Pickavet, P. Demeester, "Optical packet switches based on a single AWG", Proc. HPSR 2003, Torino, Italy, 24-27 June 2003.
[17]. C. Develder, J. Cheyns, M. Pickavet, P. Demeester, "Multistage architectures for optical packet switching using SOA-based broadcast-and-select switches", Tech. Digest OFC 2003, Atlanta, GA, 23-28 Feb. 2003.
[18]. S. Bjornstad, D.R. Hjelme, N. Stol, "An optical packet switch design with shared electronic buffering and low bit rate add/drop inputs". Proc. ICTON 2002, Warsaw, Poland, 21-25 Apr. 2002
[19]. S. Yao, B. Mukherjee, S. J. Ben Yoo, S. Dixit, "All-Optical Packet-Switched Networks: A Study of Contention-Resolution Schemes in an Irregular Mesh Network with Variable-Sized Packets", Proc. SPIE Vol. 4233, (OPTICOMM 2000, Plano, TX.) Oct. 2000.
[20]. C. Develder, M. Pickavet, P. Demeester., "Strategies for an FDL Based Feed-Back Buffer of an Optical packet Switch with QoS Differentiation", Proc. COIN 2002, Cheju, Korea, 21-25 Jul. 2002.
[21]. A. Kaheel, T. Khattab, A. Mohamed, H. Alnuweiri, "Quality-of-service mechanisms in IP-over-WDM networks", IEEE Comm. Mag., vol. 40, no. 12, Dec. 2002.
[22]. M. Yoo, C. Qiao, "Supporting multiple classes of services in IP over WDM networks", Proc. Globecom 1999, vol. 1b.
[23]. C. Develder, J. Cheyns, M. Pickavet, P. Demeester, "Service differentiation for OBS in a switch with recirculating FDL buffer", submitted to PS 2003.

Appendix B

Active versus passive OPS architectures for metro rings: a network dimensioning point of view*

C. Develder, R. Van Caenegem, E. Baert, M. Pickavet, P. Demeester

Accepted for 29th European Conference on Optical Communications (ECOC2003), Rimini, Italy, 21-25 Sep. 2003.

Abstract: Optical Packet Switching is a promising technology for metro environments. We discuss two ring architectures (with/without active components allowing for spatial reuse) and compare them in terms of resources required for a given traffic demand.

B.1 Introduction

Next generation metro area networks (MANs) should provide high bandwidth in a flexible manner: they should efficiently exploit available resources, support multiple traffic types and offer rapid provisioning. Optical Packet Switching (OPS), with its packet-level granularity and hence efficient and flexible bandwidth sharing, fulfils these requirements very well [1].

* This work has been supported by the European Commission through the IST-project DAVID (IST-1999-11387), and by the Flemish Govern-ment through the IWT GBOU-project "Optical Networking and Node Architectures". C. Develder is supported as a Research Assistant of the Fund for Scientific Research – Flanders (F.W.O.–Vl.), Belgium

In the European DAVID project [2], multiple MAN architectures are compared. Here, we outline the DAVID metro ring architecture and discuss two different MAN optical packet add/drop multiplexer (OPADM) designs: a Passive one, and an Active one. This paper focuses on the impact of these design choices on the resources needed to build a MAN network interconnecting a given set of nodes, with a given traffic demand from one node to another.

B.2 MAN ring architectures

In the DAVID concept, sketched in Figure B.1, the MAN comprises slotted WDM rings collecting traffic from several optical packet add/drop multiplexers (OPADMs). Rings are interconnected by a buffer-less Hub, which also provides access to a backbone (WAN). The rings constitute a shared medium, requiring a medium access control (MAC) protocol [3] to arbitrate access to the slotted channels. One wavelength, λc, is a dedicated control channel.

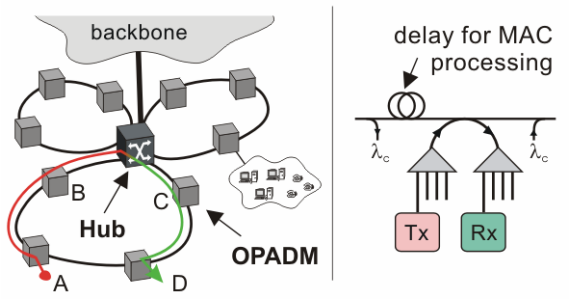


Figure B.1: Network architecture and Passive OPADM

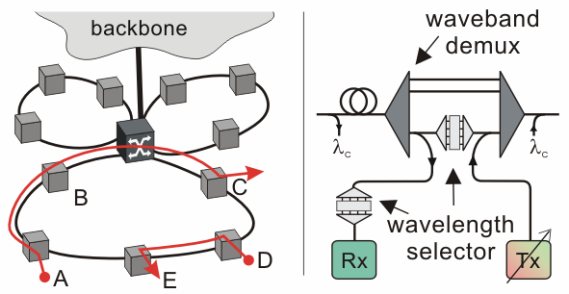


Figure B.2: Network architecture and Active OPADM

DAVID proposes two OPADM architectures. The first one limits the use of advanced optical technologies, choosing commercial and mature ones instead [4]: it uses couplers and off-line filters to minimize physical cascadeability issues. The structure of this Passive OPADM is depicted in Figure B.1. The wavelength spectrum is separated for upstream (transmitters, Tx) and downstream (receivers, Rx): the Hub will perform conversion from Tx to Rx spectrum.

The second, Active OPADM proposal of Figure B.2 —considered as longer term approach— allows an incoming packet to be erased from the ring, and to replace it with a new one. Because of this erasing capability, there is no need for spectral separation of Rx and Tx signals. This also allows for spatial reuse: whenever the path from source to destination does not cover the whole ring, the same wavelength can be re-used, as for A-C and D-E in Figure B.2. To limit the tuneability range of the Rx/Tx elements, a waveband concept is introduced: a Rx/Tx board provides access to a set of only B wavelengths (with one Rx/Tx per band).

B.3 A network dimensioning point of view

The objective of this paper is to compare the architectures in terms of the amount of resources (which will to a great extent dominate the CAPEX) required to set-up a given demand between a given set of MAN nodes. Therefore, we developed a network planning algorithm starting from an ILP-formulation of the planning problem. Yet, the many degrees of freedom hamper the finding of ILP solutions within reasonable time. Hence, we provided heuristic solutions using a tabu-search approach to find the minimal number of resources needed to fulfil a given traffic demand.

The cost indicators used are the following: (i) Rx/Tx capacity: the total number of Rx/Tx elements used, summed over all OPADMs, (ii) link capacity: the number of wavelengths effectively used per link, summed over all physical links, (iii) nr. of lambdas: the number of wavelengths used per ring, summed over all rings. The first criterion is an indicator of the OPADM costs, while the last will impact the Hub dimension and thus its cost.

Note that this dimensioning study is only a single (but quite important) facet of an in-depth assessment of the pros and cons of Active and Passive architectures. This paper therefore is to be complemented with e.g. studies on the architectures' capabilities to deal with dynamic traffic in a network with given amount of resources, as eg. in [3].

B.4 Set-up of the case study

To assess the resource requirements of the OPADM architectures, we covered a wide range of demand patterns. The demand patterns are the following (where D[i,j] denotes the bandwidth required between OPADMs i and j): (i) Uni: a uniform demand pattern, where between each two OPADMs a bandwidth d needs to be set-up (D[i,j]=d); (ii) Serv: there is one server node s, which dominates the demand matrix (D[i,s]=D[s,i]=2d, other D[i,j]=d); (iii) Neigh3: each node only communicates to 3 other nodes (D[i,i+1]= D[i,i+2]=D[i,i+3]=d, rest is zero); (v) David: a demand matrix based on real-life traffic, provided by the operators participating in DAVID.

B.5 The impact of space reuse

The main difference between the active and passive architectures from a conceptual point of view is the space reuse capability of the Active structure. Figure B.3 presents dimensioning results of the dimensioning for Passive and Active with wavebands of a single wavelength. (Note that B=1 amounts to having no waveband concept; B>1 is discussed in the next section.)

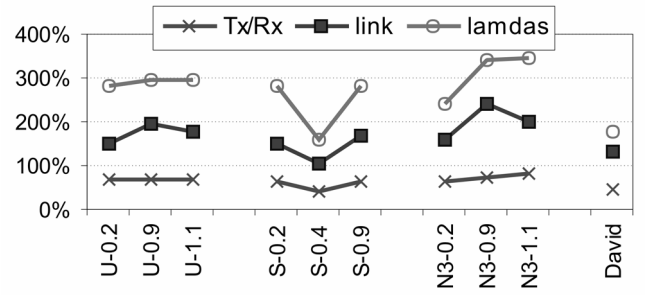


Figure B.3: Cost ratio Passive/Active; x-axis labels denote demand (U=Uni, S=Serv, N3=Neigh3; the number after the dash is the value of d).

From a Rx/Tx cost perspective, we conclude that the Active approach needs more Rx/Tx capacity. The reason is that to allow spatial reuse, the Rx and Tx have to be able to access the same wavelength, which sometimes requires an extra Rx/Tx (cf. B=1 means no tuneability).

The spatial reuse concept only proves useful when the CAPEX of the MAN is dominated by the link capacity, or the number of wavelengths per ring. This is due to the fact that there is no spectral separation for up- and downstream, and the spatial reuse capability allows for better sharing of the available bandwidth among different demands.

B.6 Wavelength bands

A second aspect in which the Active and Passive structures differ is the waveband concept. In the previous section, we used wavebands of a single wavelength (ie. no tuneability in the OPADMs). In this section we study the impact of introducing the waveband concept, again from a network dimensioning point of view. We compare the Active nodes with B=1 versus B=4 in Figure B.4.

The advantage of the band concept is that Rx/Tx capacities can be somewhat reduced. Yet, when CAPEX is dominated by link capacities, the band concept is not useful, since it heavily increases the number of wavelengths used, indicating that spatial reuse opportunities within bands are limited. This stems from the fact that the architecture is assumed to allow only a single Rx/Tx per band per OPADM.

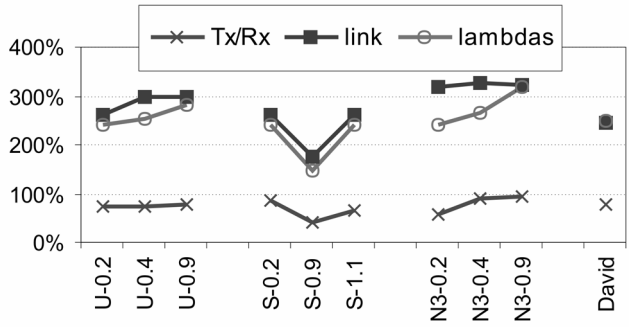


Figure B.4: Cost ratio (bands, B=4)/(no bands, B=1); x-axis labels denote demand.

B.7 Conclusions

We have discussed two optical packet switched OPADM structures, and considered the impact of the architecture on the resources needed to fulfil a particular demand. Our results show that only when the amount of wavelengths used highly affects the network cost, the advanced active node structure should be deployed. In that case, a waveband concept does not seem to be appropriate if only a single Rx/Tx is allowed per OPADM per waveband.

References

[1] S. Yao, et al., "All-optical packet switching for metropolitan area networks: opportunities and challenges", IEEE Comm. Mag., 39(3), pp. 142–148.

[2] L. Dittmann (ed.), et al., "The European IST project DAVID: a viable approach towards optical packet Switching", to appear in JSAC.

[3] A. Bianco, et al., "Access control protocols for interconnected WDM rings in the DAVID metro network", in Proc. IWDC 2001.

[4] N. LeSauze, et al., "A novel, low cost optical packet metropolitan ring architecture", Proc. ECOC 2001.

Appendix C

Multistage architectures for optical packet switching using SOA-based broadcast-and-select switches

C. Develder, J. Cheyns,
M. Pickavet, P. Demeester

Tech. Digest Optical Fibre Conference (OFC 2003), paper FS4, Atlanta, GA, USA, 23-28 Feb. 2003, pp. 794–795.

Abstract: Optical packet/burst switches can be scaled to high port counts using multi-stage architectures. To reduce the number of switching elements in SOA-based broadcast-and-select architectures, we deploy only a few stages while exploiting the WDM dimension.

OCIS-codes: (060.0060) Fiber optics and optical communications, (060.4250) Networks, (060.1810) Couplers, switches, and multiplexers

C.1 Introduction

To satisfy the ever-lasting bandwidth hunger, (D)WDM is adopted. Long-term strategies for real optical networking, that will eventually replace the currently predominant point-to-point systems, envisage Optical Packet Switching (OPS) and/or Optical Burst Switching (OBS). Those approaches fully exploit the potential of advanced optical switching technologies providing fast dynamic optical channel allocation. One such approach is proposed by the European research project DAVID [1], which aims at proposing viable approaches towards optical packet switching. It uses a broadcast-and-select switching matrix based on SOA technology [2].

In this paper, we illustrate how a multi-stage architecture can be used to scale OPS or OBS switches to high port counts. Section C.2 introduces the two most wide-spread node architectures for OPS/OBS and highlight the factors that limit their scalability. In Section C.3 we review the three-stage Clos architecture, and subsequently show how the third switching stage can be eliminated by using the WDM dimension through wavelength convertors. In Section C.4 we present a case study for the DAVID architecture showing the advantage of the two-stage architecture in terms of number of switching elements (SOA gates). The final Section 0 concludes the paper.

C.2 Node architectures for OPS/OBS

The core functionality of an optical packet switch is to selectively transmit packets from a particular input port to a particular output port. Here, "port" implies a certain wavelength on a certain fibre. For the concept of "packet" switching, two fundamentally different approaches exist: one can either opt for fixed length optical packets, or for variable length packets. The network can be operated in either a time-slotted manner, or rather an asynchronous mode. Usually the slotted approach is taken for fixed length packets, whereas the asynchronous operation is adopted in case of variable length packets. We choose to reserve the term OPS for a fixed-length packet switch using a slotted operation, whereas OBS clearly is a case of unsynchronized switching of variable length packets.

C.2.1 Switch fabric architectures

An optical packet switch from a generic viewpoint comprises three parts [3]: an input interface, a switching fabric, and an output interface. To provide the core function of switching packets from inputs to outputs, two architectures dominate the OPS/OBS approaches proposed in recent publications and research projects: (i) broadcast-and-select architectures, (ii) Arrayed Waveguide Grating (AWG).

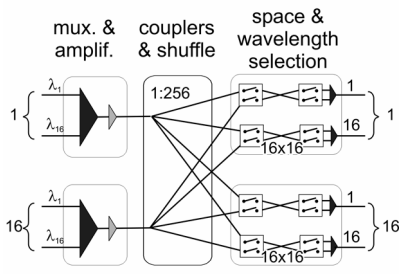


Figure C.1: A broadcast-and-select architecture as proposed in DAVID

The broadcast-and-select architecture (B&S) has been proposed in e.g. the European research projects KEOPS and DAVID. A simplified view of the switching fabric proposed by the latter is depicted in Figure C.1 [2]. The first stage multiplexes different wavelengths into a single fibre, and jointly amplifies them to compensate

for the subsequent power splitting stage. For each output wavelength, two switching stages are foreseen: the first selects one of the F input fibres, and thereof the second selects a single wavelength among the W available ones. Advantages of such a B&S architecture are that it is non-blocking, and that it can perform multicasting.

The optical switching technique based on an AWG is an approach taken by e.g. the WASPNET project, and the more recent STOLAS [4]. The wavelength of a signal offered at one of the AWG's input ports determines via what output port it will leave the AWG: through using Tuneable Wavelength Converters (TWCs) at the inputs, an AWG can be used as a switching fabric. To construct a switching fabric for F fibres, each carrying W wavelengths, in principle the tuneable wavelength converters have to range only over the W wavelengths in use. Unfortunately, the resulting switch then is a blocking one, meaning that there is no guarantee that all packets can be forwarded to a certain output fibre, even if we have only to switch W (or less) packets to each of the output fibres. However, a non-blocking fabric is obtained by using converters tuneable over F×W wavelengths. In this case, additional wavelength converters (with fixed output wavelength) at the outputs are needed.

C.2.2 Scalability

Both the B&S and AWG architectures have limited scalability. The factor that limits the maximal port count for the B&S architecture is the splitting ratio: each incoming signal is split in the broadcast stage over each of the $N = F \times W$ output ports. For the AWG-based approach, the number of output ports is limited by the tuneability range of the TWCs, since they have to be tuneable over as many wavelengths as there are output ports ($N = F \times W$). A possible solution is the adoption of multi-stage architectures, as discussed in the next section.

C.3 Multi-stage architectures

In the early days of network engineering, Clos proposed a multi-stage architecture to construct large switches based on building blocks with smaller dimensions [5]. A sketch of an N×N switching architecture using a three-stage Clos network is outlined in Figure C.2. The three stages comprise a first with N/n switches of dimension n×k, a second with k matrices (N/n) × (N/n), and a last with again N/n switches, now of size k×n.

To be non-blocking, a lower bound on k is imposed: $k \geq 2n-1$. This minimal value for k can easily be determined as follows: consider a connection between input ports A and B. As illustrated, the worst case occurs when (i) all n–1 other ports of the first-stage to which A belongs are already occupied and connected to some output ports via n–1 switches in the second stage; and (ii) the (n–1) remaining ports of the third-stage switch with B also are in use and come from n–1 other switches in the intermediate stage. Thus, to be non-blocking the switch needs at least (n–1)+(n–1)+1 connection points, hence $k \geq 2n-1$. Typically, it is convenient to set k=2n. This boundary was determined by Clos for circuit-switched networks, implicitly

assuming that connections cannot be rerouted along other switches in the intermediate stage once it has been set-up.

Unlike Clos, we focus on OPS, operating in slotted mode. In this case, the bound for k is that of a rearrangeable non-blocking switch, which is a lot smaller: k≥n. The proof why k=n suffices is a well-known result from graph theory: the problem of finding the necessary number of second stage switches is equivalent to a graph coloring problem in a bipartite graph with degree n, which is n-colorable.

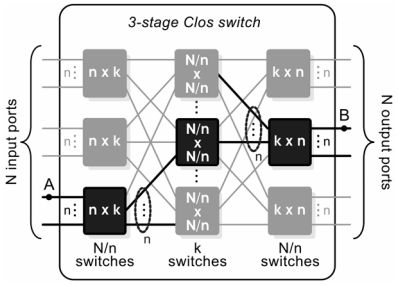


Figure C.2: Three-stage Clos architecture.

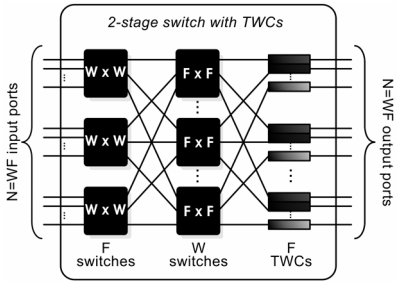


Figure C.3: Two-stage architecture with TWCs.

In case of OPS, all building block switch fabrics have symmetrical dimensions. If all wavelengths within a fibre may be considered equivalent, N input/output ports are grouped per W wavelengths into F=N/W fibres, then we can eliminate the third switching stage switches if we choose n=W. The third "stage" then simply consists of W wavelength converters with a fixed outgoing wavelength, as depicted in Figure C.3. Note that the choice of n=W will not necessarily be the optimal choice to contruct a (F·W)×(F·W) switching fabric using the classical Clos approach; however by eliminating the third stage switches, the overall cost can be reduced. In the case study in the next section, we illustrate this for the DAVID architecture. Note that we have not considered architectures of more than three stages: esp. for the B&S architectures, the cascade of multiple switching stages would imply a too severe signal degradation (cf. we want to avoid regeneration stages within the switching fabric).

C.4 Case study: limiting the number of SOAs in the DAVID architecture

In the previous sections, we discussed the DAVID architecture for the switching fabric, and illustrated how a multi-stage architecture can be used to construct large switch fabrics. In this section, we compare the different architectures for the case of the B&S switch proposed within DAVID: (i) single stage, (ii) three-stage Clos, (iii) two-stage with wavelength converters. The cost of this architecture will be dominated by the number of SOA components, thus the comparison will focus on the number of SOA gates needed to implement the switch.

The architecture of the DAVID switching fabric was discussed earlier and sketched in Figure C.1. The number of SOA gates needed to construct a single-stage N×N switch is given in eq. (1): for each of the N output ports, N/w gates are needed for space selection, while w gates are needed for wavelength selection. Since the switching matrix will be surrounded with wavelength converters (actually 3R regenerators, see [2]) the number of wavelengths w can be optimized (and chosen different from w, the number of wavelengths on the input/output fibres) to minimize the number of SOA gates. The optimal choice is w=N1/2, which leads to the minimal number of SOA gates for a single-stage switch as given in eq. (2).

$$s_{sym}(N,w) = N \cdot \left((N/w) + w \right) \tag{1}$$

$$s_{sym,opt}(N) = 2 \cdot N \cdot \sqrt{N} \tag{2}$$

For OPS switches, we have indicated that the number of second stage switches needed to provide a non-blocking fabric to operate in slotted mode is k=n. The optimization of n to reduce the number of SOA gates in the overall multistage architecture leads to the choice n=0.5·N1/2, see eqs. (3 4).

$$\begin{aligned} c_{slotted}(N,n) &= N/n \cdot s_{sym,opt}(n) + n \cdot s_{sym,opt}(N/n) + N/n \cdot s_{sym,opt}(n) \\ &= 2 \cdot N \cdot \left(2 \cdot \sqrt{n} + \sqrt{N} \cdot \frac{1}{\sqrt{n}} \right) \end{aligned} \tag{3}$$

$$c_{slotted,opt}(N) = 4 \cdot \sqrt{2} \cdot \sqrt[4]{N} \cdot N \tag{4}$$

In case the proposed two-stage architecture, the number of SOA gates needed is given by eq. (5).

$$t_{slotted}(F,W) = F \cdot s_{sym,opt}(W) + W \cdot s_{sym,opt}(F) = 2 \cdot F \cdot W \cdot \left(\sqrt{W} + \sqrt{F} \right) \tag{5}$$

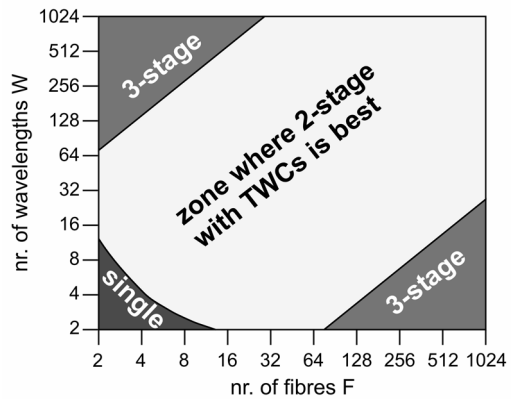


Figure C.4: Boundaries of the regions where each of the three node architectures is the most advantageous one.

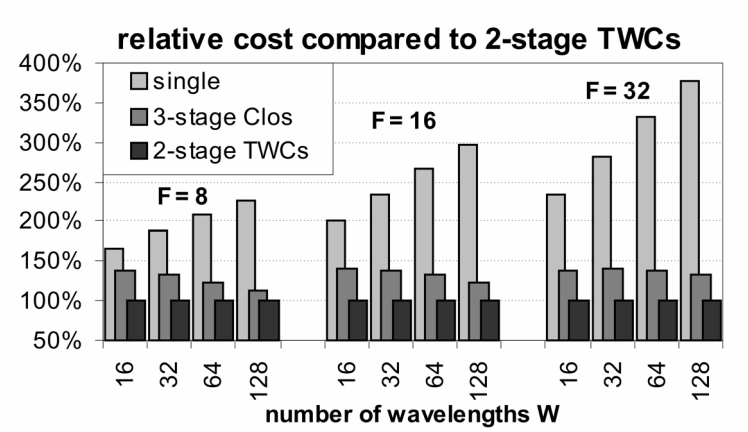


Figure C.5: Comparison of the three architectures for F=8,16,32 input/output fibres and W=16,32,64,128 wavelengths per fibre.

Inspection of the number of SOA gates needed leads to the choices as illustrated in Figure C.4. From this comparison, it is clear that the single stage architecture is only advantageous for small port counts ($N=W\cdot F<12$). The two-stage architecture with TWCs is generally the cheapest choice: the three-stage Clos architecture only can be advantageous when either the number of wavelengths per fibre or the number of input/output fibres is very large.

C.5 Conclusion

We have discussed two switching fabric architectures (broadcast-and-select, and AWG-based) and outlined how their respective scalability limits can be overcome by adopting a multi-stage approach. We have shown that in case of OPS we can exploit the WDM dimension to eliminate the last switching stage and replace it with wavelength converters. For the B&S architecture proposed in DAVID, we have shown that this two-stage architecture with TWCs is the cheapest architecture in terms of used SOA gates.

References

[1] DAVID: Data And Voice Integration over DWDM, project IST-1999-11387, http://david.com.dtu.dk

[2] D. Chiaroni, et al., “First demonstration of an asynchronous optical packet switching matrix prototype for MultiTerabitclass routers/ switches,” in Proc. 27th European Conf. on Optical Communication, (Amsterdam, The Netherlands, September 30 – October 4, 2001).

[3] C. Develder, J. Cheyns, E. Van Breusegem, E. Baert, A. Ackaert, M. Pickavet, and P. Demeester, “Node architectures for optical packet and burst switching”,

in Proc. Int. Topical Meeting on Photonics in Switching (PS2002), (invited) paper PS.WeA1, (Cheju Island, Korea, July 21–25, 2002), pp. 104–106.

[4] STOLAS: Switching Technologies for Optically LAbeled Signals, project IST-2000-28557, http://www.ist-stolas.org

[5] C. Clos, "A study of non-blocking switching networks", Bell System Technical Journal, Vol. 32, 1953, pp. 406-424.

Appendix D

Assessment of packet loss for an optical packet router with recirculating buffer

C. Develder, M. Pickavet, P. Demeester

Proc. 6th IFIP Working Conference on Optical Network Design and Modelling (ONDM2002), Torino, Italy, 4 6 Feb. 2002, pp. 247–261.

Abstract: The design of an optical packet switched network to transport both data and voice, focussing on an IP client layer, is the subject of DAVID (Data And Voice Integration over DWDM), a research project funded by the European Community. The network is composed of a metropolitan part, where optical rings are connected through devices denoted as Hubs. Those Hubs also provide the connection to the backbone part of the network, made up by SOA based optical packet routers (OPRs) interconnected in a mesh. This paper focuses on the performance evaluation of the backbone network, in casu an OPR, in terms of packet loss rate (PLR): different optical buffer implementations and scheduling strategies are proposed, through simulation the resulting PLR is determined.

Keywords: DWDM, optical packet switching, simulation

D.1 Introduction

The research project Data And Voice Integration over DWDM (DAVID), is funded by the European Community as part of the Information Society Technology (IST) Program. DAVID aims at proposing a viable approach to optical packet switching [1]. The network is conceived as an optical packet switched transport infrastructure, interconnecting IP routers and/or switches collecting traffic from legacy networks. To this end, a traffic format independent of the client traffic type will be adopted.

The network architecture that will be adopted, and demonstrated in a testbed, consists of a ring-based metropolitan area network (MAN), and a wide area backbone network (WAN). In the MAN, ring nodes are interconnected through a Hub that also provides interconnection to the backbone part of the network. This wide area backbone network (WAN) will consist of optical packet routers (OPRs), connected in a mesh via DWDM links.

This paper focuses on the WAN part of the DAVID network, more in particular the study of a single OPR: through simulation, we try to assess the performance of the OPR for different optical buffer configurations and scheduling strategies.

The rest of this paper is structured as follows: in Section D.2, we briefly describe the DAVID network architecture. The description of the adopted simulation approach is presented in Section D.3, the first results obtained with this simulation approach follow in Section D.4. The final Section D.5 concludes the paper.

D.2 The David network

Within the DAVID network, the optical packet switching approach is evaluated in both a metropolitan area and a backbone. This is reflected in the network architecture, which is based on a hierarchical interconnection of optical MAN rings to a mesh backbone WAN, as depicted in Figure D.1. The network will deploy a mixed WDMA/TDMA approach: a fibre will be carrying multiple wavelengths (up to 32 channels, at 2.5 or 10 Gbit/s) and time will be divided in fixed size slots carrying an optical packet made up of an optical header and payload. The payload part will be switched transparently through the network.

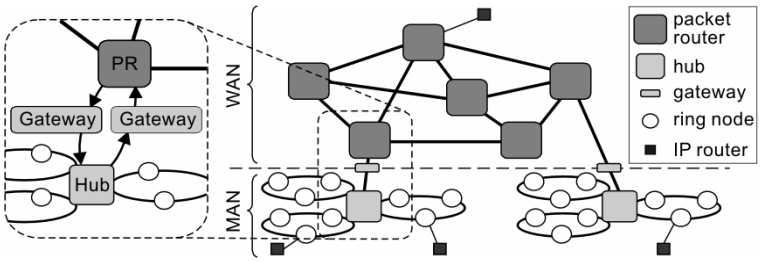


Figure D.1: The DAVID network architecture

The metro network comprises one or more uni-directional optical physical rings interconnected in a star topology by a Hub, collecting traffic from several nodes. Through a Gateway, the Hub will be connected to an OPR, thus providing the connection to the backbone WAN. The functionality of the Hub is interconnecting rings: also the connection towards the WAN (via the Gateway) logically can be seen as an extra ring to switch traffic to and from. The role of the Hub is then to switch traffic between Metro rings and to/from the WAN: basically the Hub behaves as a space switch, and will be buffer-less. Indeed, the Hub will be an all-optical device, comprising a WDM synchronisation stage, a space switching stage, a wavelength conversion stage and possibly 3R regeneration. The absence of buffering in the Hub,

and on any path between Ring Nodes not leaving the MAN, means that contention will need to be solved by means of a MAC protocol, dictating when what Ring Node can access the ring. How this MAC and the switch scheduling at the Hub may be implemented, is described and analysed in [2].

In the WAN, a mesh network will be formed where links consisting of one or more fibres will interconnect Optical Packet Routers (OPRs). This WAN will be operated in a way similar to electrical packet networks, and will adopt a hierarchical networking approach [3]. Contrary to the MAN, buffers will be present to aid in solving contention in the optical domain by means of Fibre Delay Lines (FDLs). Another means to tackle contention is the exploitation of the wavelength dimension: when two optical packets destined for the same output fibre need to be switched, they will be placed on different wavelengths. Only when this proves not to be possible, the optical FDL buffer will be addressed. These FDLs will be deployed in a recirculating buffer scheme: in DAVID, the OPR will be made up of a broadcast-and-select switching fabric based on SOA technology [4,5], where some of the wavelength ports will be connected to a buffer block of FDLs. The performance, in terms of packet loss ratio, of such an OPR for various FDL schemes and buffer scheduling strategies is the subject of the simulations discussed in the rest of this paper.

D.3 Single node performance: simulator

Before trying to evaluate node architectures on a network wide level, we need to obtain information on the performance of a single node. Therefore we have developed a simulation program to investigate the performance of an OPR in DAVID. In this section, we discuss the properties of the developed simulator.

D.3.1 Node architecture

A high level overview of the simulated node is depicted in Figure D.2, and reflects the node architecture deployed in DAVID. The core of the OPR is the switching fabric, being a broadcast-and-select switching fabric based on SOA technology (see e.g. [4]). The node has N input ports and M output ports (normally with N=M), where the number of wavelengths per fibre can be set for each port (independently of the other ports). These ports include both the add/drop ports of the OPR coming from/going to the Gateway (interfacing to the MAN) and the ports for transit WAN traffic, connecting the OPR with other nodes in the backbone. From the point of view of the switch, these ports are equivalent. Also, the wavelengths within a port are considered to be equivalent, as the wavelength domain can be used to solve contention.

The number of wavelengths going to the buffer, denoted as B, and the lengths L_i (i=0..B) of the fibre delay lines they are transmitted on, is another set of parameters. A packet entering the buffer block will appear at the outputs after a certain number of timeslots, depending on the fibre length of the FDL. Indeed, the exact time is determined at the time the packet leaves the switch matrix and is put on

one of the B wavelengths of the FDLs. More about the emplOyed buffer scheduling strategy will follow in the sections to come.

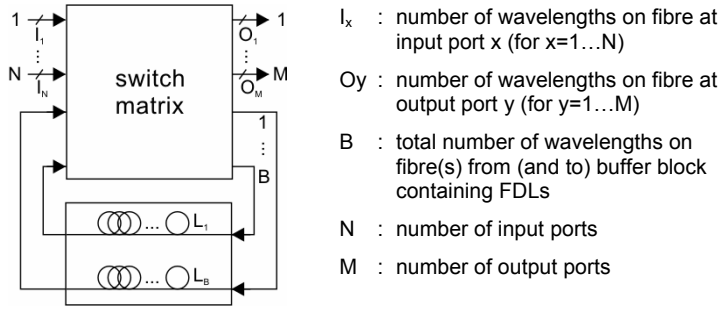


Figure D.2: The simulated node and its parameters.

D.3.2 Operation of the switch

The OPR switches optical packets of fixed length and operates in a slotted manner. Each timeslot when packets are present at the inputs, the switch matrix will forward the packets presented at the input ports (including the ports coming from the FDL buffer block) to the output ports and —if necessary— to the ports towards the buffer block of FDLs.

D.3.2.1 Packet format

To determine which packets to switch to what port, the switch makes use of certain information associated with a packet. The parameters used in our simulator are listed in Table D.1. Note that these fields will not necessarily all have a direct counterpart in a real-world implementation's packet format: some of them are merely there for tracing purposes or because they make the simulation program implementation easier.

The OPR will base its switching decisions on the following subset: src, dst, pri and *tstamp*. How these will be used is clarified below. The other fields are only used in the simulation program for the purpose of tracing and collecting statistics.

D.3.2.2 Switching operation

For each input port, the packets presented are simply forwarded to the output port they are destined for, as marked in the dst field. When more than Oy packets, say P_y, are destined for a particular output port y, preference is given to packets with higher priority. Within a particular priority class, packets that have already gone through the buffer of FDLs will be taken first. For the latter purpose, packets are marked with a timestamp (the *tstamp* field of Table D.1) the moment they enter the switch. In this way, we try to avoid recirculations of the same packets.

Thus, packets destined for a particular outgoing port will be ordered in descending priority, and within a particular priority class in ascending timestamp order. If for the highest priority class more than Oy packets remain with the same timestamp, then only Oy packets will be picked randomly (to ensure fairness).

Table D.1: Attributes associated with a packet in the simulator.

Parameter	*Meaning*
fid	flow identifier: unique number associated with traffic source that generated the packet
src	source port: this is the port (fibre) along which the packet enters the DAVID node
dst	destination port: this is the outgoing port (fibre) along which the packet will have to leave the DAVID node (unless it is dropped of course)
pri	priority: packets with higher priority will get forwarded before packets of lower priority
tstamp	timestamp: packet will be marked with a timestamp (e.g. by means of a local clock maintained at DAVID node) when it enters the switch

The remaining ($P_y - Oy$) packets that could not be sent, because of lack of wavelengths on output port y, will be buffered if possible. When the buffer is full, the remaining packets will be dropped (resulting in packet loss).

D.3.2.3 Scheduling algorithm

The strategy outlined above is implemented in our simulator using the algorithm illustrated in Figure D.3. The algorithm, which is repeated each timeslot when packets are to be sent, consists of four phases:

- **(1)** The first phase is to inspect new packets. All the packets entering the node at the same input port (fibre) are put in a queue[*]. This means we have N input queues. The sole purpose of these input queues is to ensure we will not process more than I_x packets per timeslot at input port x. Indeed, as we use different sources per (source port, destination port, priority)-triple, it may happen — depending on the traffic model used— that at a particular timeslot the total number of generated packets at input port x accidentally exceeds the number of available wavelengths on that port (even if we ensure that the total average rate is lower than I_x packets per timeslot). In fact, these input priority queues simply model the previous parts of the network where the traffic has gone through. Indeed, the traffic will be shaped already by being transported through the

[*] Note that these queues are really nothing more than an aid in implementing the desired priority scheme: packets often have to be processed in order of descending priority, and within a same priority class in order of increasing timestamp; from the set of packets within the same priority class with the same timestamp, a certain number of packets has to be randomly picked. This is exactly what the queues are used for: all packets in a priority queue will be sorted as indicated before, and the "pop" operation will randomly pick a packet from the queue belonging to the set of packets with highest priority and lowest timestamp within that priority class.

previous nodes. This means that the input priority queues are not really part of the optical packet router model, but rather model the rest of the network.

- **(2)** In the second phase, the packets are forwarded from input ports to output ports. For each input port x, at most I_x packets are taken, marked with a timestamp (based on local clock of OPR), and put in the appropriate output queues (based on the dst field of the packet). Also, the packets at the output ports of the buffer at the current timeslot are put in the appropriate output queues.
- **(3)** In the third phase, the output queues are emptied. For output port y, the first Oy packets are forwarded to the receivers. (In our simulation program, that simply means we e.g. increase the receive-counter for the fid of the packet.) The remaining packets are put in the single buffer queue.
- **(4)** The fourth phase is the buffering phase: all packets that could not be sent (because of temporary overload) are now in the buffer queue. From this queue, as many as possible packets are taken and put in a FDL. As the buffer has only B wavelengths (see Figure D.2), at most B packets can be put in the buffer. The remaining packets in the buffer queue are dropped. Which delay line to use for a particular packet, will be discussed in detail in Section D.3.3.

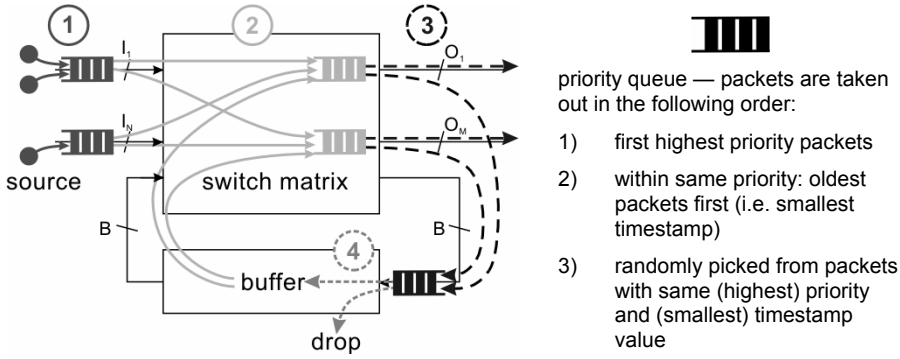


Figure D.3: Implementation of the scheduling strategy in the simulation program.

Remark that at the beginning of each cycle in this algorithm (i.e. each timeslot), the output port queues (grey in the right of the figure above), and the buffer queue (black, bottom) will be empty.

D.3.3 Buffer scheduling strategy

In step 4 of the scheduling algorithm outlined above, all packets that could not be sent directly to the output ports because of contention are joined in a priority queue. Now, for each of these packets, it has to be decided what delay to use. So far, we have provided three buffering strategies, of increasing complexity, in our simulator:

- **(a) FDL with smallest delay.** Each packet taken from the buffer queue is put in the FDL having the smallest delay that is not occupied already. When all FDLs are occupied, the remaining packets are dropped, resulting in packet loss.

- **(b) FDL with smallest delay, avoid recirculation.** In this buffer strategy, it is avoided as much as possible to have more than O_y packets (O_y being the number of wavelengths on the output port y the particular packet is destined for) leaving the buffer at the same time and heading for the same output port. This results in a buffer strategy where for each packet the buffer contents are inspected: the packet is given the smallest possible delay D, such that a free FDL with delay D can be found and the entire buffer contains less than O_y packets with the same or higher priority. If no such delay D can be found, the packet is dropped.
- **(c) FDL with smallest delay, avoid recirculation, but fully use FDLs.** This buffer strategy is the same as buffer strategy (b), except that if no suitable delay D can be found to avoid recirculations, the packet is not dropped but put in the free FDL with the smallest delay. Only if all FDLs are fully occupied, the packet will be dropped.

Strategy (a) blindly tries to minimise the delay. The advantage of the strategy is its simplicity: it just takes the first B packets and fills all the available FDLs. No information on previously scheduled packets is needed, which implies that no state information on the contents of the buffer block has to be maintained.

The second strategy (b) is more intelligent and exploits the knowledge of the buffer contents in order to try to avoid recirculation. For this strategy, it is assumed that the control logic of the OPR maintains information on the complete contents of the fibre delay lines. It avoids recirculation of the highest priority packets completely. Indeed, if a packet with priority P, destined for output Y is taken out of the buffer queue, it will be scheduled in the fibre delay line of length D iff the number of packets already scheduled for $t_{now} + D$, destined for Y and with priority greater then or equal P is less than the number of output wavelengths at port Y. At the time P leaves the buffer, it will only be recirculated (or even lost) if at that time new packets arrive with priority higher than P (cf. packets of same priority have lower precedence because they have a lower timestamp).

The third strategy (c) extends the second, by trying to avoid losses as much as possible by fully using the FDLs. Indeed, strategy (b) suffers from the fact that packets may be dropped, even if the buffer is not completely filled, and even if some buffer ports are still free. This means that we might end up with losing more packets than when we blindly put packets in the free FDL with smallest delay. It is easy to conceive an artificial situation where this is possible: e.g. consider the situation where we have only FDLs of length 1, and we have multiple input ports with traffic destined for same output port. Now, suppose that at a certain time T more than O_y sources offer a packet (where O_y is the number of lambdas at the output port). Due to the limitation of entering only a maximum of O_y packets, we will lose some packets. However, if the buffer block has more than O_y ports, we could avoid some of the losses by entering all offered packets (and recirculate them at next timeslot).

It is clear that other strategies can be devised, e.g. strategies that not necessarily minimise the delay (by distributing packets for the same output port equally over the available FDLs). The impact of the scheduling strategy is clearly an issue that will need to be resolved. The simulation results in the next sections all use strategy (a). However, we will present results for different strategies at the conference.

D.3.4 Traffic models

As the DAVID node operates in a slotted manner, we use a discrete event simulator. For each time slot we need to know how many packets are offered for forwarding at the input ports. The packets will be generated by traffic sources. Each source will produce packets for a particular (input port, output port)-pair, with a given priority, at a given rate (i.e. average number of packets per time unit). We have implemented different traffic models, including:

- **POISSON:** Poisson inter-arrival times between consecutive packets;
- **BURST_GEO_POISS:** bursty traffic, generating bursts with geometrically distributed number of packets per burst and Poisson distributed inter-arrival times between bursts (packets within a burst arrive at same time).
- **PARETO:** On/Off traffic using Pareto distribution for both on- and off-periods. This model was added because it is well-known that an aggregate of these sources will produce a packet stream that exhibits self-similar behaviour [6,7]. The Pareto distribution's probability mass function is given by eq. 1, thus we have four free parameters for both the on- and off-periods: the shape parameters λ_{on} resp. λ_{off} and the minimum burstsizes b_{on} resp. b_{off}.

$$p(x) = \frac{\alpha \cdot b^{\alpha}}{x^{\alpha+1}}, \text{ for } x \geq b, \text{ with } \alpha, k > 0 \qquad (1)$$

D.4 Single node prformance: simulation results

In this section we briefly discuss the first simulation results obtained with the node simulator as discussed in the previous section.

D.4.1 Simulation set-up

The simulations presented in this section are based on the parameter values that are envisaged in the demo testbed under development within the DAVID framework. The switching fabric to be used in DAVID will have a dimension of 256×256, which means that it will be a matrix interconnecting 256 input wavelengths and 256 output wavelengths. There will be 6 input/output ports in the OPR, each carrying 32 wavelengths. That leaves 64 wavelengths providing connection to the fibre delay lines in the recirculating buffer.

The traffic pattern offered at the inputs was considered to be uniform, meaning that for each input, the load was equally distributed over the 6 output ports. Also, 3 priorities were considered, where 50% of the traffic had the highest priority (2), 25% was of middle priority (1), and the other 25% was of lowest priority (0). To create this set-up, we used a single traffic source per (input port, output port, priority)-triple, thus a total of 6·6·3=108 traffic sources. The total offered load at the inputs by these sources was set at 0.8 (meaning that per input port, on average 0.8·32=25.6 packets were offered per timeslot). These parameter values are summarised in Table D.2.

Table D.2: Simulation parameters

Parameter	*Value*
nr. of input/output ports	6
nr. of λs per input/output port	32
priorities	50% high (2), 25% middle (1), 25% low (0)
traffic load	0.8

The traffic source models used are the Poisson and Bursty models mentioned in D.3.4. The Poisson model uses Poisson distributed inter-arrival times between successive packets generated by the same source. The Bursty model uses Poisson inter-arrival times between bursts of geometrically distributed size with an average of n packets per burst. We have set n to 2 and to 4. This means that for those cases, the traffic arrives in bursts of average size 2 (or 4) packets per timeslot.

With the simulations presented below, we have explored two buffer configurations. The "Fixed Loop Size" case of Section D.4.2 investigates the effect of adding extra wavelengths on FDLs of the same size (Figure D.4), whereas a second set of simulations (Section D.4.3) explored the advantage of using the extra wavelengths for FDLs of increasing length (Figure D.5).

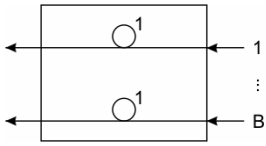


Figure D.4: Fixed loop size.

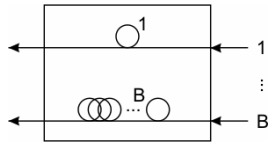


Figure D.5: Increasing loop size.

D.4.2 Fixed loop size

The simulations presented in this section considered the buffer configuration sketched in Figure D.4. We consider a buffer block consisting of fibre delay lines of length 1 (timeslot). The graphs show the loss rates for an offered load of 0.8 for an increasing number of wavelengths (parameter B on the figure). The plots marked as BURST-4 are for Bursty traffic (see D.3.4) with average burst size 4, BURST-2 for burst size 2. The curves for POISSON are for Poisson traffic. The curves for PARETO-1 (resp. PARETO-2) are for Pareto On/Off sources using α_{on}=1.3, b_{on}=1 (resp. b_{on}=2) and α_{off}=1.5. The remaining parameter boff follows from the constraint that the load has to be a given value. The values were inspired by [7], which reports on-times being heavy tailed with $\alpha_{on} \sim 1.0$ to 1.3, and $\alpha_{off} \sim 1.5$.

From these graphs, several observations can be made. In the first place, we notice that the loss probability is the highest for low priority packets. Also, the effect of increasing the number of FDLs on the PLR is the biggest for the high priority packets: the slope of the declining curves (for increasing number of wavelengths) is steeper for higher priority packets. This confirms that the priority scheme is

working: only if the highest priority packets are satisfied other packets can use the FDLs.

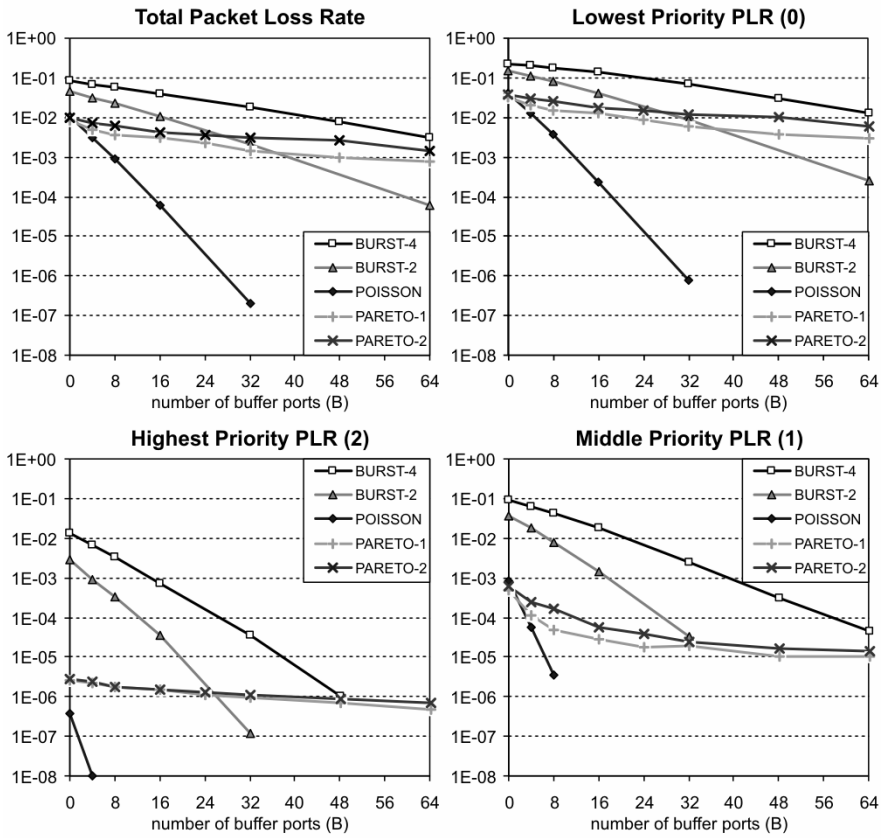


Figure D.6: PLR for increasing number of loops of length 1 (i.e. delay of single slottime).

With regard to the traffic source models, the differences are striking. Increasing the buffer size for Bursty traffic is far less effective than for Poisson traffic: for Bursty traffic, the slopes of the declining curves are fare less steep, and the PLRs are significantly higher, especially for the high priority packets, where the difference is about a factor 10^5: Whereas for Poisson traffic the high priority traffic could possibly do without buffering (loss rate less than 10^{-6}), this is most certainly not the case for Bursty traffic. From this, we may conclude that —if possible— we should avoid creating bursts of packets all destined for the same output port. The results seem to indicate that shaping the traffic at the inputs, in order to spread packets for same destination over different timeslots can help a lot. This means also that we should avoid any scheduling strategy that would create such bursts. Surely, use of randomness when choosing between packets of the same priority from different flows, is important here. This randomness will smoothen burstiness.

D.4.3 Increasing loop size

The results presented below again show packet loss rates for an increasing number of wavelengths used. However, in this case we use a FDL of a different length for each additional wavelength: if we use B wavelengths, we use FDLs of lengths 1, 2, 3, …, B timeslots as illustrated in Figure D.5. The traffic sources used are again of Poisson, Bursty, and Pareto types, and all results are for load 0.8.

When we compare the PLRs for different priorities, we notice that again the highest priority packets lose less. However, if we look at the rate of PLR decrease with increasing number of buffer ports (wavelengths), we see a striking difference with the graphs for the fixed length delay line case (Figure D.7).

Adding additional buffer ports with increasing FDL lengths (instead of more of the same) has a major impact on the PLR. The highest priority traffic gains about as much as in the case of fixed FDLs, but for FDLs with increasing lengths, the lower priorities gain also. From the individual graphs per priority (that are omitted due to space limitations), we see that the decreasing slope is now more of the same order for the different priorities.

These results seem to suggest that it is advantageous to add buffer ports of increasing FDL length instead of more of the same.

D.4.4 A note on fairness

To verify the operation of our simulator, we have also investigated the fairness among different flows of the same priority. In the graphs below, the result of that analysis is illustrated for a single simulation: the case where there is a buffer of 8 FDLs of length 1, and the load is 0.8 generated by traffic sources of the Burst type, with mean burst size of 4. Similar graphs are obtained for other cases.

According to our scheduling strategy, we expect that losses for packets going to the same destination are equally distributed over the flows coming from the different input ports. In Figure D.8, we see the loss ratios for al the individual sources generating traffic with output port 1 as destination. As expected, we see that the losses within a priority class are neatly distributed over all the flows coming from the different input ports.

In Figure D.9, we see the averages for the different output ports. (This means that the three bars at “dest port” 1 are actually the averages of the corresponding bars in Figure D.8. The graph shows that the loss rates are the same for every output port. This indicates that the losses are fairly distributed over all output ports, which is what we wanted (and thus expected).

For completeness, we have also compared sources coming from a single source port (going to the different output ports). The resulting graphs (which are not shown due to space limitations) again indicate that the losses are fairly distributed among the different flows.

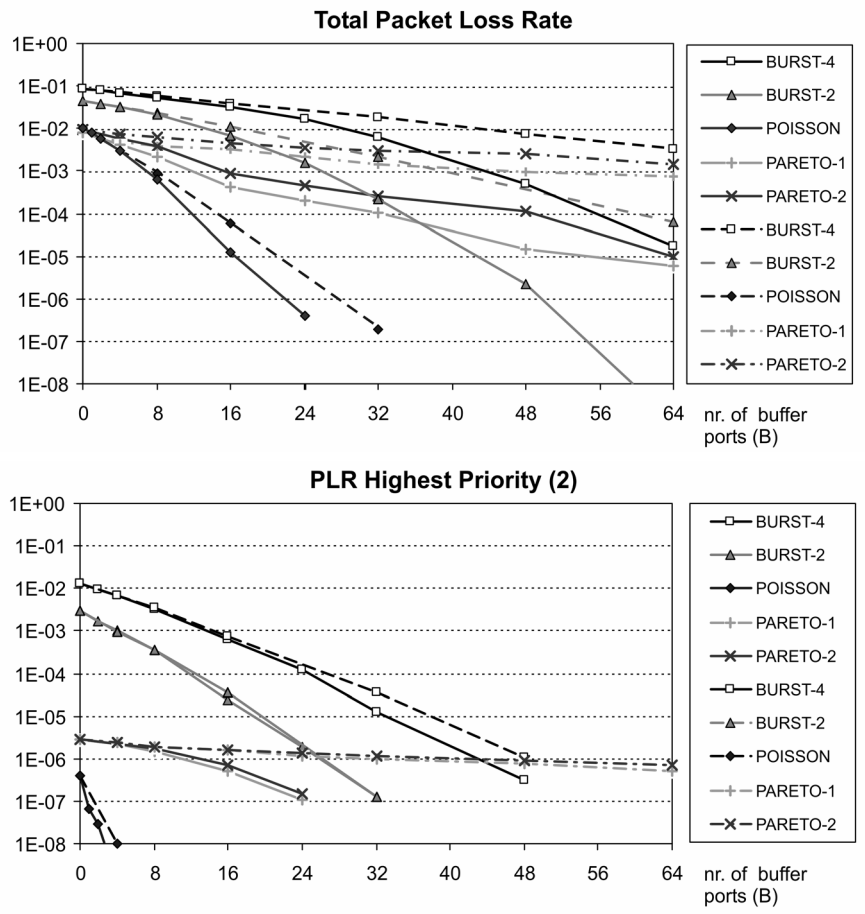


Figure D.7: Comparison of PLR for FDLs of fixed size (dashed lines) versus increasing size (full lines) for an increasing number of buffer wavelengths.

D.5 Conclusion

We have presented part of the work carried out within the DAVID project on assessing the performance of the proposed OPR. The presented results show the packet loss rate for streams of fixed size packets (length is equal to DAVID slot length) with service differentiation: we implement a strict priority scheme with three priorities. It is shown that through introducing randomness in the decision procedure, the losses are equally spread among flows of the same priority.

Through our simulation results, we demonstrate that it is advantageous to add extra FDLs to the optical buffer structure, instead of providing more wavelengths on the same FDL (i.e. same buffer depth). However, the traffic source model has a

strong impact on how effective the reduction in loss rate, compared to no buffering, is.

We have also proposed different buffer scheduling strategies to decide upon what packets to put in which delay line. The results covered in this paper only reflect one strategy, but results for other strategies will be presented at the ONDM conference.

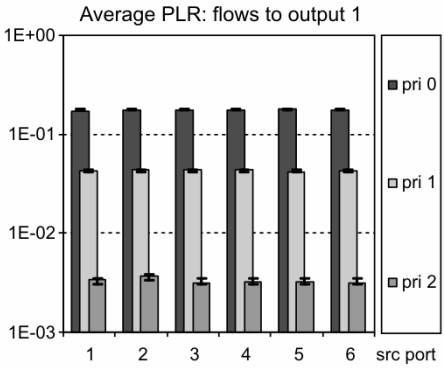


Figure D.8: PLR for individual sources generating traffic for output port 1; the deviation markers indicate the interval (avg-stdev, avg+stdev) taken over the six traffic sources per priority class.

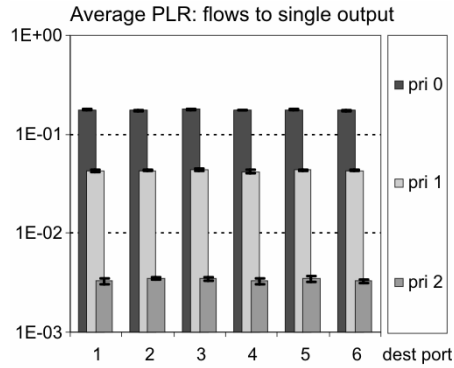


Figure D.9: Average PLR over all sources generating traffic for a particular output port; the deviation markers indicate the stdev on those averages.

Acknowledgements

This work has been supported by the European Commission through the IST-project DAVID, and by the Flemish Government through the IWT-project ITA/980272/ INTEC. The first author is a Research Assistant of the Fund for Scientific Research – Flanders (F.W.O.-Vl., Belgium).

References

[1] L. Dittman, H. Christiansen, D. Chiaroni, "DAVID – An approach towards MPLS-based optical packet switching with QoS support", in Proceedings of Photonics In Switching 2001 (PIS 2001), Monterey, CA, June 2001.
available online: *http://david.com.dtu.dk*

[2] A. Bianco, G. Galante, El. Leonardi, F. Neri, "Access control protocols for interconnected WDM rings in the DAVID metro network", in Proceedings of Thyrrenian International Workshop on Digital Communications, Taormina, Italy, September 2001.

[3] Dittman, H. Christiansen, M. Berger, "Hierarchical MPLS – An approach for

efficient resource administration in multi-technology networks", in Proceedings of the 6th European Conference on Networks and Optical Communications (NOC 2001), Ipswich, England, June 2001.

[4] D. Chiaroni, "Status and applications of optical packet switching", Tutorial at 27th European Conference on Optical Communication (ECOC 2001), Amsterdam, The Netherlands, October 2001.

[5] D. Chiaroni, N. Le Sauze, T. Zami, J.-Y. Emery, "Semi-conductor optical amplifiers: a key technology to control the packet power variation", in Proceedings of 27th European Conference on Optical Communication (ECOC 2001), Amsterdam, The Netherlands, October 2001.

[6] W. Willinger, M.S. Taqqu, R. Sherman, D.V. Wilson, "Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level", IEEE/ACM Transactions on Networking, Vol. 5, No. 1, January 1997, pp. 71-86.

[7] M.E. Crovella, A. Bestavros, "Self-similarity in World Wide Web traffic: evidence and possible causes", IEEE/ACM Transactions on Networking, Vol. 5, No. 6, December 1997, pp. 835-846.

Appendix E

Strategies for an FDL based feed-back buffer for an optical packet switch with QoS differentiation*

C. Develder, M. Pickavet, P. Demeester

Proc. Int. Conf. on Optical Internet (COIN2002), paper COIN.TuD1, Cheju Island, Korea, 21–25 Jul. 2002, pp. 114–116.

Abstract: To match switching technology to the huge capacities provided by (D)WDM, migration to Optical Packet Switching (OPS) is foreseen. A crucial issue in packet switching is avoiding losses when multiple packets arriving at the same time contend for the same resource, in casu the wavelength(s) on an output fibre. In optics, no RAM is available: the only memory available is based on the use of Fibre Delay Lines (FDLs) which store packets for a pre-defined period of time by sending light through a fibre of well defined length. In the frame of the European research project DAVID, we study a so-called Optical Packet Router (OPR) with feed-back buffer that switches fixed-length optical packets, and compare different FDL based structures and scheduling strategies. As the delay in the access part of the network usually is by far larger than in the backbone where OPRs are to be deployed, the main criterion is the packet loss rate (PLR), which we assess through simulation. Various strate-gies are compared for memory-less Poisson and bursty traffic types, as well as self-similar sources.

* This work has been supported by the European Commission through the IST-project DAVID (IST–1999–11387), and by the Flemish Government through the IWT GBOU-project "Optical Networking and Node Architectures". C. Develder is supported as a Research Assistant of the Fund for Scientific Research – Flanders (F.W.O.–Vl.), Belgium

Index terms: (D)WDM, Optical Packet Switching, logical performance, simulation.

E.1 Introduction

The deployment of (D)WDM networks successfully answers the ever lasting hunger for bandwidth. Recent research projects and related work within standardisation bodies (e.g. ITU and IETF with ASON-related work and GMPLS) focus on moving onwards from the deployment of point-to-point connections to real optical networking. The approach taken is essentially a circuit-switched one: (virtual) wavelength circuits are set-up between ingress-egress pairs of the network. Although the operation and design of those networks is relatively easily achievable, they are hard to efficiently exploit in scenarios with highly variable traffic patterns. Optical Packet Switching (OPS) addresses this issue by exploiting TDM. While profiting from advances in optical technology, OPS offers better bandwidth granularity, thus efficiency, and flexibility.

To guarantee successful operation of an OPS network, the Packet Loss Rate (PLR) needs to be sufficiently low. In electronics this is achieved by temporarily storing the packets in RAM. In the optical domain, Fibre Delay Lines (FDLs) are used to temporarily delay excess packets. In the following, we discuss FDL buffer structures and schedul-ing strategies for an Optical Packet Router (OPR) with a feed-back FDL buffer, proposed within the framework of the European research project DAVID. To provide service differentiation, the OPR adopts a priority mechanism.

The remainder of the paper is structured as follows: Section E.2 presents the architecture under study, continuing with the OPR operation in Section E.3. Two major feed-back buffer structures are compared in Section E.4: having only one FDL length, or rather multiple FDL lengths. For the latter, Section E.5 discusses various scheduling strategies. The impact of service differentiation on the overall PLR is elaborated on in Section E.6. The final Section E.7 summarizes the conclusions.

E.2 Network and node architecture

The European research project DAVID (Data And Voice Integration over DWDM) aims at proposing a viable approach towards OPS. A network architecture is proposed encompassing both metro and backbone DWDM networks. In the backbone, Optical Packet Routers (OPRs) are interconnected in a mesh used to transport fixed-length packets, which are synchronized at the input ports of each OPR, operating in slotted mode.

A broadcast-and-select switching fabric using SOA technology, described in [1], forms the core of the OPR with a logical structure as sketched in Figure E.1. The input and output ports of this matrix are divided between the W wavelengths of the F fibres connecting the OPR to its neighbours. To solve possible contention, the wavelength domain is exploited: wavelength converters are foreseen at the switch's

ports. In addition, B wavelength ports are reserved for connection to and from the buffer block made up of one or more FDLs (free of switching elements).

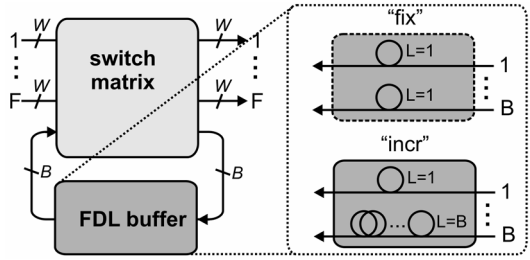


Figure E.1: Logical structure of the OPR with two sample FDL configurations.

E.3 Packet scheduling

The OPR operates in a slotted way: every timeslot, it inspects packets arriving at its input ports, and subsequently decides what packets to forward (to the output ports or the feed-back buffer) or to drop. This decision is taken by following a fixed procedure, comprising two phases: (i) for each output fibre of the OPR, elect at most W packets to be forward directly, (ii) from the remaining packets, elect at most B to put in the buffer; any other packet will be lost.

Election of packets for forwarding and buffering is based on two criteria: the priority attached to the service class the packet belongs to, and the time it already spent in the OPR. The service differentiation is based on a pure priority scheme: packets of a higher priority class are given precedence over lower priority ones. Within the set of packets with the same priority, the one which has spent most time in the OPR already is favoured. Among multiple packets sharing the same priority and time spent in the OPR, one is selected randomly.

E.4 Choosing a buffer configuration

For feed-back buffer's FDL structure, there are essentially two options: use a single FDL length for all B buffer ports, or adopt different FDL lengths. The latter offers greater buffer capacity for the same number of switching fabric ports devoted to the recirculating buffer.

For the two buffer structures outlined in Figure E.1, we have assessed the logical performance for an OPR with F=6 input and output ports, each carrying W=32 wavelengths. Figure 2 plots the PLR comparing the fixed FDL case (fix, L=1 for all B ports), and the case with increasing FDL lengths (incr, L=1,2,3…B) for increasing number of buffer ports B=0…64. Results are shown for three traffic source types. The first is the well-known Poisson process. The GeoOnOff source generates bursty trains of packets: an on/off source with geometrically distributed lengths of both on- and off-periods. Self-similar traffic labelled ParetoOnOff was generated using on/off sources with Pareto distributed on- and off-times [2]. A

uniform traffic matrix was used, for a total load of 0.95. Three traffic priority classes were used for all presented simulation results: 50% highest, 25% middle and 25% lowest priority traffic.

In accordance with intuition, we find that the buffer with increasing FDL lengths for the B buffer ports largely outperforms the buffer with a single FDL length, and the more with increasing B (cf. growing difference in "storage" capacity). For the memory-less Poisson traffic sources, the advantage of the incr approach over fix mounts up to a factor close to two orders of magnitude already for B=32. Also for bursty GeoOnOff traffic, the increasing FDL length buffer gives PLRs that may differ an order of magnitude or more. Yet, for the self-similar traffic model ParetoOnOff, the differences are far less striking: the effect of adding buffer space is not that effective.

Obviously, the better logical performance of using different FDL lengths needs to be counterposed by the risk of re-ordering of packets belonging to the same flow, which can be avoided completely by using a single FDL length of 1 slot-time. In addition, the single FDL length implies that a single physical FDL can be used for all B buffer ports, through (D)WDM. Also, the multiple FDL length approach suffers from its need for a more complex buffer scheduling algorithm, as discussed in the next section.

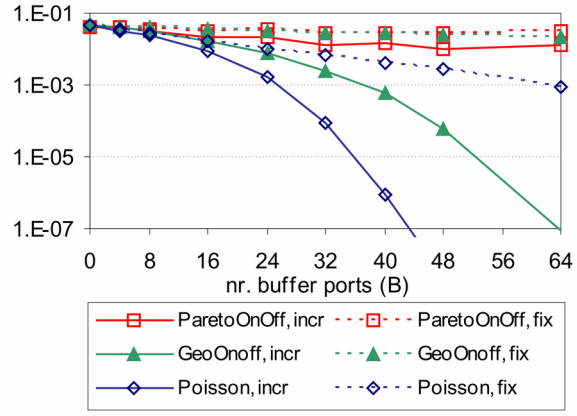


Figure E.2: Comparison of using a single FDL length (fix, dashed lines) and increasing FDL lengths (incr, full lines).

E.5 Buffer strategies

For a buffer with differing FDL lengths, not all the B buffer ports are equivalent. Thus, in addition to the election procedure of packets to buffer, a decision procedure is needed to determine the FDL length to use. The following four strategies were compared:

- ***MinDelay:*** for each packet entered in the buffer, the free buffer port with smallest corresponding FDL length is chosen; this is the strategy used in Figure E.2.

- ***NoOvr:*** to buffer packet p, take the FDL with smallest length L such that no more than W packets of the same or higher priority than p will leave the buffer at now+L for the same output fibre of the OPR; otherwise drop the packet.
- ***AvoidOvr:*** first seek the free port with smallest FDL length that would not cause overload; enter the packet at the free port with the smallest FDL length if no such overload-avoiding port can be found.
- ***Balance:*** contending packets are spread in time. To buffer a packet p, count (N_L) for each available FDL length L, the packets scheduled at now+L for the same output port destination as p, and of the same or higher priority as p. The packet is then put in the free FDL with the smallest count N_L.

Figure E.3 compares the PLR achieved. The Balance strategy largely outperforms the others for both Poisson and the bursty GeoOnOff models (factors up to 6, resp. 3, for B=40). For the self-similar ParetoOnOff traffic, no significant reduction of PLR can be achieved through choosing an appropriate strategy.

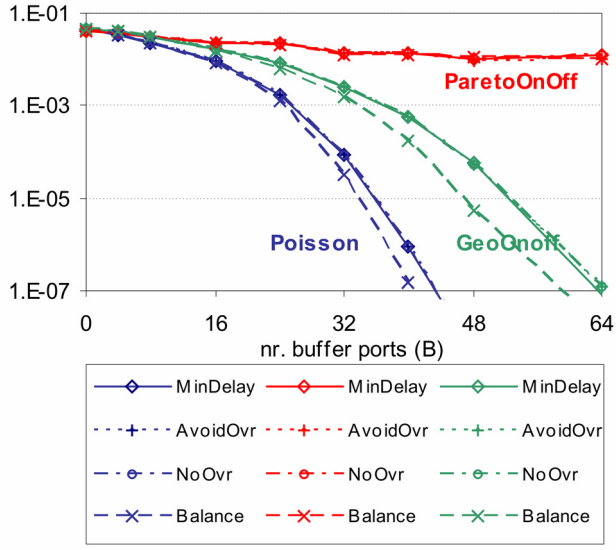


Figure E.3: Comparison of four buffer strategies for a feed-back buffer with increasing FDL lengths (configuration incr).

E.6 The cost of service differentiation

As outlined before, the proposed OPR architecture provides service differentiation by giving absolute precedence to packets of higher priority (recall the election scheme for forwarding and buffering, Section C). To assess the impact of using a priority-based scheme, we have compared its logical performance with a scheduling algorithm that discards traffic class information (i.e. attaches the same priority to all traffic classes).

The results of that comparison are presented in Figure E.4 for the Balance buffer strategy. For GeoOnOff traffic, the priority scheme surprisingly outperforms the one where they are ignored, especially for a large number of buffer ports. The reason is that for this particular traffic type, with limited On-periods, it is better to favour packets destined for an output port suffering heavy contention at the time we are making the buffering decision: the chance that the overload caused on that port will have subsided when a packet comes out of the FDL is bigger for the delays offered by the longer FDLs. Such spreading in time of packets is effectively achieved for the Balance strategy under study. Favouring packets destined for ports suffering from heavy contention is more pronounced when using priorities than when not, which strengthens the advantageous effect of service differentiation for larger buffer sizes in the GeoOnOff case.

The reduction in PLR does not show up when there is no significant correlation of the amount of overload on a particular output port between timeslots spaced at scales in the range of the FDL delays, e.g. for the memory-less Poisson traffic. In case of ParetoOnOff traffic, because of its long-range correlations, the PLR is even slightly increased when deploying service differentiation. For other buffer scheduling strategies (not shown in Figure E.4), which do not successfully spread packets destined for the same output port in time, the peculiar reduction in PLR for GeoOnOff traffic is not that pronounced.

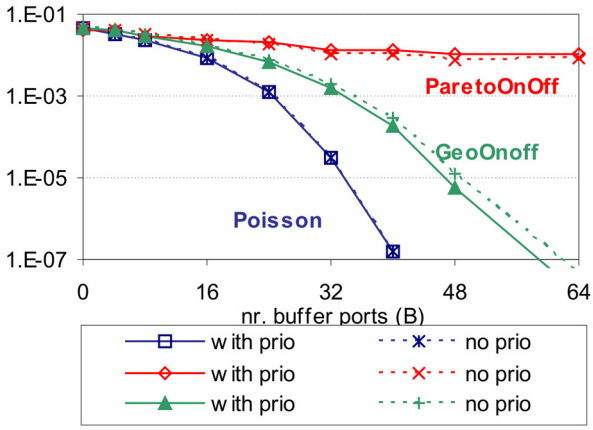


Figure E.4: Overall PLR when using service differentiation (with prio) vs. when ignoring traffic priority class information (no prio) for the buffer with increasing FDL lengths (incr) using the Balance strategy.

E.7 Conclusions

We have investigated two distinct FDL configurations for a feed-back buffer in OPS. By using FDLs of different lengths, the PLR can be cut down significantly (up to multiple orders of magnitude) compared to a fixed length approach. The PLR can be further reduced if an appropriate buffer strategy is chosen. The penalty of using

service differentiation was shown to be limited, or even non-existing, in terms of PLR.

However, the effectiveness of the FDL buffer and associated buffering strategies largely depends on the traffic type: for self-similar traffic, the PLR can not be effectively reduced through the use of FDL buffering.

References

[1] D. Chiaroni, et al., "First demonstration of an asynchronous optical packet switching matrix prototype for MultiTerabitclass routers/switches", Proc. of ECOC2001, Amsterdam, The Netherlands, October 2001.

[2] W. Willinger, et al., "Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level", IEEE/ACM Transactions on Networking, vol. 5, no. 1, Jan. 1997, pp. 71–86.

Appendix F

On trains and wagons: switching variable length packets in a slotted OPS network

C. Develder, M. Pickavet, P. Demeester

Proc. Conference on Optical Internet / Australian Conference on Optical Fibre Technology (COIN/ACOFT 2003), paper TuA2-6, Melbourne, Australia, 13-16 Jul. 2003, pp. 233–236.

Abstract: Optical packet switching allows to fully and efficiently exploiting the capacities offered by (D)WDM. In this paper we investigate how slotted optical switches can deal with variable length packets, which is typical of IP.

F.1 Introduction

To satisfy the ever lasting hunger for bandwidth, nowadays' communication networks resort to the deployment of (D)WDM networking. An initial step taken today is migration from still predominant point-to-point systems to real optical networking supporting circuit-switched optical paths [1]. Yet, despite their relative ease of design and operation, they suffer from the difficulty of dealing with highly variable traffic. Optical Packet Switching (OPS, [2]) is a longer term strategy exploiting fast optical switching techniques to offer better bandwidth granularity, efficiency and flexibility. The main difference with the Optical Burst Switching (OBS) concept [3], is that OPS operates in a slotted mode: packet arrivals at the inputs are aligned to slot boundaries and packets arriving in the same slot can be switched jointly.

Despite the essentially slotted concept, OPS switches can be used to deal with variable length packets by chopping them into chunks fitting within one slot. Thus, we obtain a train of slots constituting a single variable length packet. As indicated in Figure F.1, there are essentially two ways to treat these trains: either treat them as a

whole and take decisions for the whole train at once, or rather treat each wagon (i.e. slot) independently.

A comparison of the train versus the wagon approach has been presented in [4] for shared optical busses using an access protocol for high-speed LANs/MANs. The authors discussed the overhead reduction attained by using a train-approach, and studied the delay vs. troughput behavior to conclude that for short train lengths the wagon approach proved to be more efficient.

In this paper we try to find out which approach is the best in terms of logical performance (ie. data loss due to contention, delay, service differentiation capabilities) for an optical packet switched WAN comprising slotted switches. In the next section F.2, we outline the switch architecture and the scheduling algorithm used. The subsequent Section F.3 presents the simulation set-up taken to answer the train-or-wagons question. The results are summarized in Section F.4, before concluding in Section F.5.

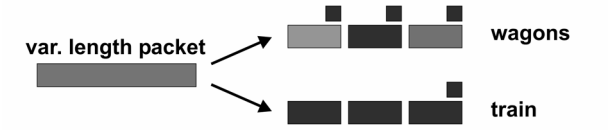


Figure F.1: Dealing with variable packets in a slotted switching concept: train versus wagons.

F.2 An optical packet switch

The switch architecture we focus on was proposed within the European research project DAVID [6]. It consists of an all-optical switch matrix based on SOA technology. In- and output ports of this matrix are connected to F fibers (each carrying W wavelengths) providing connections to other switches. Wavelength convertors are provided to help solving contention. In addition, B ports are connected to a recirculating FDL buffer, which is fully shared among all I/O ports. The node structure is outlined in Figure F.2.

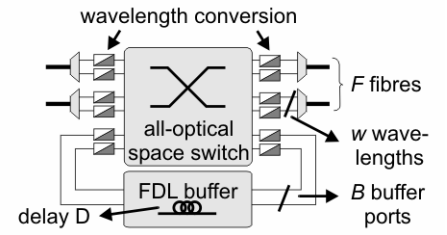


Figure F.2: The OPS switch architecture under study.

The switch operates in a slotted way: every timeslot, it inspects packets arriving at its input ports, and subsequently decides what packets to forward or to drop. This decision is taken by following a fixed procedure, comprising two phases: (i) for each

output fiber of the OPR, elect at most W packets to be forward directly, (ii) from the remaining packets, elect at most B to put in the buffer; any other packet will be lost. We do not consider deflection routing, since it is only effective at low network loads and can lead to out-of-order packet deliveries.

In this paper we investigate the behavior of such a single switch when the traffic it needs to forward consists of variable length packets. To handle these packets with the slotted switch, they need to be segmented into slots. One way to forward the resulting trains is to deal with each "wagon" individually. This implies that each slot needs to have an individual header, as indicated in Figure F.1.

The alternative, sometimes referred to as Slotted Variable Length Packets (SVLP) [7], is to treat the complete train as a whole and take a decision for the train upon arrival of its first wagon. Thus, a single full header (containing e.g. source, destination address, traffic class) suffices for the forwarding process. The fact that the following wagons belong to the same train can be indicated through e.g. a continuation bit field [4].

Note that we assume that the headers are transmitted on an orthogonal channel, e.g. through ASK/DPSK modulation [5]. Thus, the train length measured in slots will be the same for the train and wagon approach.

F.3 Simulation scenario

To compare the train versus wagon approaches, we focus on a single switch. To obtain the various performance parameters, we resorted to simulation. To guarantee trustworthy results, we used a high-quality random generator and assured 95% confidence on the results (for the sake of clarity, error margins are however not shown on the graphs). The parameters used are listed in Table F.1.

Table F.1: Simulation parameters

Parameter	*Value*	*Meaning*
F	6	nr. of I/O fibers
W	8	nr. of wavelengths per fiber
B	[0, 64]	nr. of recirculating buffer ports
D	2·L	FDL delay
L	[1.5, 20]	average train length (unit = slots)

F.3.1 Traffic model

The traffic model used for the simulations is clearly a packet train model, similar to the one proposed in [8]. For both the train lengths and gaps between successive trains we used a negative exponential distribution, while the inter-wagon gaps were always zero (cf. a train consists of wagons in successive slots). While this distribution may not be the most realistic one when trains are interpreted as being IP

packets, the qualitative conclusions of our results are fairly independent of the train length distribution.

F.3.2 Performance criteria

The main performance criterion in an OPS environment is the loss rate: packets (trains) can get lost if both wavelength conversion and buffering fail to solve inevitable contention. Since a train is considered to be a single data unit, we assume it to be lost as soon as a single wagon is dropped at the switch. Important in the variable packet length concept is also the fairness of the scheduler: does it discriminate long trains against shorter ones?

Another performance criterion is delay. However, in the context of OPS WANs, it is only of secondary importance, since it will be limited compared to propagation delays and delays in access and metro parts of the network. Therefore (and because of space limitation), we will not discuss it in this paper.

As the OPS network will need to transport various traffic classes, it should be apt to support service differentiation. In [9], we investigated multiple service differentiation approaches for asynchronous variable length packets. In a slotted environment, the simplest approach is to use a simple priority mechanism: indicate the priority in the packet's header and give strict preference to higher priority packets when making the forwarding and buffering techniques. This was proven to be very effective in a fixed-length packet environment [10]. In this paper we investigate if it also is suitable for a train or wagon approach for variable length packets.

A last criterion is processing overhead. It is clear that since a wagon model requests every wagon to have its own header, the amount of forwarding decisions to be made at the switch will be a factor higher than in case of a train approach, roughly equal to the number of wagons per train.

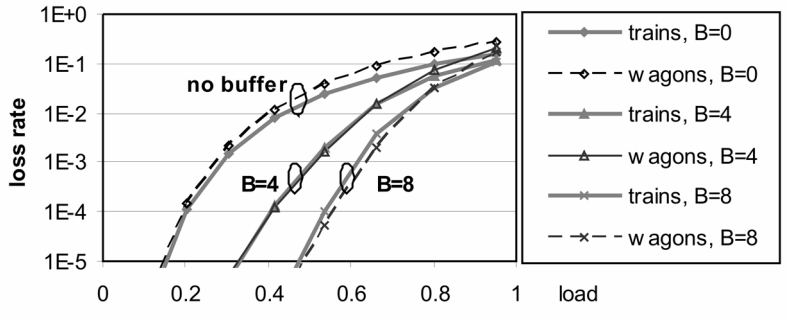
F.4 Trains or wagons

In this section we try to answer the question: should we adopt a train or a wagon approach? We first look at the loss rates for increasing loads in subsection F.4.1. The influence of the slot granularity (ie. ratio of slot length vs average train length) is investigated in part F.4.2. The last subsection F.4.3 focuses on service differentiation capabilities of both train and wagon approaches.

F.4.1 Influence of load

Since the loss rate for a given load will clearly depend on the amount of buffer, we provide results for three sample buffer sizes: no buffer (B=0), four (B=4) and eight (B=8) recirculating buffer ports. The loss rates for this set-up are plotted in Figure F.3(a). Clearly, the loss increases with higher loads, and buffering aids in limiting the loss.

(a)



(b)

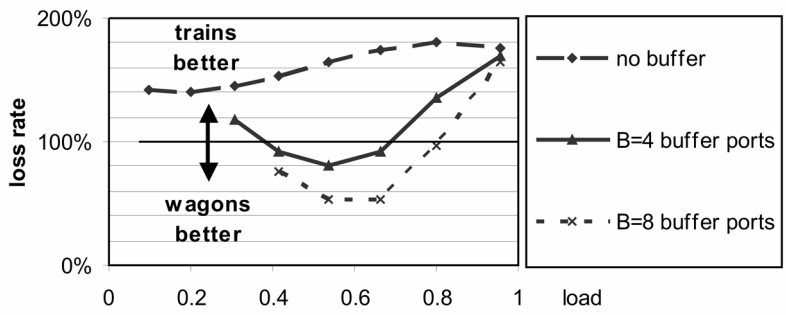


Figure F.3: Wagons vs trains: loss rates for increasing load, with B=0,4,8 buffer ports: (a) loss rate, (b) ratio of loss rate: wagons / trains.

Comparing the wagon versus train approach, we show the ratio of the loss rate attained by the wagon approach divided by that of the train approach in Figure F.3(b). When there is no buffer (B=0), we find that the wagon approach performs worse (ratio above 100%). However, when a buffer is present, the wagon approach achieves lower losses. Still, this is only the case for a particular load range: for very high or very low loads, the train approach performs better.

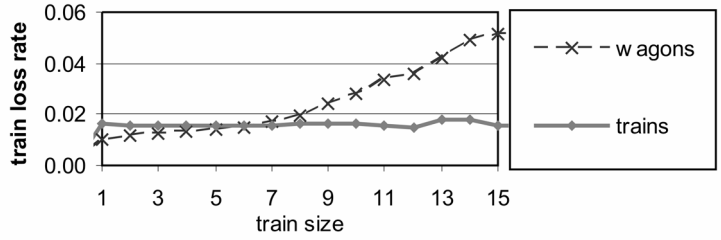


Figure F.4: Illustration of unfairness in loss rates for the wagon approach: loss rate per train length, for B=4 and a load of 0.62.

As indicated previously, loss rates may depend on the train length. In Figure F.4, we plot the loss rates per train size for B=4 buffer ports and a load of 0.62. Since the buffer size is chosen such that it can accommodate about 95% of the train lengths, the train scheduling approach is quite fair. For the wagon approach however, since each slot is treated independently and a train is lost as soon as a single wagon is dropped, the unfairness is quite severe (max vs min loss rate differ with about an order of magnitude).

F.4.2 Influence of granularity

The efficiency of handling variable length packets with a slotted switch will greatly depend on the slot resolution. For a given train size distribution, the choice of a given slot size will obviously determine the amount of wasted bandwidth because of padding. But even when this is ignored, the performance in terms of loss will also be influenced.

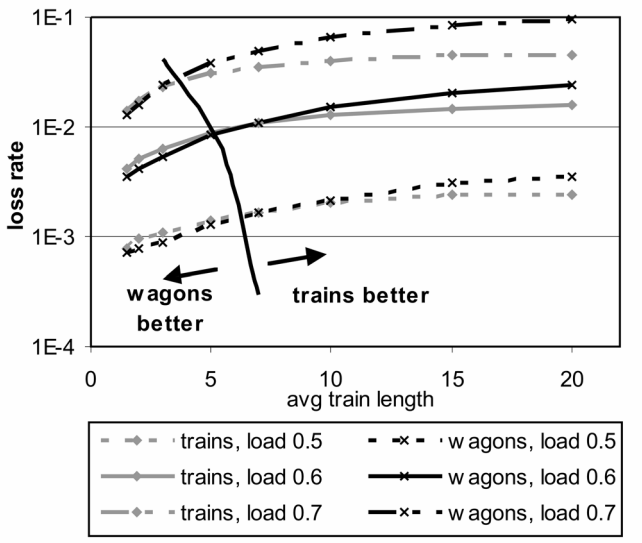


Figure F.5: Wagons vs trains: loss rates for increasing train lengths, with B=4 buffer ports, loads=0.5, 0.6, 0.7

In Figure F.5 we plot the loss rates for increasing average train length. As intuitively expected, the wagon approach only performs better for small train lengths. The crossover point moves slightly to larger train lengths when the buffer is increased.

F.4.3 Service differentiation

A simple priority mechanism, based on priority indicated in a packet's header showed to provide adequate class separation in a fixed packet length environment [10]. In this section we consider the same approach for trains an wagon approaches.

As an example, Figure F.6 shows the loss rates when using two priority classes, with 40% of the traffic having the high priority. Since in the train approach, high priority packets cannot preempt lower priority trains that arrived a few slots earlier, the differentiation achieved is far less pronounced compared to the wagon approach. The low priority loss rates dominate the overall loss rate, which evolves as in the priority-less case as plotted earlier in Figure F.3(a).

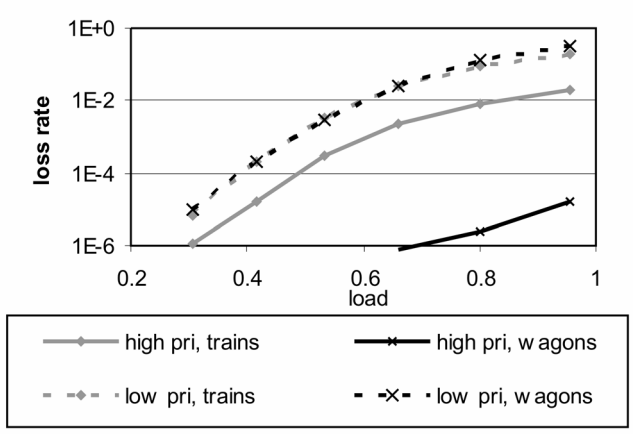


Figure F.6: Service differentiation of wagons vs trains: loss rates for increasing loads, with B=4 buffer ports.

F.5 Conclusions

To our knowledge, this is the first paper to quantitatively compare the train vs wagons approach for optical packet switches in a WAN context, ie. a mesh of optical packet switches rather than a MAN/LAN environment with an access protocol. In a slotted OPS switch variable length packets, which are splitted into slots, can be treated either as a whole (trains) or on a slot-by-slot basis (wagons).

The wagon approach can help to reach lower overall (train) loss rates when there is a buffer, and trains are relatively short (ie. a few slots). The more buffer, and the shorter the trains, the greater the potential advantage is. However, this only holds for a limited range of loads: when the load is either low or rather high (order 0.8 and above), the train approach is to be preferred.

From a service differentiation point of view, the wagon approach is able to reach more pronounced service differentiation when a simple priority-based approach is adopted.

The potential advantages of a wagon approach are paid for by an increased control overhead and load on the scheduler (factor of order of average train length measured in slots) and unfairness, in the sense that it more severely discriminates longer trains.

References

[1] B. Mukherjee, “WDM optical communication networks: Progress and challenges”, JSAC, vol. 18, no. 10, Oct. 2000, pp. 1810–1824.
[2] D.K. Hunter, I. Andonovic, “Approaches to optical Internet packet switching”, IEEE Comm. Mag., vol. 38, no. 9, Sep. 2000, pp. 116–122.
[3] C. Qiao, M. Yoo, "Optical Burst Switching - A new paradigm for an optical Internet", J. High Speed Networks, vol. 8, no. 1, Jan. 1999, pp. 69–84.
[4] B. Mukherjee, A.E. Kamal, “Scheduling variable-length messages on slotted, high-speed fiber optic LANs/MANs using the continuation-bit approach”, Proc. INFOCOM 1991, vol. 2, pp. 678–687.
[5] N. Chi, et al., “Optical Label Swapping and Packet Transmission Based on ASK/DPSK Orthogonal Modulation Format in IP-over-WDM Networks”, Proc. OFC2003, paper FS2, vol. 2, pp. 792–793.
[6] L. Dittmann (ed.), et al., “The European IST Project DAVID: a Viable Approach towards Optical Packet Switching”, to appear in JSAC.
[7] F. Callegati, et al., “Assessment of packet loss for an optical feedback buffer node using slotted variable-length packets and heavy-tailed traffic”, Proc. ICTON 2002, vol. 1, pp. 51-56.
[8] R. Jain, S. Routhier, “Packet Trains — Measurements and a New Model for Computer Network Traffic”, JSAC, vol.4, no. 6, sep. 1986, pp. 986–995.
[9] C. Develder, J. Cheyns, M. Pickavet, and P. Demeester, “Service differentiation for OBS in a switch with recirculating FDL buffer”, in press.
[10] C. Develder, M. Pickavet, and P. Demeester, “Strategies for an FDL based feed-back buffer for an optical packet switch with QoS differentiation”, Proc. COIN 2002, paper COIN.TuD1, pp. 114–116.

Appendix G

Service differentiation mechanisms for variable length packets in an optical switch with recirculating FDL buffer

C. Develder, M. Pickavet, P. Demeester

Tech. Digest Photonics in Switching (PS 2003), Versailles, France, 28 Sep.-2 Oct. 2003, pp. 92–94.

Abstract: In this paper we focus on providing service differentiation for variable length packets. Apart from the well-known OBS approach using differentiated offsets to introduce multiple service classes, we consider a look-ahead approach allowing for later arriving high priority packets to pre-empt earlier arrived low priority packets, and a slotted control approach. All approaches attain service differentiation without any resource reservation, and are of limited complexity, to minimize packet processing requirements. Through simulation, we assess the quality of the approaches under varying loads, buffer dimensions and QoS algorithm parameters. The main criterion used is the packet loss rate per service class.

Keywords: WDM, Optical Packet Switching, Optical Burst Switching, service differentiation, FDL, simulation

G.1 Introduction

The answer to the ever lasting hunger for bandwidth is being met by the deployment of (D)WDM networking [1]. To ensure efficient dealing with variable traffic patterns (both geographically and over time), Optical Packet Switching (OPS) has been devised. Profiting from cutting edge technology, it exploits fast optical switching tech-niques to offer better bandwidth granularity, efficiency and flexibility

than circuit-switched approaches. To relax some of the challenges involved (e.g. synchronization, high processing overhead), the asynchronous Optical Burst Switching concept using longer, variable length data units, has been devised. In this paper, we aim at finding a suit-able technique to provide service differentiation for optical switches dealing with variable length packets.

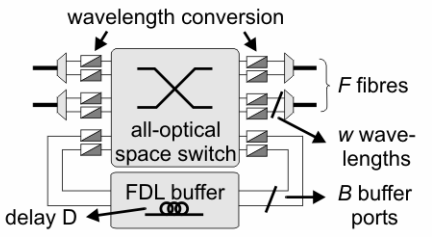


Figure G.1: Switch architecture under study.

The switch architecture we will focus on is a very generic one, and has been proposed e.g. in the European re-search project DAVID [2] for slotted OPS. Its functional architecture is sketched in Figure G.1: It has F input/output fibres, each carrying W wavelengths in WDM. The switch is capable of wavelength conversion, and exploits this capacity to solve contention [3]: packets arriving simultaneously and destined for the same outgoing fibre may be converted to other wavelengths to allow concurrent for-warding on the same output fibre. Where this does not suffice, an optical buffer can be used: B ports of the switching matrix are connected to Fibre Delay Lines (FDLs). In this paper, we assume that all recirculating buffer ports have the same delay D.

We will continue the paper as follows: in the next Section 1, we describe and discuss the compared QoS approaches. The methodology used is outlined in Section 3, followed by the results in Section 4. Conclusions are sum-marized in Section 5.

G.2 Approaches to QoS

The approaches we will compare are illustrated in Figure G.2 and comprise the following:

- ***Header offset differentiation ("offset"):*** This is OBS-JET where QoS differentiation is realized through giving higher priority packets a longer header offset [4]. This way, the arrival of high priority packets is known in the switch longer beforehand and reservations can be made before low priority packets.
- ***Look-ahead ("look"):*** The offset for different priority classes is the same but service differentiation is attained by assuming that the switch controller needs to make a decision only H after it has received the packet header. This can be achieved by having a fixed input buffer (e.g. by extending the one accounting for packet header proc-essing). Lower priority packets can be pre-empted by higher priority packets arriving up to a time H later.

- ***Slotted control ("slot"):*** Again without offset differentia-tion, the switch controller operates in a slotted mode. Each time-slot of duration T, we jointly make a decision for packets whose headers have arrived since the previous timeslot (just as a slotted OPS switch would do [5]). Thus, we can give precedence to high priority packets that ar-rived up to T later than low priority ones. This can be real-ized through a synchronous control channel (since elec-tronic header processing is assumed, this is realistic), or at least (electronic) synchronization in the switch controller and a sufficiently large input buffer (FDL) on the data path.

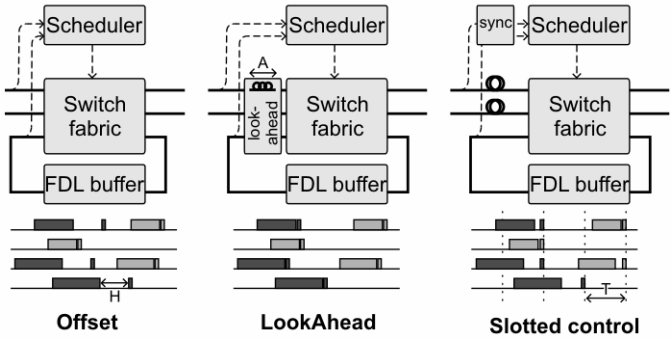


Figure G.2: The three QoS approaches we focus on.

Clearly, these approaches are not suitable for a large number of service classes: offsets, look-ahead times or slot sizes would need to be quite large and lead to unacceptable latency for high priority packets. However, since it is widely acknowledged that few (two or three) classes will be required in the core networks where switches as in Figure G.1 will be deployed, we believe the suggested ap-proaches are valid candidates towards service differentiation.

Note that these are not the only possible approaches to providing QoS in an IP-over-WDM scenario [6]. The QoS methods analysed in the following are all based on explicit indication of the class of service (priority), which applies to the whole packet. It all are cases without segmentation [7], without a priori resource reservation (to maximize resource utilization), without intentional dropping (opposed to e.g. [8]) and without the need to revoke anything sent out on output fibres. These properties ensure that the packet scheduling algorithm's complexity is quite limited and thus restricts time- and resource consuming packet processing.

The scheduling algorithms used all follow the PostRes [9] approach, meaning that no reservations are made for buffered packets until they leave the FDL and re-enter the switch (cf. otherwise, precautions have to be taken to avoid that buffering interferes with the differentiation mechanism [9]). The scheduler thus performs the same task for newly arriving packets as for recirculated ones: (i) use LAUC-VF [10] to find an available wavelength chan-nel on the output fiber the packet is destined for, (ii) if none is free, use LAUC to find the most suitable free FDL port, (iii) otherwise drop the packet.

G.3 Methodology

The parameters used for the node architecture of Figure G.1 are: F=6 i/o fibers, W=8 wavelengths per fiber, B=0..64 buffer ports. For the packet arrival process, we used Poisson arrivals and packet lengths based on a negative ex-ponential distribution: packets have a minimal length of L/2 and mean length L (the length minus L/2 follows a nega-tive exponential distribution). Traffic was uniformly spread over all output fibers. We considered two priority classes, where 60% of the packets were of the low priority and 40% high priority. In our simulations, we ignored header proc-essing times and thus set the basic header offset to zero. The offset-times clearly (see Figure G.2) depend on the QoS approach taken (zero or O for offset differentiation; zero for look-ahead; in the range [0,T) for slotted control).

To assess the major differences in performance between the three analyzed QoS approaches, we focus mainly on the loss rate achieved, i.e. the fraction of packets that is lost (which should be considerably lower for high priority traffic). Since this traffic is composed of variable length packets, we are also interested in the "fairness" within a single priority class: do all packets belonging to the same priority class experience the same QoS? It is indeed a well-known fact that short packets usually have lower chances of being dropped [11].

G.4 Results

G.4.1 Influence of the number of buffer ports

In a first experiment we focused on the efficiency in ex-ploiting the available buffer resources: we kept all parame-ters constant except the number of buffer ports B. Figure G.3 shows the evolution of the loss rate for an increasing number of recirculating buffer ports in case of a load of 0.8. The QoS parameters were the following: (i) differenti-ated offsets: high-priority offset O=2L; (ii) look-ahead: look-ahead time H=2L; (iii) slotted control: slot resolution T=2L. The buffer length was set to D=2L.

The slot approach achieves loss rates that are higher than the other two approaches, esp. when the number of recir-culating buffer ports increases. Whereas the difference in overall loss rates is limited, the loss rate for high priority packets is multiple orders of magnitude bigger than for offset or look-ahead. Still, even such a simple mechanism is able to provide clear service differentiation. When com-paring offset with look-ahead, the differences are small, with a slightly better performance of look-ahead.

From a fairness perspective, results (not plotted because of space limitation) confirmed our earlier statement that short packets are subject to lower loss rates. For the look-ahead strategy, this consistently favoring of short packets is more pronounced, due to the preemptive nature of the look-ahead strategy. Packets are scheduled upon arrival, and successively arriving packets of the same (or lower) priority are scheduled taking into account this schedule. For look-ahead, this schedule may be changed when later on (less than H) a higher priority packet arrives

destined for the same output fiber. Thus, the allocation of packets to fibers is not optimal, and the longer packets are the first to suffer from this effect.

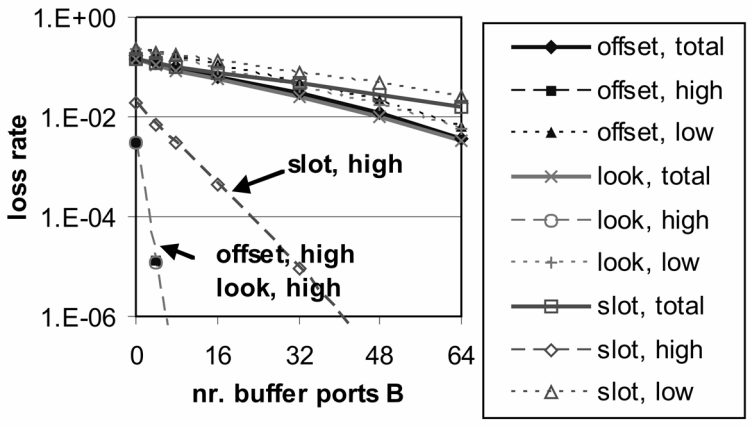


Figure G.3: Loss rates for increasing number of recirculating buffer ports B (load=0.8, F=6, W=8, 40% high priority traffic).

G.4.2 Influence of the class offset

Class separation depends on the parameter setting of the various QoS approaches: the actual parameter differs for each of the proposed approaches, but we will refer to it by "class offset". For the differentiated offset approach, it is the difference O in header offset between two successive priority classes. For look-ahead, it is the look-ahead delay H. For the slotted approach, it is the slot resolution T.

To assess the influence of the "class offset", we carried out simulations for a load of 0.8, and a buffer with B=8 ports and FDL length D=4L, leading to loss rates plotted in Figure G.4(a). The "class offset" was varied from L/2 to 4L.

By increasing the class-offset, the overall loss rate rises: high priority packets are considered more important, and their loss rate drops accordingly, but at the price of disre-garding more low priority packets. Clearly, there is a limit to the improvement: as soon as they don't "see" any low priority packets anymore, i.e. sufficient class isolation is achieved, the loss rate stabilizes. The point at which this isolation is achieved depends on the packet size distribu-tion. For the negative exponential packet length distribu-tion at hand, nearly complete isolation is reached for a "class offset" around 2L (less than 5% of the packets are longer than 2L for the distribution) for offset differentiation and somewhat earlier for look-ahead. This threshold is flattest for look-ahead. For the slotted approach, there seems to be much more room for improvement by increas-ing the "class offset" (thus slot size) further.

Note that to limit latency, we should aim at restraining the "class offset": since this is related to the average packet length L, this implies that packet lengths should be limited (i.e. at least those of high priority packets).

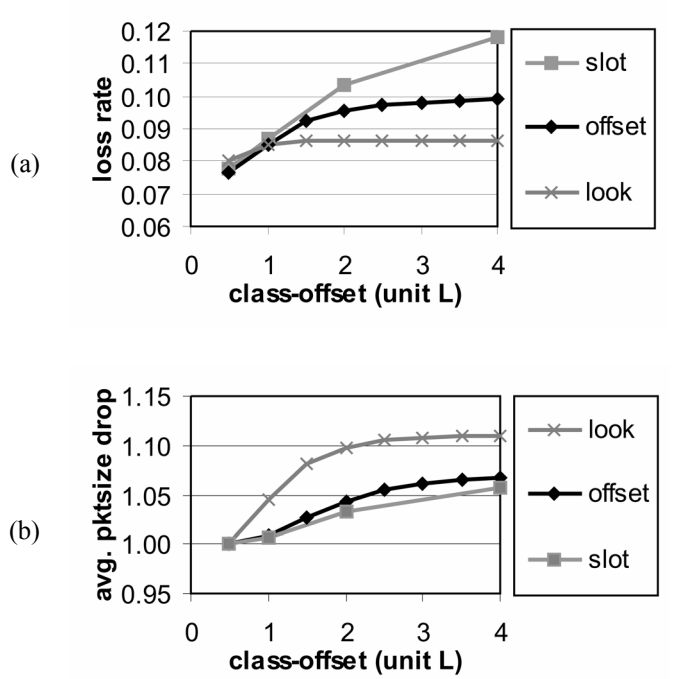


Figure G.4: (a) Overall loss rate and (b) avg. size of dropped packets for increasing class-offset O=H=T. (load=0.8, F=6, W=8, 40% high priority traffic; L=avg. packet length.).

To answer the question of intra-class fairness (how bad is the discrimination of long packets?) we plot in Figure G.4(b) the evolution of the average size of the packets dropped. With increasing "class offset", and thus class isolation, the un-fairness rises (only packets longer than average are dropped), but it is far more pronounced for look-ahead.

G.4.3 Influence of buffer delay

By varying the delay realised on the recirculating path through the FDLs, we unsurprisingly [9] find lower loss rates for increasing delay D. However, as for the "class offset", there is a floor: when the buffer is large enough to contain almost all packets (around 2L, see above), we see no further improvement of the loss rates.

G.4.4 Influence of the load

To verify that the approaches achieve sufficient isolation under all loads, we have analyzed the evolution of the loss rates for loads ranging from 0.1 to 0.9. When comparing the loss rates, the difference between both classes de-creases for increasing load for both the differentiated off-set and the look-ahead approach. (For

the bufferless case, it decreases from a factor ~700 to ~60). For the slotted control approach, the difference in loss rates is smaller, but the relative difference does not diminish that much with increasing loads. It is worth noting that even for the very simple slotted control mechanism, even the lowest priority packet loss rate stays below 10-3 for loads as high as 0.5.

G.5 Conclusion

We introduced and compared three scheduling ap-proaches that attain service differentiation for variable length packets in an optical packet switch with a recirculating FDL buffer. We compared the well-known differenti-ated offsets approach with a look-ahead approach that proved to achieve comparable loss rates. A slotted control approach which could simplify the burst scheduler imple-mentation achieves almost equal overall loss rates and delays, but does not achieve the same class separation. Still, for low to medium loads, with a moderate buffer, the performance attained by slotted control may be accept-able. From a fairness point of view, the look-ahead ap-proach most severely discriminates against longer bursts.

The robustness of each of the service differentiation mechanisms was assessed by varying the key parameters influencing their class isolation: (i) number of buffer ports, (ii) class offset, (iii) buffer delay, and (iv) load. It was con-cluded that (i) all approaches greatly benefit from adding buffer space, but the slotted approach does not succeed in exploiting it as efficiently as the others; (ii) the class offset needs to be set according to the burst length distribution (somewhat smaller for look-ahead); (iii) performance is improved if recirculation delay is sufficiently large; (iv) class isolation tends to slightly decrease for increasing loads.

Acknowledgment

This work has been supported by the European Commission through the IST-project DAVID (IST-1999-11387), and by the Flemish Government through the IWT GBOU-project “Optical Networking and Node Architectures”. C. Develder and J. Cheyns are supported as a Research Assistant of the Fund for Scientific Research – Flanders (F.W.O.–Vl.), Belgium.

References

[1] B. Mukherjee, “WDM optical communication networks: Pro-gress and challenges”, JSAC, vol. 18, no. 10, Oct. 2000.

[2] L. Dittmann, et al., “The European IST Project DAVID: a Viable Approach towards Optical Packet Switching”, to appear in JSAC.

[3] S. Yao, et al., “All-optical packet-switched networks: A Study of Contention-Resolution Schemes in an Irregular Mesh Network with Variable-Sized Packets”, Proc. OptiComm 2000.

[4] M. Yoo, C. Qiao, “Supporting multiple classes of services in IP over WDM

networks", Proc. Globecom 1999.
[5] C. Develder, et al., "Strategies for an FDL based feed-back buffer for an optical packet switch with QoS differentiation", Proc. COIN 2002.
[6] A. Kaheel, et al., "Quality-of-service mechanisms in IP-over-WDM networks", IEEE Comm. Mag., vol. 40, no. 12, Dec. 2002.
[7] A. Detti, et al., "Performance evaluation of a new technique for IP support in a WDM optical network: opti-cal composite burst switching (OCBS)", J. Lightwave Tech., vol. 20, no. 2, Feb. 2002.
[8] Y. Chen, M. Hamdi, D.H.K. Tsang, "Proportional QoS over OBS networks", Proc. Globecom 2001.
[9] C. Gauger, "Dimensioning of FDL Buffers for Optical Burst Switching Nodes", Proc. ONDM 2002.
[10] Y. Xiong, et al., "Control architec-ture in optical burst-switched WDM networks", JSAC, vol. 18, no. 10, Oct. 2000
[11] K. Dolzer, et al., "Evaluation of reservation mechanisms for optical burst switching", AEÜ Int. J. Electron. Commun., vol. 55, no. 1, Jan. 2001.

Glossary

(D)WDM	(Dense) Wavelength Division Multiplexing
FDL	Fibre Delay Line
LAUC	Latest Available Unused Channel
LAUC-VF	Latest Available Unused Channel with Void Filling
OBS	Optical Burst Switching
OBS-JET	Optical Burst Switching with Just Enough Time
OPS	Optical Packet Switching
QoS	Quality of Service

Appendix H

Routing strategies to minimize packet loss in an optical packet switched network with recirculating FDL buffers*

E. Baert, C. Develder, D. Colle, M. Pickavet, P. Demeester

To appear in Photonic Network Communications (PNET), vol. 7, no. 2, Mar. 2003.

Abstract: The major goal of Optical Packet Switching (OPS) is to match switching technology to the huge capacities provided by (D)WDM. We study optical packet switches with recirculating Fiber Delay Line (FDL) buffers. Through simulation, we have assessed the logical performance of a single Optical Packet Router (OPR), focusing on Packet Loss Rate (PLR). By verifying that our scheduling algorithm does not alter the traffic profile characteristics from in- to output, we illustrate how the single node results can be used to assess network-wide performance.

---

* This work has been supported by the European Commission through the IST project DAVID (IST-1999-11742) and the Flemish Government through the IWT GBOU-project "Optical Networking and Node Architectures". The work of C. Develder was supported by the Fund for Scientific Research — Flanders (FWO-Vl.). D. Colle would like to thank the IWT for its financial support for his postdoctoral grant.

The authors are with the Dept. of Information Technology (INTEC), Ghent University – IMEC, Sint-Pietersnieuwstraat 41, 9000 Ghent, Belgium (tel: +32-9-264.99.70; fax: +32-9-264.99.60; e-mail: {elise.baert, chris.develder, didier.colle, mario.pickavet, piet.demeester}@intec.UGent.be).

We use the capability of assessing end-to-end PLRs to develop network-wide routing algorithms designed to minimize the maximal PLR occurring in the network. In case studies on pan-European networks, we first compare two algorithm variants and thereafter we compare the PLR-based routing algorithm with both load balancing and shortest path routing. While load balancing achieves PLRs that are multiple orders of magnitude lower than shortest path routing, the PLR-based algorithm can reach PLRs up to two orders of magnitude better. The improvement in PLR comes at the price of only a small increase in used bandwidth (a few percent).

Subsequently we show that the discussed PLR-based routing algorithm can be easily extended to multiple priorities. By introducing multiple priorities we can keep the loss rates for high priority traffic very low. However, it may lead to an increase of the obtained minimal max-PLR value for low priority traffic. But as we prove this increase to be limited, the cost of introducing multiple priorities is small.

Keywords: Optical Communication, Optical Packet Switching, Routing, Simulation, Priorities.

H.1 Introduction

The rapidly increasing demand for bandwidth in telecommunication networks is met by the huge capacities provided by (D)WDM. The first step in moving from point-to-point systems to real optical networking encompasses a circuit-switched approach [1]. This approach however has difficulties dealing with highly variable traffic (both in volume and traffic pattern). Optical Packet Switching (OPS, [2-4]) offers a longer-term solution that provides better bandwidth granularity, efficiency, and flexibility.

We consider a core OPS network consisting of so-called Optical Packet Routers (OPRs) that synchronously switch fixed-length packets (as opposed to asynchronous switching of the variable-sized bursts in e.g., Optical Burst Switching, OBS [5-6]).

The major problem that needs to be addressed in any packet switched concept is contention resolution: what if multiple packets need to be switched simultaneously to the same output port of the switch? In an OPS environment, three different techniques can be identified to solve this: (i) wavelength conversion, (ii) buffering, and (iii) deflection routing. The use of wavelength conversion implies that multiple packets will be switched to the same outgoing fiber using WDM, where some of them may be forwarded on another wavelength than they entered the switch. It has been shown that this exploitation of the wavelength domain greatly reduces the need for buffering (e.g., [3], [7] and [8]). Still, using wavelength conversion alone, contention still can arise, which can most straightforwardly be solved through the use of buffering of some kind. However, since buffering in optics implies the use of Fiber Delay Lines (FDLs), also deflection routing has been proposed: some of the contending packets are sent to a “wrong” output port, forcing them to make a detour, in the hope to avoid the congested network part. Clearly, this only works when enough free capacity is available in the other parts of the network, thus for reasonably low overall network loads. The soundness of this intuitive insight has been confirmed by a comparison of the three approaches to contention

resolution, showing that deflection routing is outperformed by the other two techniques [9-10].

So, to ensure the efficiency of the OPS network, and to obtain low Packet Loss Rates (PLRs) , the OPR considered in this paper will use wavelength conversion and an optical feedback buffer [11] with FDLs. The logical structure of the OPR, proposed within the framework of the European research project DAVID [12], is depicted in Figure H.1 (for the physical structure of the switching matrix proposed in DAVID, see [13]).

The OPR operates in a slotted way: at every slot time, it inspects packets arriving at its input ports, and subsequently decides which packets to forward (to the output ports or the feedback buffer) or to drop. This decision is taken by following a fixed procedure, comprising two phases: (i) for each output fiber of the OPR, elect at most w packets to be forwarded directly, (ii) from the remaining packets, elect at most B to put in the buffer; any other packet will be lost. Election of packets for forwarding and buffering is based on two criteria: the priority attached to the service class the packet belongs to, and the time it already spent in the OPR. Service differentiation is based on a pure priority scheme: packets of a higher priority class are given precedence over lower priority ones. Within the set of packets with the same priority, the one which has spent the longest time in the OPR is favored. Among multiple packets sharing the same priority and with the same time spent in the OPR, one is selected randomly.

For the FDL-buffer used, two cases are considered. In the case labelled “fix”, we use the same FDL length of a single slot for each of the ports. The “incr” case uses a different FDL length for each of the wavelength ports: for the B buffer ports, lengths of 1, 2, 3…,B slots are used. When a buffer with multiple FDL lengths is adopted (incr), the B buffer ports are no longer equivalent. Thus, the election procedure of packets to direct to the buffer needs to determine what FDL length to use. An obvious strategy could be to simply use the smallest FDL length for which no other packet has been elected yet. We label this approach as MinDelay. This strategy does not take into account packets put into the FDL buffer at earlier times. A more intelligent approach, denoted as Balance, inspects the buffer contents to choose an appropriate FDL length. For each available FDL length L, we count the total number of packets N_L already present in the complete buffer, destined for the same output fiber, that will leave the buffer at time now+L slots. We choose the free buffer port with FDL length L having the smallest count N_L. Thus, the Balance strategy tries to minimize the number of packets, destined for the same output fiber, leaving the optical buffer at the same time.

A detailed analysis of the performance of a single OPR, for different traffic profiles, focusing on the buffer structure and scheduling strategies can be found in [14-15]. Alternative switch structures are discussed for instance in [16].

In this paper, we discuss routing strategies to minimize the PLR in a meshed network of OPRs. In contrast with [17] where routing is only performed with the OSPF (Open Shortest Path First, [18]) protocol, we will adopt an MPLS (Multi Protocol Label Switching, [19]) approach, where routing and forwarding are separated and other than pure destination-based routing can be used, which results in a greater flexibility.

In Section H.2, we explain how we assess the network-wide PLR based on simulation results for a single OPR. The routing algorithms based on PLR-estimation are discussed in Section H.3. In the subsequent Sections H.4 and H.5, we evaluate the performance of the PLR-based routing algorithm through case studies on pan-European networks. In Section H.4, we compare two PLR-based variants, and we continue in Section H.5 by comparing the developed PLR-based routing algorithm with less complex load balancing and straightforward shortest path routing. In the next Section H.6, we discuss the influence of introducing multiple priorities in the PLR-based routing algorithm (in all previous simulations all traffic was of one single priority). The paper is concluded in Section H.7.

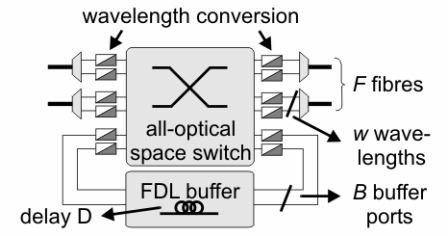


Figure H.1: Optical Packet Router architecture.

H.2 Assessing network-wide performance

To assess network-wide performance, especially in terms of PLR, a straightforward solution could be to simulate the network as a whole, using e.g., the simulation tool we have built to assess single node performance [14]. However, since this is quite time-consuming and therefore prohibitive for the iterative routing strategies we propose in the following Section H.3, we assess the PLR of each individual OPR using an approximation of the PLR by an analytical formula.

In Section H.2.1, we first describe an analysis of the traffic pattern at the OPR's outputs which shows that the statistical properties of the input traffic are not significantly altered, thus validating the approach of establishing end-to-end performance through analyzing each OPR in turn. In the following Subsection H.2.2, we heuristically derive an analytical formula for the PLR inflicted at a single OPR, this by performing extensive simulations of a single OPR under varying traffic conditions, thereby using the single node simulator described in [14]. This formula will be used to quickly calculate the PLRs in the different iterations of the routing algorithm.

H.2.1 Cascadeability of the single node model.

The studies presented in [14] and [15] focused on the performance of a single node under various traffic profiles. However, in real life, such Optical Packet Routers will be interconnected in a (backbone) network. This implies that the output of a particular OPR will be the input of another one. In this context, the results of the single node studies are useful only if the input traffic profile can be assumed to be

similar for all OPRs (in terms of type of packet arrival distribution, but clearly not necessarily in terms of average load). In particular the question arises whether the profile of the traffic on an output fiber of an OPR is similar to that offered at its inputs. This is the question we address in this section.

To compare the traffic profiles at the inputs and outputs of the OPR, we have traced the number of packets arriving at each of the input ports and leaving on each of the output ports, and this for each priority. Three traffic source types were considered. The first is the well-known Poisson process. The GeoOnOff source generates bursty trains of packets: an on/off source with geometrically distributed lengths of both on- and off-periods. Self-similar traffic labelled ParetoOnOff was generated using on/off sources with Pareto distributed on- and off-times [20]. During the on-periods a packet is sent every timeslot.

The traffic was divided in three priority classes: 50% highest, 25% middle and 25% lowest priority traffic. For the analysis of the traces, we have focused on the number of packets n_τ that arrive in the same timeslot τ on a particular input or output fiber. To characterize the packet arrival process, the two foremost important parameters are the probability distribution of n_τ and the correlation between n_τ and $n_{\tau+lag}$ (i.e., correlation between the number of packets arriving in timeslots spaced by lag slot-times).

When switching packets from input ports to output ports, the OPR will impact the traffic profile in two ways: (i) by dropping packets, and (ii) by delaying packets. Dropping packets will mainly impact the distribution of n_τ. Delaying packets will mainly impact the correlation between the number of packets in successive timeslots: buffering will spread packets in time (especially the Balance strategy for the incr buffer structure, see introduction), thus potentially lowering the correlation of the input traffic. In Figure H.2, we have plotted, for Poisson traffic, the comparison of n_τ histograms for input and output port traffic, for the incr buffer structure with $B = 64$ ports, using the Balance strategy. The accompanying lag correlations between n_τ and $n_{\tau+lag}$ are presented in Figure H.3 Similar curves have been analyzed for other buffer structures and other traffic profiles leading to the same conclusions.

Inspection of the histograms confirms our intuitive expectations: for the priority classes suffering from higher drop probabilities (i.e., the packets with the lowest priority), the effect of these drops is a lowering of the probabilities for higher n_τ values and corresponding rise for lower n_τ, leading to a slightly lower average number of simultaneous packet arrivals for these classes (Figure H.2, (c)-(d)). For the overall number of simultaneous arrivals (Figure H.2, (a)), the losses result in a slight lowering of the probability of 32 simultaneous arrivals, and thus a relative increase for lower n_τ values. The correlation plots of Figure H.3 show that for each individual priority class, the correlation structure of the input traffic is not significantly impacted by the OPR. For the packet counts over all priority classes, we notice a reduction of the correlation for the OnOff traffic types due to the aforementioned spreading in time achieved through buffering.

From our probability and correlation analysis, we may conclude that it is safe to describe the input and output traffic profiles by the same model, at least to estimate packet loss rates. Indeed, the correlation structure is not significantly impacted, and

nor are the probabilities of n_τ simultaneous packet arrivals. This result allows to estimate PLRs in different OPRs interconnected in a network independently. Such network-wide PLR estimation can be used to make routing decisions, as discussed in Section H.3.

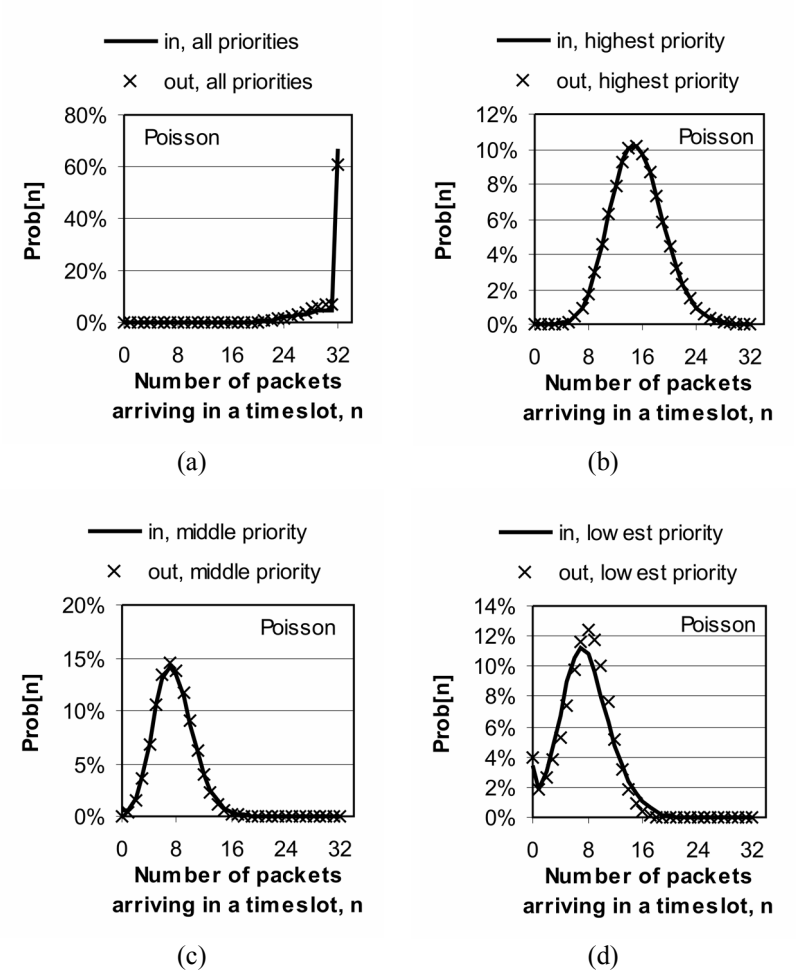


Figure H.2: Comparison of the histograms of the number of packets per timeslot (w = 32) arriving on a particular input port (solid lines) and leaving the OPR on a particular output port (marked dashed lines).

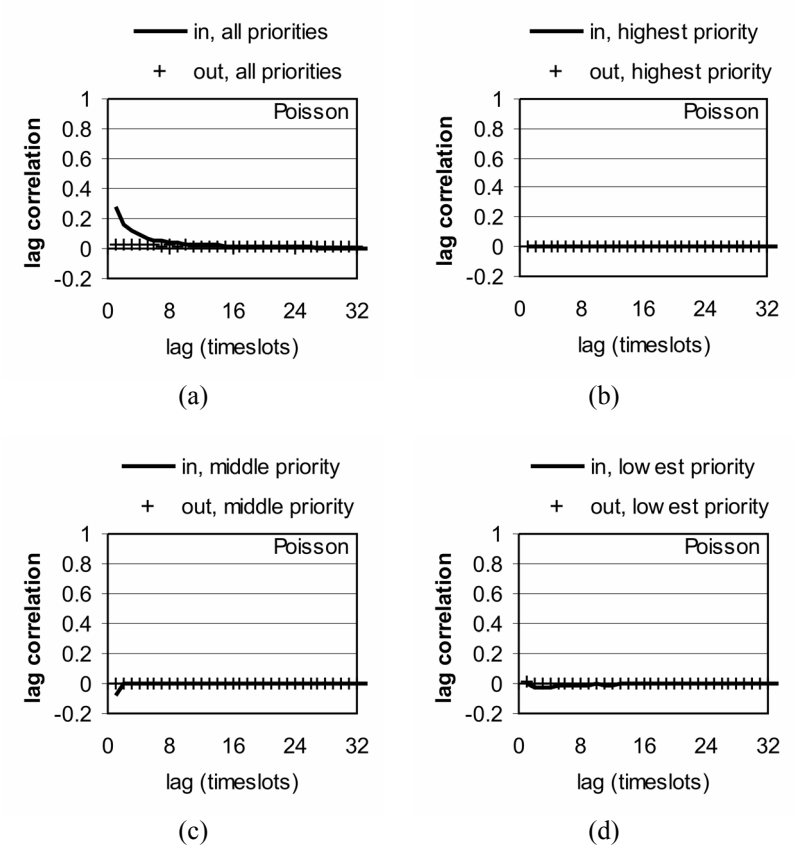


Figure H.3: Comparison of the correlation between the number of packets per timeslot (w = 32) arriving in timeslots τ and τ+lag on a particular input port (solid lines) and leaving the OPR on a particular output port (marked dashed lines).

H.2.2 Capturing the single node performance in a simple formula.

The routing algorithms, proposed in the next Section H.3, are of an iterative nature, and thus we need a reasonably fast method to estimate the PLRs in all nodes for each iteration step. Therefore, we will use an approximation by an analytical formula. The form of the chosen formula, and the parameter values, will be discussed in this section.

The objective is to find a formula predicting the loss on a certain output fiber, given the load offered to the OPR. Under the assumption that the traffic on such an output fiber is an aggregate of traffic coming from multiple input ports, the main factor impacting the loss for traffic passing through the OPR to this fiber will be the

offered load for that output port. However, since the OPR's recirculating buffer is shared among all outputs, the loads on other output ports will also affect the PLR. To capture this correlation between packet loss rates on different output fibers, we propose a formula of the form given in Eq. (1) where L_f stands for the load on output fiber f (ranging from 0 to F–1, where F is the number of output fibers of the switch). The measure L_α is an average output load giving more weight to the higher loads, since these are the only ones that will use the recirculating buffer intensely and thus impact the losses on other fibers.

$$\mathrm{PLR}(L_f \mid L_0 \ldots L_{F-1}) \approx \mathrm{g}(L_f, L_\alpha) \text{ , with } L_\alpha = \sqrt[\alpha]{\frac{\sum_{i=0}^{F-1} L_i{}^{\alpha}}{F}} \tag{1}$$

To get an idea on the analytical form to use for the function g, we have performed a series of simulations for each combination of output loads L_f, where each load was taken from the set {0, 0.7, 0.8, 0.9}. We analyzed the PLR curves for these given output load combinations for two sorts of traffic matrices. In a first traffic matrix, named symm, we considered the case where each input port equally contributed to the load L_f on each output fiber f (each input fiber i contributes part L_f / F). The second traffic matrix type, denoted asym, focused on asymmetrical contributions from each of the input ports to the load on output f. The formulas used to set the load of the traffic generated for a particular (input,output)-pair in case of asym is given in Eq. (2).

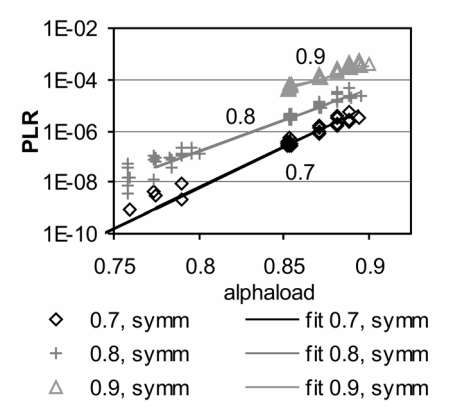
$$L_{i,f} = \begin{cases} L_f \,/\, 2^{1+(i-f) \bmod F} & , \quad i \neq f \bmod F \\ L_f \,/\, 2^{F-1} & , \quad i = f \bmod F \end{cases} \tag{2}$$

The resulting plots of the PLR in function of L_α for each of the values used for the output loads are plotted in Figure H.4 for the case of an OPR with F = 6 input/output fibers, w = 32 wavelengths per fiber, and a fix buffer structure with B = 32 recirculating buffer ports using the MinDelay strategy. By varying α, we observed for reasonably large values a strong separation of the measure points for different output load values L_f and small spreading for the measure points for same (L_f, L_α)-pairs. The resulting points for the same L_f values almost fall onto the same straight line in a logarithmic plot. This observation led to the proposal of the formula given by Eq. (3), with a factor and an exponent depending only on L_f. The meaning of the factor $p(L_f)$ in that formula is the packet loss rate for $L_\alpha = L_f$, thus the case where all output ports have the same load of L_f. The packet loss rate in case of a uniform traffic matrix is plotted in Figure H.5, along with the outcome of the analytical formula we proposed for this packet loss rate as given in Eq. (4). The exponent $r(L_f)$ was chosen to be linear in L_f, as in Eq. (5). All the parameters in formulas (1, 3–5) were fit using the method of least squares, with the results listed in Table H.1. The plots show that for these values, the correspondence with the simulation results is very satisfactory.

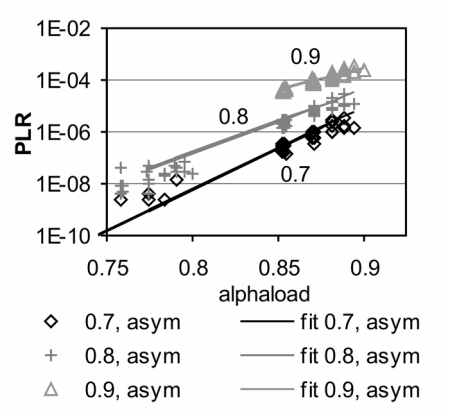
$$PLR(L_f \mid L_0 \ldots L_{F-1}) = p(L_f) \cdot 10^{r(L_f)\cdot(L_\alpha - L_f)} \tag{3}$$

$$p(L_f) = PLR(L_f \mid uniform) = a \cdot {L_f}^b \cdot 10^{c \cdot L_f} \tag{4}$$

$$r(L_f) = u \cdot L_f + v \tag{5}$$



(c)



(d)

Figure H.4: Packet loss rates and fits with analytical formula in (a) for the symm case where each input port contributes in the same way to the output load on a particular output port, and in (b) for the asym case with asymmetrical contributions by the input ports.

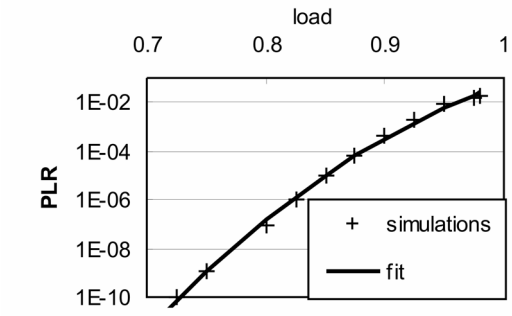


Figure H.5: Packet loss rates in case of a uniform traffic matrix for increasing loads.

Table H.1: Parameter values of formulas (1, 3-5).

Parameter	*Value*	*Parameter*	*Value*
α	32.97	*a*	62.59
u	71.58	*b*	189.15
v	-82.77	*c*	-63.81

H.3 Routing algorithms

The PLR formula explained above (Section H.2.2) enables us to assess network-wide PLR within a reasonable time. We can use this PLR-estimation technique to drive routing algorithms aiming at minimizing the maximal PLR that occurs for a given traffic demand matrix. In this section we are going to elaborate on this routing problem and the developed routing algorithms. In Section H.3.1 we give a detailed problem description, followed in Section H.3.2 by an in-depth discussion of the routing algorithms, which will set up paths between sources and destinations, starting from an initial shortest path routing and then trying to achieve lower PLRs by rerouting some paths.

H.3.1 The routing problem

The following optimization problem is considered: given the network topology (nodes and links) and capacity; the cost for nodes and links (in function of capacity), the node model (packet loss rate in terms of load), the traffic demand matrix (stating not only the required capacity between two OPRs, but also an upper bound giving the tolerable PLR for this demand), and the maximal tolerable packet loss, we want to find the (cheapest) routes fulfilling the demands and their maximal PLR requirements. The problem is illustrated in Figure H.6.

More specifically, we here want to route the demands in the network so that the maximum PLR (this is, considering the PLR of the traffic on each link of the network, the maximum of these PLRs) occurring in the network is as low as possible, and this as cheap as possible.

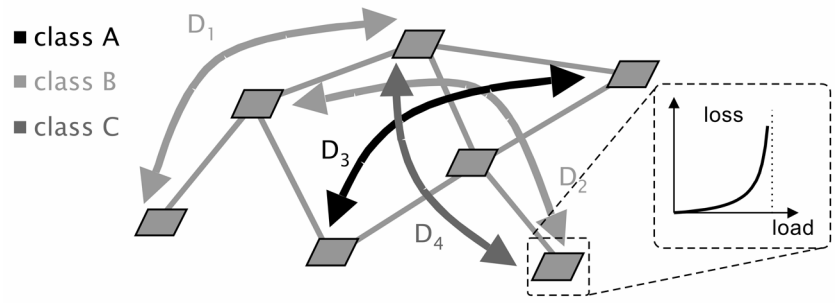


Figure H.6: Routing problem.

The general routing problem is defined as follows. The network can be represented as a directed graph G = (N,A) whose nodes and arcs represent optical packet routers and the links between them. Each arc a has a capacity c(a) which is a measure for the amount of traffic flow it can take. In addition to the dimensioned network, a given demand matrix D for each pair (s,t) of nodes gives the traffic flow from source s to destination t. Many of the entries of D may be zero, and in particular, D(s,t) should be zero if there is no path from s to t. We also have PLR requirements which state the maximum PLRs the demands may have. In our case studies we do not have specific values we want to obtain, we just iterate until we reach the lowest max-PLR possible. So, the objective is to minimize the max-PLR.

- ***Given:***
  - G = (N,A) is the considered network with nodes n ∈ N and arcs a ∈ A
  - c(a) is the capacity of arc a
  - D(s,t) is the demand to be set from node s to node t
  - $x_{st}(a)$ is a zero-one variable and is equal to 1 ⇔ arc a belongs to the selected path for D(s,t)
  - $f_{st}(a)$ is the flow from the demand from s to t over arc a = $D(s,t)\cdot x_{st}(a)$
  - L_a is the load on arc a = $\dfrac{\sum_{s,t} f_{st}(a)}{c(a)}$
  - PLRa is the PLR on arc a = PLR(La|L0… L_f -1)
  - I(n) is the from-incidence of node n = set of all arcs leaving node n
  - I'(n) is the to-incidence of node n = set of all arcs arriving at node n
- ***Objective:***
  - minimize MAX(PLR_a) ∀a
- ***Constraints:***
  - $\sum_{a \in I(n)} x_{st}(a) \le 1, \quad \forall n \in N, \forall s,t \in N$
  - $L_a \le 1, \quad \forall a \in A$
  - $$\sum_{a \in I(n)} f_{st}(a) - \sum_{a \in I'(n)} f_{st}(v) = \begin{cases} D(s,t) \text{ for } n = s & (i) \\ 0 & (ii) \\ -D(s,t) \text{ for } n = t & (iii) \end{cases}$$

H.3.2 The routing algorithm

The heuristic algorithms we developed are iterative, and consist of multiple phases, as depicted in Figure H.7. They start with calculating the shortest paths for each (source, destination)-pair. The subsequent phases will reroute some of these paths to lower the max-PLR. A Zoom-In philosophy is used [21], starting with a first phase that takes a global perspective, after which the result is refined in a second, local phase.

The global phase (Figure H.8) considers the network as a whole by giving penalties to links exhibiting high PLRs and recalculating the routes for all demands. When changing these penalties does not lower the max-PLR anymore, we go on to the second, local phase where only a single path is re-routed in each step.

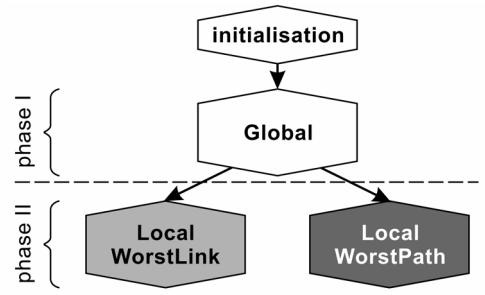


Figure H.7: Phases of the algorithm.

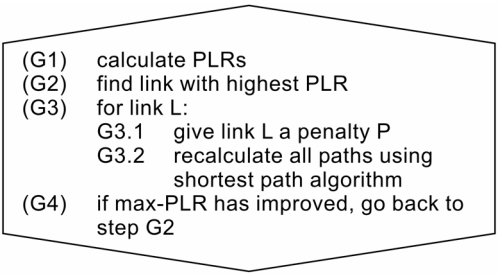


Figure H.8: Routing algorithm: algorithmic description of global phase.

In the local phase —which was added because the first phase only gives a small max-PLR reduction (see Section H.4)— we compared two variants: WorstLink (Figure H.9) and WorstPath (Figure H.10). The former zooms in on the link with the max-PLR, and tries to reroute one by one (in a random order) all demands crossing this link by giving this link a high penalty. WorstPath focuses on a particular path, starting with the demand with the worst end-to-end PLR. When rerouting is successful, WorstPath continues with the path that now has the worst end-to-end PLR; but when, after a fixed number X of attempts to reroute this demand, this lowering of the max-PLR still is unsuccessful, the algorithm ignores this demand and moves on to the next demand in line (sorted on end-to-end PLR in decreasing order). Rerouting a demand in WorstPath is done by giving penalties to each of the links of the route currently followed for that particular demand (starting with the one exhibiting the highest PLR), in order to relieve the links most heavily suffering from loss.

The algorithm stops when all routes have PLRs below the requested upper bounds, or when the max-PLR cannot be improved any more (for the WorstLink variant this happens when, after changing the paths over the link with the worst PLR, this link still is the one with the max-PLR; in WorstPath this is when all paths have been tried X times without success).

Since the PLR is strongly related to the load on the links (see Section H.2.2), one can expect that a load balancing algorithm —aiming at lowering the maximal load on each of the network's links— will also achieve significant PLR reduction compared to shortest path routing. Clearly, the advantage of such a load balancing

approach is that the routing algorithm does not require any PLR-estimation. The results presented in the next section show that the more complex PLR-based approach can reach PLRs more than an order of magnitude lower than with load balancing.

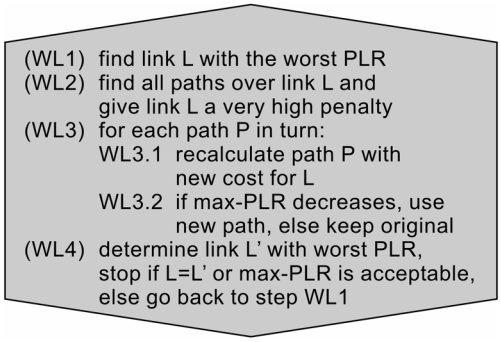


Figure H.9: Routing algorithm: algorithmic description of local phase – WorstLink variant.

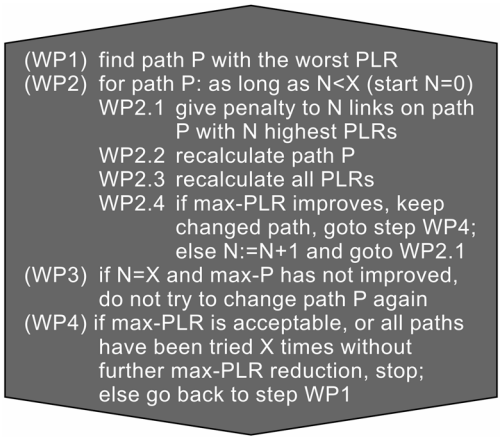


Figure H.10: Routing algorithm: algorithmic description of local phase – WorstPath variant.

In the algorithm descriptions above, we only considered traffic with one priority, but extension to traffic with multiple priorities is straightforward: the different phases can be run consecutively for all classes/priorities (starting with the class with the most strict packet loss demand, as lower priority traffic does not influence higher priority traffic). In Section H.6 results of simulations with traffic of multiple priorities are presented.

Note that our heuristic is a local search technique, the descent method: a path-change is only accepted when the objective function (i.e., lower the max-PLR value) improves, which avoids cycling but could lead to a solution that is a local minimum far from the optimal solution. To prevent from ending up with such a local

minimum, we perform a couple of different simulations (in average five simulations per demand matrix) for the same inputs (the same network, capacity, costs, demand matrix), but for a different (random) order of the demands. We observe that we indeed sometimes become another ultimate max-PLR when routing the demands in a different order. If this happens, we keep the lowest max-PLR value.

H.4 Routing algorithm performance

To evaluate the performance of our routing algorithms, we carried out case studies on two meshed pan-European networks, whose topological characteristics are summarized in Table H.2. The first, illustrated in Figure H.11, connects 19 European cities. A joint effort from the IST project LION and COST action 266 resulted in topologies for a pan-European fiber-optic network [22]. We used such a network, sketched in Figure H.12, as a second test-case. It is somewhat sparser than the first: it has the same number of links, but interconnects 27 cities.

Given these topologies and the traffic demand matrices (bandwidth and PLR requirements), we want to find routes for the traffic that satisfy these demands. In the following results the networks comprise bi-directional links (one direction equals one fiber) and costs are in function of fiber length and capacity. Each of the fibers consists of 32 wavelengths and the buffer contains 32 FDLs, all of a length of one timeslot (a fix buffer, see introduction). In this section we assume Poisson traffic with a single priority (for which we want to determine the lowest PLR possible). We studied various demand patterns, both uniform and non-uniform.

A first set of simulations was used to compare the two alternatives (WorstLink and WorstPath) of the local part of the algorithm. These simulations are described in this section.

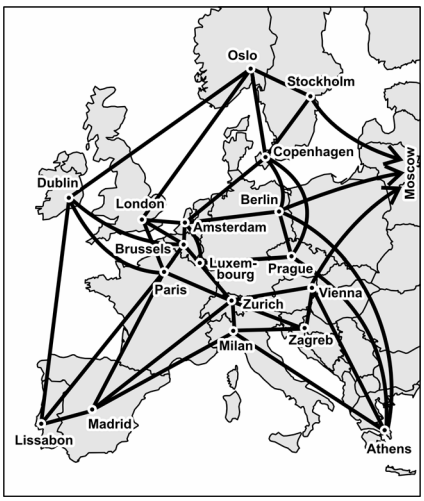


Figure H.11: The dense 19-node network.

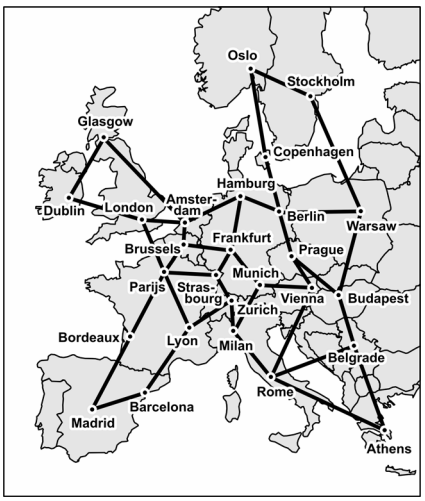


Figure H.12: The sparse 27-node network.

Table H.2: Network scenarios.

Parameter	*Network 1 (Dense)*	*Network 2 (Sparser)*
Nr. of nodes	19	27
Nr. of links	40	40
Avg. node degree	4.21	2.93
Min. node degree	3	2
Max. node degree	6	5

For the two variants of the algorithm, Figure H.13 shows the evolution of the max-PLR for a random (non-uniform) demand between all nodes from the second pan-European network, in function of the number of iterations, where a single iteration is an (attempt to) change a single path. Since we adopt a heuristic descent method, the maximal PLR does not increase for successive iterations.

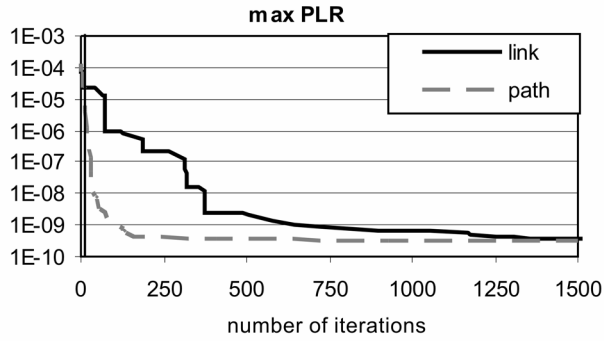


Figure H.13: Evolution of max-PLR (in function of number of iterations) for a random demand on the second pan-European network. The full line is the WorstLink version of the algorithm, the dashed line the WorstPath version. Only the first 1500 iterations are shown, since after that number the max-PLR doesn't change anymore. The vertical line on the graph marks where the global phase of the algorithm ends and the local phase begins.

When we compare the WorstLink and WorstPath version, we notice that the two versions give the same final result, for the plotted case, but WorstPath reaches this minimum max-PLR faster, i.e., after fewer iterations, than WorstLink. This observation is valid for the majority of the simulated cases. The reason why WorstPath reaches the minimum faster is that it always tries to change the worst path (i.e., the path with the worst end-to-end PLR) first. In WorstLink this is not necessarily so: the algorithm tries to change all the paths over the worst link in a random order, even if these paths are not the worst paths (anymore — if a path already has been changed and now another link suffers from the worst PLR).

When looking at the number of iterations the algorithm needs to end (see Table H.3), WorstPath always needs more iterations to stop. WorstPath tries, if it is not successful, to change every path a fixed number of X times but WorstLink stops when all paths over the worst link were tried and this link has still the max-PLR over the complete network, so it does not try to change all possible paths.

In terms of the routes followed for each of the demands, we noticed that unsurprisingly this sometimes proves to be different for the two versions of the algorithm. This was to be expected, since the order in which the demands are (re)routed differs between the WorstLink and WorstPath variants. In terms of overall link capacity used, WorstPath is somewhat worse of (i.e., between 2% and 5% more used capacity) than WorstLink.

Although we see in most cases that WorstLink and WorstPath give the same final result and that WorstPath reaches this minimum max-PLR faster than WorstLink, also a few exceptions to this general rule were observed. A first exception that occurs is that WorstLink sometimes reaches the (same) minimum faster than WorstPath. A more important exception is illustrated below. Figure H.14 shows simulation results for the dense pan-European network: it plots the evolution of the max-PLR for an increasing number of iterations for another non-uniform demand.

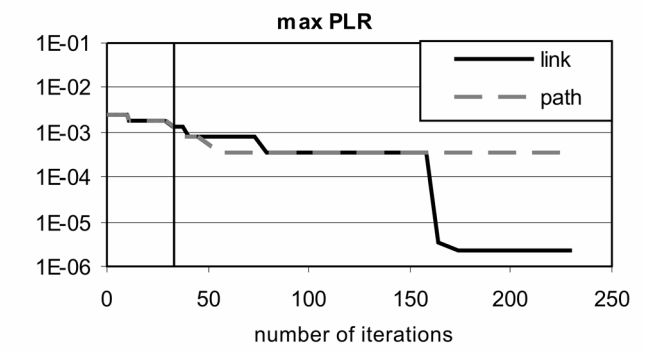


Figure H.14: Evolution of max-PLR (in function of number of iterations) for a random demand on the first pan-European network: we omitted the iterations beyond the first 250 because no further lowering of the max-PLR was attained. The vertical line marks the start of the local phase of the algorithm.

Comparing the WorstPath and WorstLink variants, we see that the WorstPath version never reaches the same minimum as WorstLink, so here the WorstLink version gives a much better solution (i.e., lower max-PLR) than WorstPath. An explanation for this worse performance of WorstPath is that after trying to change a path X times without success, this path is never tried again. However, it is possible that changing this path at a later stage, after having changed a few other paths (with a lower PLR), would lead to a lower max-PLR. The WorstLink algorithm indeed can re-consider a path tried before in a next iteration, when the worst link of that path becomes again link with overall max-PLR (this may happen after another link

has exhibited the max-PLR which has successfully been lowered by rerouting some path(s)). If and when this difference in final result occurs, depends on the network, the demands, and the randomly chosen order in which the demands are routed.

In terms of capacity-use, these exceptional cases with respect to best PLR performance, do obey the formerly reached conclusion on capacity-use: we also notice a higher capacity-use with WorstPath.

We can thus summarize the conclusions of our WorstPath versus WorstLink algorithm variants as follows: WorstLink always achieves the lowest PLR and it also has the lowest penalty (in terms of extra capacity-use compared to shortest path routing). In the following, we will therefore apply the WorstLink variant.

Table H.3: Number of iterations the algorithm needs to stop for WorstLink and WorstPath (for different demands).

	nr. of iterations to end	
scenario	*WorstLink*	*WorstPath*
2-uni 0,38	320	2013
2-rt 0,36	1789	2564
1-uni 0.64	432	1872
1-S 0.54	186	733

H.5 Performance of PLR-based routing compared to shortest path routing and load balancing

To evaluate the developed PLR-based algorithms, we compared the max-PLR found with the WorstLink variant with the max-PLR for the initial shortest path routing based on link costs. Clearly, if the reduction in PLR would amount to only a small fraction of the original value, the usefulness of PLR-based routing would be marginal. To measure the cost of using a more intelligent routing strategy, we look at the price paid in terms of excess capacity-use when using the PLR-based routing.

Secondly, to assess the importance of estimating the PLR accurately, we also used a load balancing algorithm to find the routes to follow. Indeed, since the main factor impacting the PLR on a link is the load generated for it (see Section H.2.2), we expect that by load balancing and thus limiting the loads, we can achieve already a substantial reduction of PLR compared to shortest path routing. The load balancing algorithm used was identical to the WorstLink algorithm described above, but using the link load as cost measure to minimize: in each iteration, we strive at lowering the maximal load on each of the network's links (i.e., lower max-load instead of max-PLR).

Figure H.15 below shows, for different demands on the two studied networks (on the X-axis), the max-PLR values reached: (i) with the WorstLink variant of the

PLR-based routing algorithm, (ii) with load balancing, and (iii) with shortest path routing. The type of demand is indicated on the X-axis: uni stands for uniform, rl for real non-uniform —based on the traffic forecast in [22]—, S for random demands all of the same order, and L for random patterns where a few demands are of a larger order; the 1 and 2 refer to the 19-node and the 27-node network, respectively; the last number in each label on the X-axis is the overall mean network load when using shortest path routing (range [0,1]).

Table H.4 shows the numerical values of the max-PLR values plotted in Figure H.15, the ratio shortest path routing/load balancing and the ratio load balancing/PLR-based.

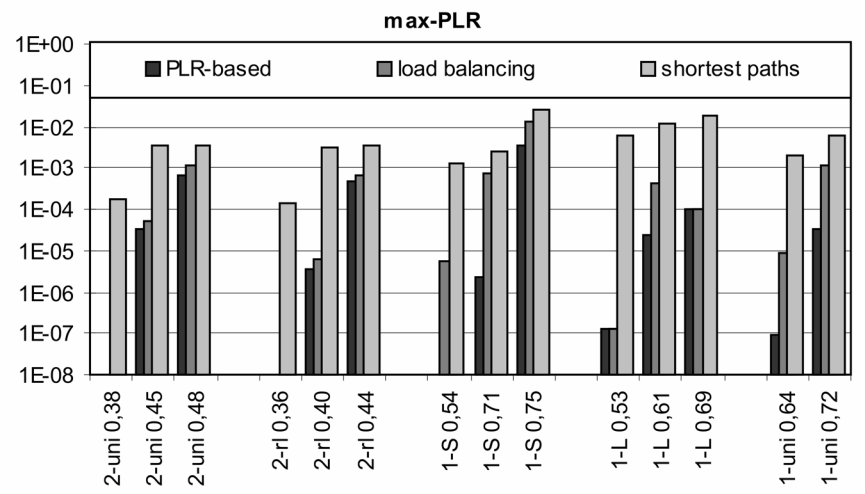


Figure H.15: Comparison of max-PLR reached with PLR-based routing, shortest path routing, and load balancing for different demands.

The results presented in Figure H.15 and Table H.4 show that the max-PLR obtained with the PLR-based algorithm can lie up to multiple orders of magnitude lower than when shortest path routing is used. Clearly, the type of the demand influences the factor of improvement: for non-uniform demands the reduction factor is higher because it is then more likely to have an unbalanced load in the considered equally meshed networks. Within a certain type of demand the reduction factor depends on the mean load on the network: the higher the overall load, the less room for improvement because lack of free capacity to reroute paths. Thus, apart from cases with very high overall network loads, the reduction of the max-PLR can be huge (multiple orders of magnitude).

The comparison of our PLR-based algorithm with load balancing shows why PLR-estimation can be useful: while load balancing offers huge improvement over shortest path routing, the PLR-based algorithm may offer additional reduction with a factor up to over two orders of magnitude (range of 1.5 to 900 for the analyzed cases). The reason is that load balancing ignores correlation between PLRs on outgoing links of the same node. This correlation stems from the sharing of the FDL

buffer. For low mean network loads (0.35-0.5) the factor lies between 1.5 and 2: the load on the links is so low that the buffer is not heavily used, resulting in negligible correlation. For mean network loads above 0.5, the factor ranges from 2 to 900: the higher link loads result in heavy use of the buffer and hence correlated loss rates, since the (limited) buffer space has to be shared by all traffic crossing the node. However, there is also a bound on the improvement: with very high link loads (e.g., 0.69 of L in the first studied network) there is not much room to reroute the paths, regardless of the algorithm used.

Table H.4: Comparison of max-PLR reached with PLR-based routing, shortest path routing and load balancing for different demands.

	max-PLR			*ratio of max-PLRs*	
max-PLR	*shortest path*	*load balancing*	*PLR-based*	*shortest/ load*	*load/ PLR-based*
2-uni 0,38	1,77E-04	1,93E-10	1,21E-10	9,17E+05	1,60
2-uni 0,45	3,36E-03	5,31E-05	3,24E-05	6,33E+01	1,64
2-uni 0,48	3,48E-03	1,12E-03	6,80E-04	3,10E+00	1,65
2-rl 0,36	1,39E-04	6,89E-10	3,53E-10	2,01E+05	1,95
2-rl 0,40	3,13E-03	6,21E-06	3,19E-06	5,05E+02	1,95
2-rl 0,44	3,46E-03	6,82E-04	4,84E-04	5,08E+00	1,41
1-S 0,54	1,27E-03	5,42E-06	5,83E-09	2,34E+02	929,67
1-S 0,71	2,39E-03	7,77E-04	2,38E-06	3,07E+00	326,47
1-S 0,75	2,54E-02	1,35E-02	3,60E-03	1,87E+00	3,76
1-L 0,53	6,11E-03	1,33E-07	3,50E-08	4,60E+04	3,80
1-L 0,61	1,20E-02	4,38E-04	1,46E-06	2,74E+01	300,27
1-L 0,69	1,92E-02	1,02E-04	9,65E-05	1,88E+02	1,06
1-uni 0,64	1,94E-03	9,05E-06	4,51E-08	2,14E+02	200,67
1-uni 0,72	5,76E-03	1,19E-03	3,34E-05	4,86E+00	35,56

Especially for the demand-types S and uni the PLR-based algorithm reaches max-PLR values more than one order of magnitude lower than the values obtained with the load balancing algorithm. An explanation for this can be found in the relatively high overall link load in these cases. As the PLR-based algorithm takes into account the influence of other heavily loaded output fibers, the routing is performed more accurate and lower max-PLR values are reached. For demand-type L the overall link load is also high, but the larger demands occurring here can not so easily be rerouted.

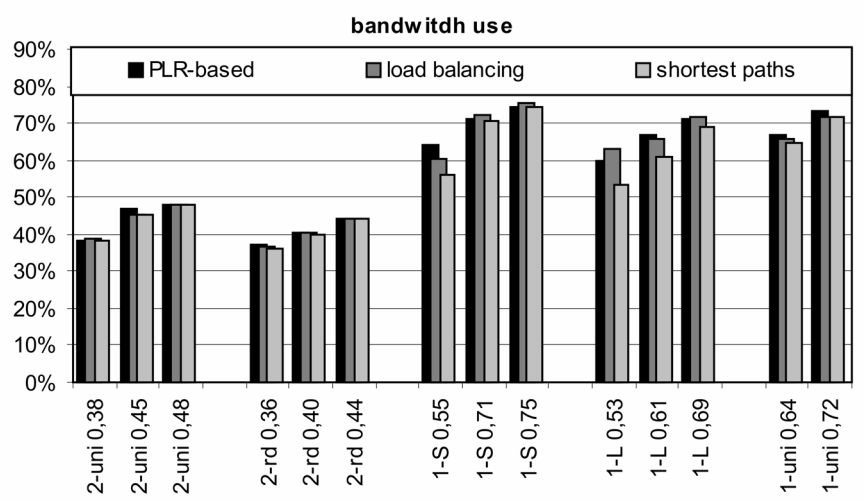


Figure H.16: Comparison of bandwidth-use (%) obtained with PLR-based routing, shortest path routing and load balancing for different demands.

The "penalty" for the more intelligent routing (in terms of PLR) is a small increase in bandwidth-use (a few percent) compared to shortest path routing. This can be seen in Figure H.16, where the bandwidth utilization is plotted for the same demands as above (Figure H.15). Here, bandwidth utilization is defined as the sum over all links of the bandwidth used on that link, divided by the sum of the total available bandwidth on each link (a bandwidth utilization of 100% means all links in the network are fully used). The graph in Figure H.16 shows that both load balancing and PLR-based routing algorithms lead to slightly higher bandwidth utilization than shortest path routing. Clearly, this is caused by rerouting some demands from their original shortest path: more links are used to fulfill the demand. Obviously, the increase is the most pronounced in the case with some large demands (L). Still, even here the difference in bandwidth-use does not exceed 7%. Whether load balancing uses either more or less bandwidth than PLR-based routing depends on the case at hand, since the order in which demands are rerouted from their original shortest paths differs.

We can conclude that depending on how strict the max-PLR restriction for the demands are, one can either opt for the very simple shortest path routing, i.e., when no restrictions are given for the max-PLR; or, when lower max-PLR values are the goal, one can choose the more complex load balancing and finally when one wants an even better routing (in terms of PLR) one should opt for the even more complex PLR-based routing.

H.6 Influence of introducing multiple priorities on PLR-based routing

All simulations performed and discussed up to now considered traffic of one single priority. In this section, we are going to explore how introducing multiple priorities affects the max-PLR values reached with our developed PLR-based routing algorithm.

We start in Section H.6.1 to compare the max-PLR values reached with monolithic traffic streams all of the same priority, and the values reached with the same amount of traffic but now dividing each source-destination stream into two parts of a different priority (of which 15, 30, or 45 percent highest priority traffic and correspondingly 85, 70, or 55 percent lowest priority traffic). Note that for this comparison two effects come into play: (i) the effect of splitting demands into smaller parts and (ii) the effect of using different priorities for these parts. To identify the influence of each of them, we separate them and discuss their impact individually in the following Subsections H.6.2 and H.6.3.

All assumptions concerning networks and node parameters in this section are the same as for the simulations with traffic of a single priority (see Section H.4). In addition, we consider one extra network to perform simulations on: the well-known NSFNET (see Figure H.17). This network consists of 14 nodes and 21 links and has an average node degree of 2.93, a minimum node degree of 2 and a maximum node degree equal to 4.

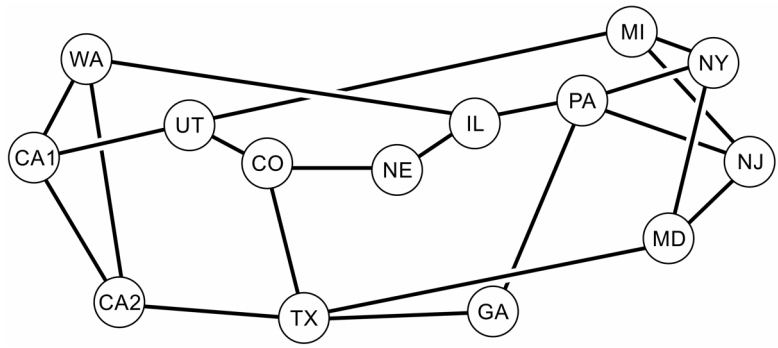


Figure H.17: The NSFNET.

Regarding the traffic assumptions there are, like with one priority, different (non-)uniform demands and the traffic is Poisson-distributed. The only difference with the single priority-case is that the traffic consists of two priorities with following 'high priority traffic/low priority traffic' ratios: 15/85 - 30/70 - 45/55. The goal was to obtain, for both priorities, max-PLRs as low as possible.

As indicated before (see Section H.3.2), our PLR-based algorithm can easily be extended from one to multiple priorities. Therefore, the algorithm used in this section is the WorstLink variant of the developed PLR-based algorithm. This algorithm is consecutively run two times: first for the highest priority traffic, then

for the lowest priority traffic. For the PLR calculation we always use the formula of Section H.2.2, which was actually developed for one priority traffic but can also be used with multiple priorities. In the first run of the algorithm we calculate the PLR for the highest priority traffic with the formula (as lower priority traffic does not influence higher priority traffic, see introduction). In the second run, for the lowest priority, we calculate the PLR for the total amount of highest and lowest priority traffic and substract the PLR obtained for the highest priority class traffic.

H.6.1 Splitting demands into multiple priorities.

Figure H.18 shows, for different demands on the three studied networks (on the X-axis), a comparison of the max-PLR values reached with the WorstLink variant of the PLR-based algorithm, when (i) considering traffic of a single priority (dash-dotted line with x markers in the figure, 100/0 highest/lowest priority ratio) and (ii) when considering different ratios of traffic of two priorities (ratio high priority traffic/low priority traffic respectively dotted with diamond markers for 15/85; full with square markers for 30/70; dashed with triangular markers for 45/55 in the figure).

As before, the type of demand is indicated on the X-axis: uni stands for uniform, rl for real non-uniform, rd for random non-uniform, S for random demands all of the same order, and L for random patterns where a few demands are of a larger order. The 1 and 2 refer to the 19-node and the 27-node network, nsf refers to the NSFNET; the last number in each label on the X-axis is the overall mean network load when considering shortest path routing (range [0,1]).

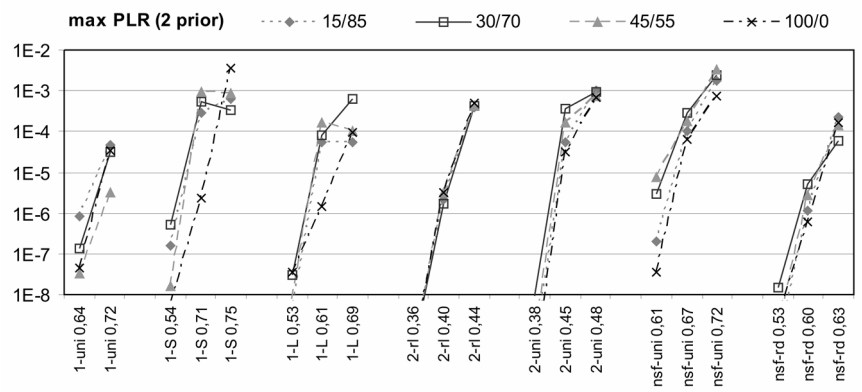


Figure H.18: Comparison of max-PLR values reached with the WorstLink variant of the PLR-based algorithm, when (i) considering traffic of a single priority (100/0 ratio highest/lowest priority ratio) and (ii) when considering different ratios of traffic of two priorities (15/85;30/70;45/55 ratio high priority traffic/low priority traffic). This for different demands on the three studied networks (on the X-axis).

Table H.5 shows the minimal max-PLR values reached for the different demands and the different highest/lowest priority ratios. It also shows (in the last three columns) the ratio between the max-PLR value reached with one priority traffic and the max-PLR value reached with two priorities of traffic (respective ratios: 15/85, 30/70, 45/55).

Table H.5: Comparison of max-PLR values reached with the WorstLink variant of the PLR-based algorithm, when (i) considering traffic of a single priority (100/0 ratio highest/lowest priority ratio) and (ii) when considering different ratios of traffic of two priorities (15/85;30/70;45/55 ratio high priority traffic/low priority traffic). This for different demands on the three studied networks (on the X-axis).

	max-PLR for two priorities (high/low)				*ratio of PLRs (with/no prio)*		
scenario	*15/85*	*30/70*	*45/55*	*100/0*	*15/100*	*30/100*	*45/100*
1-uni 0,64	8.28E–07	1.31E–07	3.26E–08	4.51E–08	18.36	2.90	0.72
1-uni 0,72	4.64E–05	3.04E–05	3.28E–06	3.34E–05	1.39	0.91	0.10
1-S 0,54	1.61E–07	5.09E–07	1.63E–08	5.83E–09	27.62	87.31	2.80
1-S 0,71	2.95E–04	5.54E–04	9.60E–04	2.38E–06	123.95	232.77	403.36
1-S 0,75	6.32E–04	3.31E–04	8.92E–04	3.60E–03	0.18	0.09	0.25
1-L 0,53	3.79E–09	2.96E–08	7.74E–09	3.50E–08	0.11	0.85	0.22
1-L 0,61	5.58E–05	8.09E–05	1.69E–04	1.46E–06	38.22	55.41	115.75
1-L 0,69	5.36E–05	6.45E–04	1.05E–04	9.65E–05	0.56	6.68	1.09
2-rl 0,36	3.66E–10	4.02E–10	3.37E–10	3.53E–10	1.04	1.14	0.95
2-rl 0,40	2.23E–06	1.74E–06	3.07E–06	3.19E–06	0.70	0.55	0.96
2-rl 0,44	3.96E–04	4.23E–04	4.36E–04	4.84E–04	0.82	0.87	0.90
2-uni 0,38	3.58E–10	9.55E–09	8.24E–10	1.21E–10	2.96	78.93	6.81
2-uni 0,45	5.52E–05	3.75E–04	1.62E–04	3.24E–05	1.70	11.57	5.00
2-uni 0,48	1.00E–03	9.39E–04	7.50E–04	6.80E–04	1.47	1.38	1.10
nsf-uni 0,61	1.94E–07	3.01E–06	7.72E–06	3.66E–08	5.30	82.24	210.93
nsf-uni 0,67	9.90E–05	2.97E–04	1.74E–04	6.59E–05	1.50	4.51	2.64
nsf-uni 0,72	1.80E–03	2.49E–03	3.30E–03	7.62E–04	2.36	3.27	4.33
nsf-rd 0,53	6.56E–10	1.53E–08	1.35E–09	2.69E–09	0.24	5.69	0.50
nsf-rd 0,60	1.15E–06	4.94E–06	2.81E–06	5.92E–07	1.94	8.34	4.75
nsf-rd 0,63	2.22E–04	5.89E–05	1.39E–04	1.71E–04	1.30	0.34	0.81
mean	3.84E–06	8.66E–06	4.64E–06	1.87E–06	2.05	4.64	2.48

When comparing the max-PLR values reached with traffic of one single priority (hundred percent of one priority) and the values reached with traffic of two priorities, we observe (see Table H.5 and Figure H.18) that in most cases (i.e., for most of the demands) the minimal max-PLR values obtained with traffic of two

priorities are higher than the values reached with traffic of one single priority. The minimal max-PLR values reached for traffic of two priorities given, are values for the lowest priority traffic, the PLR values for the highest priority traffic are much lower (<< 1E-10). So, the penalty of introducing priorities is a small increase of the minimal max-PLR value for the lowest priority traffic, while the PLR-values of the higher priority traffic can be kept very low.

This increase of the minimal max-PLR can be explained as follows. With two priorities of traffic, we first route the traffic with the highest priority as good as possible. After this, we route the lower priority traffic, but, when routing this traffic, we can not change the routes of the higher priority traffic any more, so we have less flexibility to reroute the lowest priority traffic, which results in a slightly higher minimal max-PLR value.

We however observe also some results where the opposite holds: the same amount of traffic divided into two priority classes returns better results (i.e., a lower max-PLR) than all traffic of the same priority. In most of these exceptional cases, the network load is relatively high. An explanation for this could be that we not only have to consider the effect of introducing priorities but we should also take into account the influence of the splitting of the demands into smaller parts, on the PLR-values. With high network loads we could benefit from this division of large demands into smaller parts. As all traffic of one demand is sent over the same path, dividing the traffic over two different paths could give better results because we then route smaller traffic parts. This issue is addressed in the following Subsection 6.2. Rarely (e.g., in case 1-L 0.53) the exception of reaching lower PLRs when splitting traffic into two priorities also occurs for a low network load. This can be put down to inaccurate PLR-values in this range.

Averaging the values over the different demands/networks (see last row of Table H.5), we observe that the averaged max-PLR value obtained with traffic of two priorities (for the low priority traffic), is, for all high/low priority ratios, at most a factor 4.7 higher than the averaged max-PLR value reached with pure one priority traffic. Thus, for two priorities traffic there is only a small increase of the max-PLR value for the lowest priority traffic: while attaining negligible loss rates for high priority traffic, loss rate for low priority streams slightly increases but remains within the same order of magnitude.

In the following we split up the two effects that occur when introducing priorities, therefore we first (Subsection 6.2) take a look at the individual impact of splitting demands into smaller parts on the max-PLR values and next (Subsection 6.3) we study the effect of using different priorities for these smaller parts.

H.6.2 The effect of splitting demands into smaller parts

To study the effect of splitting (large) demands into smaller parts, we split up the monolithic traffic streams of one priority into two smaller parts/demands of the same priority. We start with routing the demands with 15/30/45 percent of the traffic, followed by the larger demands of 85/70/55 percent of traffic of the same priority. As a result we consider now the same traffic demands as in the two priorities case, except that demands now all have the same priority. The difference with the case with two priorities is that here the smaller parts of the demands, which are routed

first, can, in contrast to the highest priorities after the first run of the algorithm in the two priorities case, still be rerouted in the following iterations of the algorithm, even when bigger parts have already been rerouted.

Figure H.19 shows, for the same demands as above, the max-PLR values reached with the WorstLink version of the algorithm for demands of pure one priority traffic (dash-dotted line with x markers) and for the same demands split into two parts of the same priority: 15/85 (dotted line with diamond markers); 30/70 (full line with square markers); 45/55 (dashed line with triangular markers).

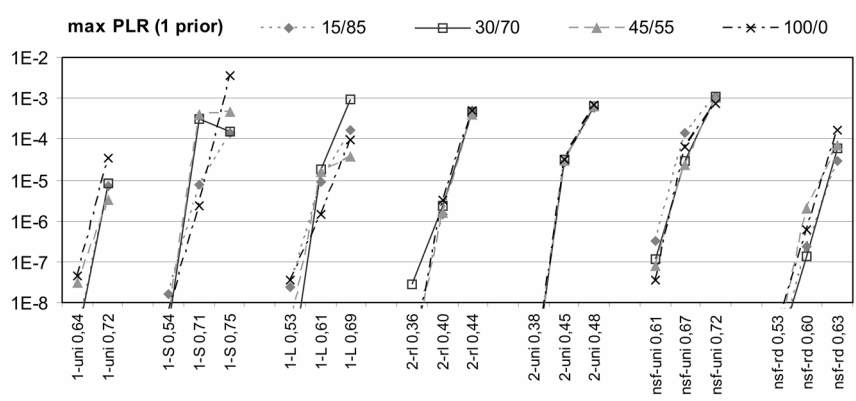


Figure H.19: Comparison of max-PLR values reached with the WorstLink variant of the PLR-based algorithm for traffic of one priority (i) for demands of 100% (100/0) and (ii) for the same demands split into two parts: 15/85; 30/70; 45/55.

Comparing now (see Figure H.19) the max-PLR values reached when splitting the demands into smaller parts (15/85, 30/70, 45/55) with the values reached with demands which are not split, we see that indeed in many cases the minimal max-PLR values reached with the demands split into two smaller parts, are lower than when demands are not split. When averaging the values over the different demands, we observe a small decrease of the mean value reached when demands are split, showing a (limited) advantage of dealing with finer granularity streams.

H.6.3 Routing smaller parts: the effect of introducing priorities.

Figure H.20 to 22 show, again for the same demands as above, for the different traffic demand ratios (15/85; 30/70; 45/55) the minimal max-PLR values reached when the two parts the demands are split into are (i) of the same priority, (ii) of a different (high/low) priority. Table F.6 also shows these values and a ratio of (ii) to (i).

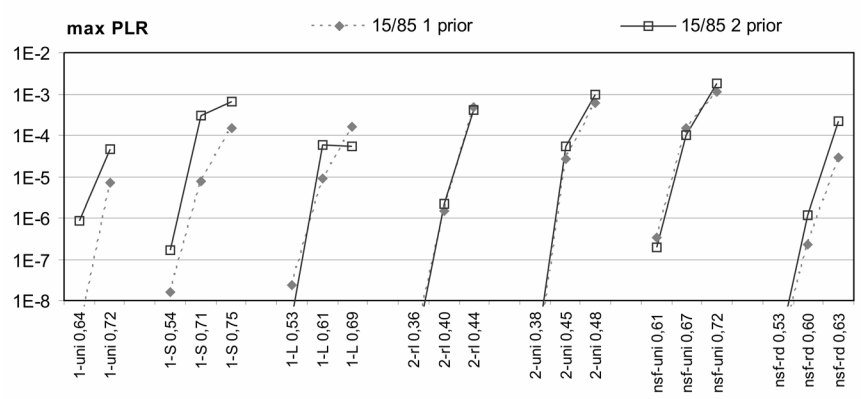


Figure H.20: Comparison of the max-PLR values reached for the traffic demand ratio 15/85 when the two parts the demands are split into are (i) of the same priority, (ii) of a different (high/low) priority.

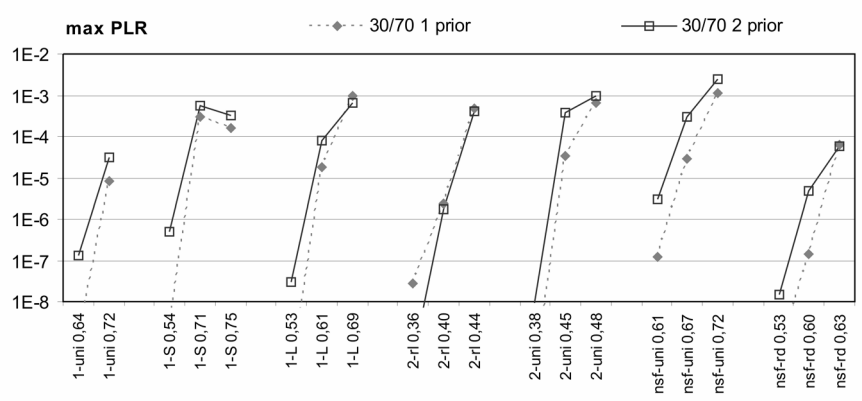


Figure H.21: Comparison of the max-PLR values reached for the traffic demand ratio 30/70 when the two parts the demands are split into are (i) of the same priority, (ii) of a different (high/low) priority.

To study the effect of introducing priorities in traffic split up in smaller parts, we compare the max-PLR values reached with the demands of one priority traffic split into two parts (15/85, 30/70, 45/55) with the max-PLR values obtained with the same demands, but where the two parts the demands are split into are of a different priority.

We see (Figure H.20 to 22 and Table F.6) that in most cases the minimal max-PLR values reached for the two priorities case (which are again the values for the lowest priority traffic as the values for the highest priority traffic are much lower) are slightly higher than the ones for split up one priority traffic. In only a few cases the minimal max-PLR value reached with traffic with two priorities is lower than with one priority traffic.

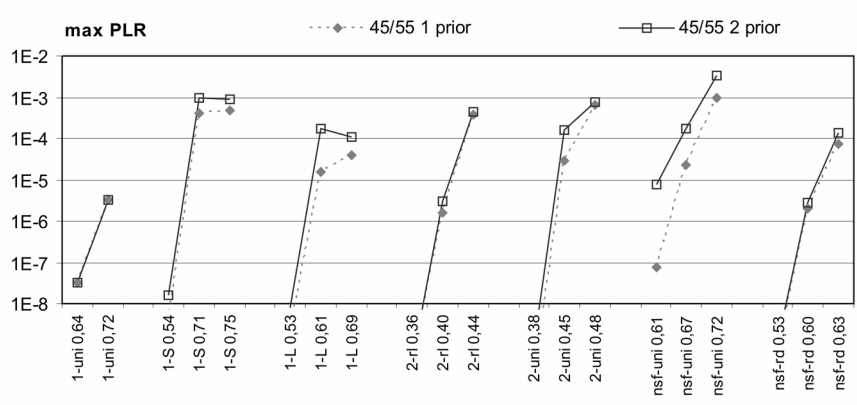


Figure H.22: Comparison of the max-PLR values reached for the traffic demand ratio 45/55 when the two parts the demands are split into are (i) of the same priority, (ii) of a different (high/low) priority.

When we average the values over the different demands and networks (for these values see the last row of Table F.6), we see that the averaged value of traffic of two priorities, is at most a factor 5.7 higher than the averaged max-PLR value reached with traffic of one priority.

An explanation for these higher values with two priorities of traffic is thus the smaller flexibility in rerouting the lowest priority traffic: once the highest priority traffic has been routed, it is left untouched when routing the lower priority.

We conclude that introducing multiple priorities increases the minimal max-PLR value for the lowest priority traffic while for the highest priority the max-PLR value can be kept very low. As the increase of the minimal max-PLR value for the lowest priority traffic is not significant, the cost of introducing priorities is small.

H.7 Conclusion

We explained how results obtained for a single node could be used to assess end-to-end PLR for demands routed in a network of interconnected OPRs. We developed routing algorithms aiming at minimizing the PLR using this PLR-estimation technique. Two alternative algorithms were compared and the WorstLink variant proved to lead to the best results.

Case studies on pan-European networks illustrated that the PLR-based routing outperforms shortest path routing by lowering the max-PLR occurring in the network with multiple orders of magnitude. Compared to load balancing, which does not need PLR-estimation, the PLR-based algorithm can reach PLRs up to two orders of magnitude better. The price paid for the reduction in PLR is a higher overall bandwidth-use. Still, the amount of extra bandwidth needed compared to shortest path routing is quite limited (only a few percent).

.

Table H.6: Comparison of the minimal max-PLR values reached for the different traffic demand ratios (15/85; 30/70; 45/55) when the two parts the demands are split into are (i) of the same priority, (ii) of a different (high/low) priority.

	15/85			*30/70*			*45/55*			*100/0*
scenario	*1 pri*	*2 pri*	*1 pri / 2 pri*	*1 pri*	*2 pri*	*1 pri / 2 pri*	*1 pri*	*2 pri*	*1 pri / 2 pri*	
1-uni 0.64	1.61E-09	8.28E-07	5.14E+02	1.42E-09	1.31E-07	9.23E+01	3.12E-08	3.26E-08	1.04E+00	4.51E-08
1-uni 0.72	6.82E-06	4.64E-05	6.80E+00	8.03E-06	3.04E-05	3.79E+00	3.28E-06	3.28E-06	1.00E+00	3.34E-05
1-S 0.54	1.65E-08	1.61E-07	9.76E+00	2.68E-09	5.09E-07	1.90E+02	4.03E-09	1.63E-08	4.04E+00	5.83E-09
1-S 0.71	7.78E-06	2.95E-04	3.79E+01	2.99E-04	5.54E-04	1.85E+00	4.04E-04	9.60E-04	2.38E+00	2.38E-06
1-S 0.75	1.45E-04	6.32E-04	4.36E+00	1.58E-04	3.31E-04	2.09E+00	4.77E-04	8.92E-04	1.87E+00	3.60E-03
1-L 0.53	2.31E-08	3.79E-09	1.64E-01	1.46E-10	2.96E-08	2.03E+02	2.12E-09	7.74E-09	3.65E+00	3.50E-08
1-L 0.61	8.70E-06	5.58E-05	6.41E+00	1.76E-05	8.09E-05	4.60E+00	1.50E-05	1.69E-04	1.13E+01	1.46E-06
1-L 0.69	1.60E-04	5.36E-05	3.35E-01	9.42E-04	6.45E-04	6.85E-01	3.82E-05	1.05E-04	2.75E+00	9.65E-05
2-rl 0.36	7.40E-10	3.66E-10	4.95E-01	2.85E-08	4.02E-10	1.41E-02	3.40E-10	3.37E-10	9.91E-01	3.53E-10
2-rl 0.40	1.49E-06	2.23E-06	1.50E+00	2.32E-06	1.74E-06	7.50E-01	1.59E-06	3.07E-06	1.93E+00	3.19E-06
2-rl 0.44	4.95E-04	3.96E-04	8.00E-01	4.78E-04	4.23E-04	8.85E-01	3.81E-04	4.36E-04	1.14E+00	4.84E-04
2-uni 0.38	1.21E-10	3.58E-10	2.96E+00	1.21E-10	9.55E-09	7.89E+01	1.21E-10	8.24E-10	6.81E+00	1.21E-10
2-uni 0.45	2.66E-05	5.52E-05	2.08E+00	3.24E-05	3.75E-04	1.16E+01	2.86E-05	1.62E-04	5.66E+00	3.24E-05
2-uni 0.48	5.93E-04	1.00E-03	1.69E+00	6.35E-04	9.39E-04	1.48E+00	6.34E-04	7.50E-04	1.18E+00	6.80E-04
nsf-uni 0.61	3.25E-07	1.94E-07	5.97E-01	1.19E-07	3.01E-06	2.53E+01	7.82E-08	7.72E-06	9.87E+01	3.66E-08
nsf-uni 0.67	1.43E-04	9.90E-05	6.92E-01	2.80E-05	2.97E-04	1.06E+01	2.34E-05	1.74E-04	7.44E+00	6.59E-05
nsf-uni 0.72	1.11E-03	1.80E-03	1.62E+00	1.12E-03	2.49E-03	2.22E+00	1.00E-03	3.30E-03	3.30E+00	7.62E-04
nsf-rd 0.53	4.39E-10	6.56E-10	1.49E+00	5.50E-10	1.53E-08	2.78E+01	8.16E-10	1.35E-09	1.65E+00	2.69E-09
nsf-rd 0.60	2.27E-07	1.15E-06	5.07E+00	1.40E-07	4.94E-06	3.53E+01	2.00E-06	2.81E-06	1.41E+00	5.92E-07
nsf-rd 0.63	2.80E-05	2.22E-04	7.93E+00	6.07E-05	5.89E-05	9.70E-01	7.06E-05	1.39E-04	1.97E+00	1.71E-04
mean	1.41E-06	3.84E-06	2.72E+00	1.53E-06	8.66E-06	5.67E+00	1.58E-06	4.64E-06	2.94E+00	1.57E-06

It was also shown that the discussed PLR-based routing algorithm can be easily extended to multiple priorities. Introducing multiple priorities enables to keep loss rates negligible for high priority traffic, while it may lead to an increase of the obtained minimal max-PLR value for low priority traffic. However, the increase proved to be limited: the cost of introducing multiple priorities is small.

References

[1] B. Mukherjee, "WDM optical communication networks: Progress and challenges", IEEE J. Select. Areas Commun., vol. 18, no. 10, Oct. 2000, pp. 1810–1824.

[2] S. Yao, B. Mukherjee, S. Dixit, "Advances in photonic packet switching: an overview", IEEE Commun. Mag, vol. 38, no. 1, Jan. 2000, pp. 84–94.

[3] D.K. Hunter, I. Andonovic, "Approaches to optical internet packet switching", IEEE Commun. Mag., vol. 38, no. 9, Sept. 2000, pp. 116–120.

[4] A. Hill and F. Neri, guest ed., "Optical switching networks: from circuits to packets", IEEE Commun. Mag., vol. 39, no. 3, March 2001, pp. 107-148.

[5] C. Qiao and M. Yoo, "Optical Burst Switching (OBS) — a new paradigm for an optical internet", J. High Speed Networks, vol. 8, no. 1, Jan. 1999, pp. 69–84.

[6] C. Qiao, "Labeled optical burst switching for IP-over-WDM integration", IEEE Commun. Mag., vol. 38, no. 9, Sept. 2000, pp. 104–114.

[7] D.K. Hunter et al., "WASPNET – a wavelength switched packet network", IEEE Commun. Mag., vol. 37, no. 3, March 1999, pp. 120–29.

[8] Soren L. Danielsen, Carsten Joergensen, Benny Mikkelsen, Kristian E. Stubkjaer, "Analysis of a WDM Packet Switch with Improved Performance Under Bursty Traffic Conditions Due to Tuneable Wavelength Converters", IEEE/OSA Journal of Lightwave Technology, vol. 16, no. 5, May 1998, pp. 729–735.

[9] S. Yao, B. Mukherjee, S. J. Ben Yoo and S. Dixit, "All-optical packet-switched networks: a study of contention-resolution schemes in an irregular mesh network with variable-sized packets", Proc. SPIE vol. 4233, (OPTICOMM 2000, Plano, TX, USA, Oct. 2000), pp. 235–246.

[10] H. Zang, J.P. Jue, and B. Mukherjee, "Capacity allocation and contention resolution in a photonic slot routing all-optical WDM mesh network", IEEE/OSA J. Lightwave Tech., vol. 18, no. 12, Dec. 2000, pp. 1728–41.

[11] D.K. Hunter, M.C. Chia and I. Andonovic, "Buffering in optical packet switches", IEEE/OSA J. Lightwave Tech., vol. 16, no. 12, Dec. 1998, pp. 2081–94.

[12] L. Dittman et al, "The IST project DAVID: a viable approach towards optical packet switching", to be published in IEEE J. Select. Areas Commun. *http://david.com.dtu.dk*

[13] Chiaroni, et al., "First demonstration of an asynchronous optical packet switching matrix prototype for MultiTerabitclass routers/ switches", in Proc. 27th European Conf. on Optical Communication, (Amsterdam, The Netherlands, September 30 – October 4, 2001), vol. 6, pp. 60–61.

[14] C. Develder, M. Pickavet, P. Demeester, “Assessment of packet loss for an optical router with recirculating buffer”, in Proc. ONDM 2002 6th Working Conf. on Optical Network Design and Modeling (Turin, Italy, February 4–6, 2002), pp. 247-261.
[15] C. Develder, M. Pickavet, P. Demeester, “Choosing an appropriate buffer strategy for an optical packet switch with a feed-back FDL buffer”, in Proc. 28th European Conf. on Optical Communication, (Copenhague, Denmark, September 8–12, 2002), vol. 3, pp. 8.5.4 (2 pages).
[16] C. Develder, et al., “Node architectures for optical packet and burst switching”, in Tech. Digest. Int. Topical Meeting on Photonics in Switching (PS2002), (invited) paper PS.WeA1, (Cheju Island, Korea, July 21–25, 2002), pp. 104–106.
[17] B. Fortz, M. Thorup, “Internet Traffic Engineering by Optimizing OSPF Weights”, in Proc. IEEE Infocom 2000, (Tel Aviv, Israel, March 26-30, 2000), pp. 519-528.
[18] J.T. Moy, “OSPF: Anatomy of an Internet Routing Protocol”, Addison-Wesley, February 1998.
[19] B. Davie, Y. Rekhter, “MPLS: Technology and applications”, Morgan Kaufman Publishers, January 2000.
[20] W. Willinger, et al, “Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level”, IEEE/ACM Trans. on Networking, vol. 5, no. 1, Jan. 1997, pp. 71–86.
[21] M. Pickavet, C. Develder, E. Baert, P. Demeester, "A.I. Techniques for planning telecommunication networks", Proc. International Conference on Artificial Intelligence IC-AI'02, (Las Vegas, NV, June 24-27, 2002), vol.2, pp. 608–614.
[22] S. De Maesschalck, et al., “Pan-European optical transport networks: an availability-based comparison”, Photonic Network Communications, vol. 5, no. 3, May 2003, pp. 203–225.
http://ibcn.atlantis.UGent.be/projects/COST266_IST_lion/NRS/index.html

Appendix I

Influence of GMPLS recovery mechanisms on TCP performance

C. Develder, D. Colle, S. De Maesschalck, M. Pickavet, P. Demeester

Photonic Network Communications (PNET), vol. 4, no. 3/4, Jul.-Dec. 2002, pp. 321–343.

Abstract: Optical networks based on Wavelength Division Multiplexing (WDM) techniques are very likely to be omnipresent in future telecommunication networks. Those networks are deployed in order to face the steady growth of traffic, which is for a large part Internet related. In the resulting IP-over-WDM scenario, TCP/IP constitutes an important fraction of the traffic transported over these networks. As IP networks are becoming increasingly mission-critical, it is of the utmost importance that these networks (and hence the supporting transport networks) be able to recover quickly from failures such as cable breaks or equipment outages. To that end, several IP-over-WDM network scenarios and corresponding protection and restoration strategies have been devised. It is clear that some trade-offs will have to be made in order to choose an appropriate strategy. In this paper, we investigate the effects of such recovery actions on the behavior of TCP, being the ubiquitous protocol used by today's network users. We examine the influence of different parameters such as the speed of recovery actions, changing length of the routes followed by the client data (TCP flows), changes in available bandwidth, etc. Thereby, we focus on what the TCP end-users care about, i.e. the number of bytes transported end-to-end within a certain time interval.

Keywords: (G-)MPLS, IP-over-WDM, protection, recovery, TCP

I.1 Introduction

Telecommunication networks in recent years have faced an explosive traffic growth, mainly due to the popularity of the Internet [1]. For a couple of years already, the data traffic volume dominates (classical) voice traffic, and recent forecasts do not seem to predict a rapid slowdown of this greediness for bandwidth [2]. Communication networks will be more and more optimized for the dominant IP traffic, as TCP/IP is functioning as the convergence layer for practically all forms of end-user communication in today's data communication networks. It is foreseeable that it will continue to do this in tomorrow's multi-service networks, where IP-based applications such as voice, video and other multimedia applications will generate the necessary revenues (foreseen to outpace classical voice revenues). While some of these new applications tend to use the unreliable User Datagram Protocol (UDP), the reliable Transmission Control Protocol (TCP) today still is responsible for a major portion of the IP traffic [3,4] and is used extensively by many of the so-called peer-to-peer (P2P) applications [5] that arose in the wake of the popular Napster, and are used for various purposes, ranging from file and knowledge sharing to distributed computing.

To cope with the large traffic volumes that such a multi-service network necessitates, wavelength-division multiplexing (WDM) technology has been devised. Fiber exhaust is currently solved by multiplying the capacity of a fiber by means of point-to-point WDM systems; a multiplexing technique that has proven to be very cost-efficient due to the economy of scale [6]. Current optical component technologies enable to introduce networking functionality in the WDM layer (by means of optical add-drop multiplexers and cross-connects), that laid the foundations for an Optical Transport Network (OTN). Automating (the configuration of) these networking functions will result in an Automatic Switched (Optical) Transport Network, currently under development in e.g. the ITU [7].

Due to the growing importance of IP traffic and the opportunities offered by WDM technology, many research activities are dedicated to bringing the two closer together, under the flag of IP-over-(D)WDM [8]. A trend observed today is to eliminate or reduce intermediate layers between IP and WDM. Generalized Multi-Protocol Label Switching (GMPLS) [9–11] promises to offer the necessary control plane “glue”' to join them quasi directly.

As IP networks grow more and more mission critical, more stringent requirements are imposed on them: the support of service differentiation, introduction of quality of service (QoS) [12], but also the ability to survive network failures [13], all are important research topics. The survivability of a network is usually guaranteed by a set of restoration and/or protection schemes. A question that arises when comparing those schemes is what the effect of such protection actions is on the dominant client layer: TCP (according to recent measurements reported by Leinen [4], it accounts for 85% of the packets, and more than 95% of the bytes transmitted on some transatlantic links). This is the question we will focus on in this paper. Evidently, the extent to which TCP withstands the actions taken by a protection mechanism is only one of the touchstones to judge it on. Other criteria include the amount of control traffic and/or state information it demands, and

bandwidth requirements [14,15]. This however lies out of the scope of this paper: the only facet of the protection mechanisms we study here is their effect on TCP behavior.

The remainder of this paper is structured as follows: in Section I.2, we will give an overview of GMPLS protection mechanisms. In the following Section I.3, we will discuss the reactive nature of TCP. The effects of the GMPLS recovery actions on TCP that we will study in this paper, and the adopted approach, will be outlined in Section I.4. In the subsequent sections, we will present the simulations used to seek an answer to the questions raised in our discussion of TCP: in Section I.5, we will investigate the effect of the speed of protection switching, while in Section I.6 we will focus on the impact of the changing path length. The switch-back operation to the original path after the failure has been repaired will be dealt with in Section I.7. The joint impact of speed and changing RTTs will be discussed for the presented GMPLS recovery schemes is the subject of the case study in Section I.8. In Section I.9, we will summarise the conclusions of the paper.

I.2 GMPLS protection mechanisms

Protection in MPLS is based on pre-established Label Switched Paths (LSPs), spanning a single link or node from an associated working LSP, or the whole working LSP from ingress to egress. The former case is generally denoted as Local Protection, and the latter as Path Protection [16,17]. These protection mechanisms are illustrated in Figure I.1. Path Protection always (e.g. during failure 1 and failure 2) switches the traffic in the ingress (node A in the figure) to the single backup LSP. Local Protection needs a backup LSP per protected link or node: in case of failure 1, traffic will be rerouted along the middle backup LSP —indicated by the dotted arrows— that is pre-established between the end-points of the affected link B–C. In a similar way, the backup LSP indicated by the dash-dotted arrows is used in response to failure 2. The Label Switch Router (LSR) where the backup LSP is originating and the switch-over operation from primary to backup path is performed, is called the Path Switch LSR (PSL), whereas the Path Merge LSR (PML) is the one where backup and primary LSPs are merged into a single outgoing LSP.

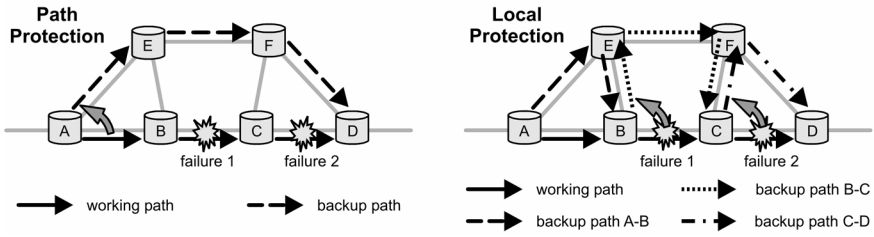


Figure I.1: Illustration of path and local protection, under two different failure scenarios.

From the figure, some weaknesses of each of the approaches are immediately apparent. Local Protection typically needs to foresee a backup LSP for each link (or

node) of the primary LSP (in Figure I.1, there is a backup LSP for each link) —yet, some workarounds exist where label stacking is available [18]. Path protection on the other hand, while requiring only a single backup LSP per primary LSP, necessitates additional signaling functionality (the PSL needs to be notified of the upstream failure, in order to decide when to switch to the backup LSP) which takes more time and thus results in more data loss.

A third protection scheme, illustrated in Figure I.2, was suggested by Haskin and Krishnan [19] and combines the "best characteristics" of both schemes: Local Loop-back. The key idea of this approach is to have only a single backup LSP per primary LSP, while permitting local protection switches (i.e. at the node detecting the failure, without requiring a signal to be sent to a distant PSL). The figure shows that the loop-back protection switch is performed in different LSRs (although a single backup LSP is required), under distinct failure conditions. The backup LSP consists of two parts: first from the penultimate node back to the source node of the primary path (using the same links in the reverse direction), and then via a disjoint path to the destination node. As shown in the figure, at what LSR exactly the loop-back protection switch is performed, depends on the failure conditions.

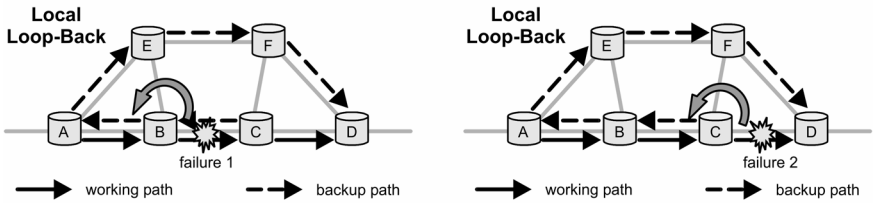


Figure I.2: The local loop-back technique combines the advantage of Path Protection (single backup LSP) and Local Protection (protection switch performed locally in an LSR adjacent to the failure).

These protection mechanisms were originally conceived for electrical MPLS, which is a packet-switch-capable (PSC) technology. The WDM layer however is an optical, lambda-switch-capable (LSC) layer in GMPLS. When porting the mechanisms to a circuit-based, non-merge-capable layer, such as the LSC layer (further denoted as MPλS), two main issues arise [15]. The first is that merging primary and backup paths at the PML may not be possible, as illustrated in Figure I.3. At all times, only one of the optical signals may be forwarded along the outgoing interface —even when the available bandwidth would suffice, when looking at the bandwidth effectively used by IP. Thus, when along both primary and backup incoming interfaces a signal carrying data would come in, at least one of them will be discarded.

A second and related issue, which has its implications on the dimensioning of the network [14], is that in MPλS a "label" corresponds with a wavelength, and therefore implies the occupation of a "circuit". Where PSC technologies allow statistical multiplexing of several (backup) LSPs over the same link, MPλS and other circuit-switched approaches, such as TDM or fiber-switching, do not.

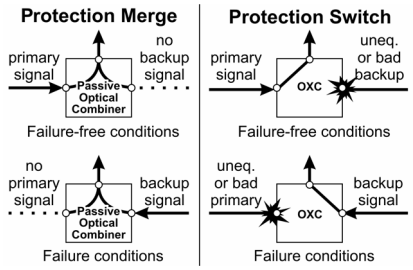


Figure I.3: A protection merge (left) can be realized by a passive optical combiner, if and only if backup and primary signals never are received simultaneously. If this condition cannot be met, a protection switch (right) is needed instead of a protection merge [13].

I.3 The reactive nature of TCP

The Transmission Control Protocol (TCP) is a connection-oriented data communication protocol, which is reliable in the sense that the sender keeps trying to send a data segment until the receiver acknowledges its proper receipt and the acknowledgement packet (ACK) does not get lost in the network. The amount of data the source may send out before it stops and waits for ACKs to come in from the receiver, is limited by the minimum of the congestion window *cwnd* and the receiver's advertised window *rwnd*.

While TCP is still being studied and continuously under development (see for example, Floyd [20] for a recent overview), the TCP basics have been described a relatively long time ago in RFC 793 [21]. The basis of TCP congestion control can be summarized in four components [20]: (i) the additive increase multiplicative decrease (AIMD), halving the congestion window in response to a packet drop, and if not, increasing it with one segment per round trip time (RTT); (ii) the use of a retransmit timer to start resending packets if they are not acknowledged within a certain period of time, denoted as the Retransmission TimeOut interval (RTO); (iii) the slow-start mechanism for initial probing of available bandwidth, where *cwnd* is increased with one Sender Maximum Segment Size (SMSS) per received ACK until it reaches the slow start threshold (*sstresh*); and (iv) the principle of ACK clocking, sending packets in response to the reception of ACKs.

When considering GMPLS, and in particular protection switching, two effects have an important impact on TCP behavior. The first is a sudden change in RTT experienced by the TCP flows. Indeed, the time needed to deliver a packet from source to destination (and vice versa for the ACKs) will change suddenly when the packets are sent along another path. The time to elapse before TCP triggers the retransmission of a packet, i.e. the RTO, is based on an estimate of this RTT that is called the Smoothed Round-Trip Time (SRTT), and is obtained by low-pass filtering the measured RTTs, which also takes into account the variance on the RTT [22]. A (protection) switch may cause the retransmission timer to expire, resulting in the unnecessary retransmission of some data segments (when packets are not lost, but simply are underway for a longer time). In order to respond reasonably fast to

changing network situations, the low-pass filtering of the average and deviation of the measured RTT should not react too slow.

A second effect is that a burst of consecutive packets may be lost when switching a flow to another path. This will be the case when considering a protection switch in response to a link failure: packets in transit on the failing link will be lost, and so will the subsequent ones, until the failure has been detected and appropriate action (in casu the protection switch) has been taken. If we consider, for example, the NewReno version of TCP [23], it will fall back to the Fast Recovery/Fast Retransmit algorithm. This procedure is triggered by the reception of three duplicate ACKs, upon which *sstresh* is adjusted to half the flight size (i.e. the amount of unacknowledged data sent out by the TCP source) and the congestion window is reset to *sstresh* plus three times the SMSS (to account for the three segments that have left the network and caused the duplicate ACKs). The first lost segment is then retransmitted. To keep track of the recovery process, the highest unacknowledged sequence number is stored in the variable *recover*. The Fast Retransmit/Fast Recovery process terminates when this sequence number has been acknowledged. Until then, the source classifies the receipt of an acknowledgement in three categories [23]: (i) a Duplicate ACK (with the same sequence number as a previously received one), (ii) a Partial ACK (with a new sequence number, but not acknowledging all data sent out before the Fast Retransmit/Fast Recovery procedure was triggered), or (iii) a Complete ACK (acknowledging all data, which had been sent out at the time the Fast Recovery/Fast Retransmit procedure was triggered, i.e. with sequence number *recover*).

In case a large burst of consecutive packets is lost —as may happen in case of a link failure— the Fast Retransmit/Fast Recovery rules cause a performance lack. Indeed, we can interpret these rules as follows. The number of duplicate acknowledgements —denoted as *ndup*— received one RTT after initiating the Fast Recovery/Fast Retransmit procedure is calculated as the flight size minus the size of the lost burst. This *ndup* is minimally three and maximally the flight size (at the moment of the initiation of the procedure, or thus twice the new value of *sstresh*) minus once the SMSS. This number *ndup* times SMSS gives the amount by which the congestion window size is incremented above *sstresh* during the considered RTT. However, only the part of the congestion window above the flight size allows sending new data (thus with a sequence number larger than the value stored in the *recover* variable). The amount of new data sent during this first RTT of the Fast Retransmit/Fast Recovery period can be estimated using Eq. (1):

$$\begin{aligned} new &= \max(0, cwnd - (lastsent - lastack)) \\ &= \max(0, (sstresh + ndup) - flightsize) \\ &= \max(0, (sstresh + flightsize - burstsize) - flightsize) \\ &= \max(0, (sstresh - burstsize)) \end{aligned} \tag{1}$$

At the end of this RTT, a partial acknowledgement should arrive as result of the retransmitted segment. The effect of the partial acknowledgement is that the flight size is decremented by the amount of acknowledged data (equaling the size of the acknowledged data segment); this is SMSS bytes more than the decrement of

congestion window size (which is typically zero). In other words, the receipt of a partial acknowledgement causes the flight size to decrement faster than the congestion window size. Therefore, after the receipt of one or more partial ACKs, the amount of new data that can be sent during the following RTT will evaluate to a strictly positive value. Subsequently, it keeps growing by one SMSS per retransmitted data segment (i.e. per RTT). Indeed, while the second lost segment is retransmitted, the source will receive a duplicate acknowledgement for each data segment potentially sent out during the first RTT. Each of those duplicate ACKs will result in the increment of the congestion window with one SMSS, allowing the same amount of new data to be sent out, in addition to the SMSS gain of the previous partial acknowledgement: the number of newly sent packets increases with one per RTT. This scenario per RTT will keep repeating until the last lost segment has been retransmitted. The amount of new data transmitted each RTT during the Fast Retransmit/Fast Recovery period is sketched in Figure I.4.

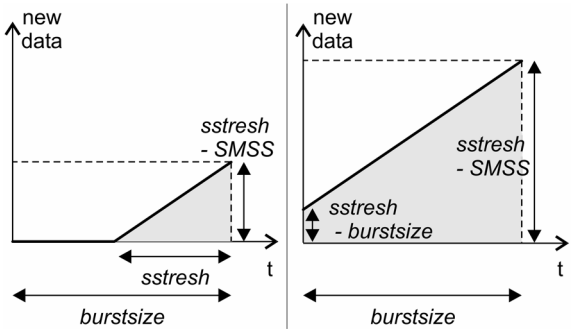


Figure I.4: Evolution of new data sent during successive RTTs in the Fast Retransmit/Fast Recovery phase of TCP NewReno. The left part illustrates the case where *burstsize* > *sstresh*, whereas the right part illustrates *burstsize* < *sstresh*. Note that the scales of the axes are not the same for both figures; the sloping part of the graphs increases with one SMSS per RTT.

In conclusion, the fast recovery/fast retransmit procedure would last for the number of lost data segments (*burstsize*) times the RTT. As long as the size of the unacknowledged burst remains higher than *sstresh* (minus SMSS), no progress will be made. The overall progress during this procedure is estimated as follows:

$$progress = \frac{\max(0, sstresh - burstsize) + (sstresh - SMSS)}{2} \cdot \frac{\min(sstresh, burstsize)}{SMSS} \quad (2)$$

This may become significantly less efficient than e.g. returning to slow start, when the size of the burst of lost data becomes too large. The drawback of slow start is that it needs to retransmit everything outstanding at the time the source detects a loss (i.e. twice the *sstresh*). Therefore, the maximal burst size (*mbz*) for which the Fast Recovery/Fast Retransmit procedure would effectively turn out to be the fastest is given by the inequality Eq. (3). The right hand side gives the amount of data sent during mbz round-trip times, having sequence numbers starting with the first packet

of the burst. The left hand side gives the number of packets that would have been transmitted using the Fast Retransmit/Fast Recovery algorithm of NewReno.

$$progress + 2 \cdot sstresh \geq (1 + 2 + \ldots + 2^{mbz}) \cdot SMSS = (2^{mbz+1} - 1) \cdot SMSS \qquad (3)$$

The conclusion is that, when developing a fast protection scheme, one should try to minimize the amount of lost packets: when this is not the case, one would intuitively expect from the above that it is better to allow the RTO timer to expire, bringing the TCP connection back into the slow start mode.

The discussion held so far focussed on a single TCP flow. However, a link will usually carry a multiplex of many TCP flows, and therefore the number of losses inflicted on a single TCP flow (i.e. the *burstsize*) and/or the number of affected flows may be relatively low. What exactly the effects of switching are on such an aggregate of flows, hence is not that straightforward to predict and far more difficult to capture in formulae. This is why we resorted to simulations, as described in the following section.

I.4 The effects of recovery actions on TCP

From our discourse on the reactive nature of TCP, we have learnt that for TCP basically two effects take place when considering network failures and GMPLS recovery actions: a burst of packet losses, and the changing RTT because of the switch to a (usually longer) backup path. Now, how well are recovery actions, and the aforementioned effects they cause, digested by the principal client layer protocol TCP? In Section I.4.1 we will clarify what exactly we will focus on. For our study, we have resorted to simulations, whose set-up will be discussed in Section I.4.2. The criterion used to quantify the behavior of TCP is the so-called goodput, which is briefly explained in Section 0.

I.4.1 Topics

We will address four topics, being (i) the effect of protection speed, (ii) the effect of changing RTTs at a switch-over, (iii) the effects of switch-back operations for both optical and electrical GMPLS layers, and (iv) how the previously discussed effects translate to the major GMPLS protection strategies.

The first issue we will investigate is what the influence is of the speed of protection switching on TCP flows. Indeed, using a (G)MPLS protection mechanism, the network can respond quite fast to network failures. The question arises how advantageous this is from a TCP point of view. Clearly, the number of losses inflicted on the TCP flows will be directly related to how fast the connectivity is restored after a failure. Also, if the time needed to perform the switch-over to an alternative path is relatively large compared to the RTT for the affected flows, then TCP's retransmission timer may expire, causing it to fall back to slow start. As indicated before, this implies that the retransmission of the lost packets may be realised at a quicker pace than with the Fast Recovery/Fast Retransmit procedure. Still, the fact remains that due to the longer outage of the connectivity more packets will be lost. What the net effect is, and how much worse off we are when the

protection switch times get larger, will be addressed in Section I.5. The issue of protection switching speed gets even more interesting when those actions are performed by an (electrical) PSC layer. In that case, the "switched flows" redirected via a backup path may have to share bandwidth with "fixed flows" already present on (parts of) the backup path and whose routing is not changed in response to the failure. Thus, not only the flows crossing the failing network part, but also other flows will be affected by a failure. The speed of protection switching will have a major impact on how this interaction between "fixed" and "switched" flows evolves.

The speed of protection switching mainly affects the number of losses inflicted on the suffering TCP flows. A second important effect of protection switching we highlighted before was the change in RTT. Usually the backup path, to which TCP flows will be switched by a GMPLS recovery mechanism, will be longer than the originally followed working path. Therefore, when switching TCP flows to the backup path, they will experience a sudden increase in RTT, possibly causing their RTO timer to expire. Indeed, this timer uses an estimate of the RTT (see Section I.3) that is based on the shorter original working path. This results in unnecessary retransmissions of segments that simply had to make a longer journey than their predecessors. Exactly how detrimental this effect is, is the subject of a second series of simulations presented in Section I.6.

A third set of experiments, focusses on the switch-back operation that may follow a repair of the network failure. When flows are switched back to the original paths, this results in a decrease of the RTT. Moreover, depending on the GMPLS layer at which the recovery actions are performed, out-of-order-delivery and/or packet losses may be inflicted on the TCP flows. If the GMPLS layer is a merge-capable one —such as electrical MPLS— the PML will merge packets redirected to the original working path with those still traveling on the longer backup path. Obviously, out-of-order delivery will result, causing some innecessary retransmissions. When both incoming links (along working path and backup path) at the merge point are highly loaded, also buffer overflows (implying packet loss) may occur. For non-merge-capable GMPLS technologies —such as any optical layer, recall Figure I.3— after the switch-back operation the packets still underway along the backup path will be discarded at the PML, thus requiring retransmission. The difference in TCP behavior for electrical and optical cases following the switch-back operation are treated in Section I.7.

Using the insights gained by adressing the outlined issues, we will try to asses the main differences between the prevailing GMPLS protection strategies. A case study will compare them from a TCP point of view in Section I.8.

I.4.2 Simulation approach

The answer to the questions raised by the four topics is sought by means of simulations, using the wide-spread tool Network Simulator (a.k.a. ns–2) [24]. This is a discrete event simulator targeted at networking research, providing substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks.

To investigate a particular protection scheme, or one of the parameters playing an important role in its effects on TCP behavior, we will always refer to the same

generic topology (or part thereof). The network we consider is made up of an (G)MPLS backbone in which we simulate a link failure, as sketched in Figure I.5. The parameters used for the simulation are listed in Table I.1. We have chosen to keep the bottleneck —causing the losses that will limit the bandwidth troughput generated by the TCP flows— to be located in the access part of the network (at least under failure free conditions) by giving the access links a lower bandwidth. The propagation delay of the links was set in the range of 1–60ms depending on the scenario at hand, resulting in RTTs of the order of one to a few hundred milliseconds (see resp. sections for exact values).

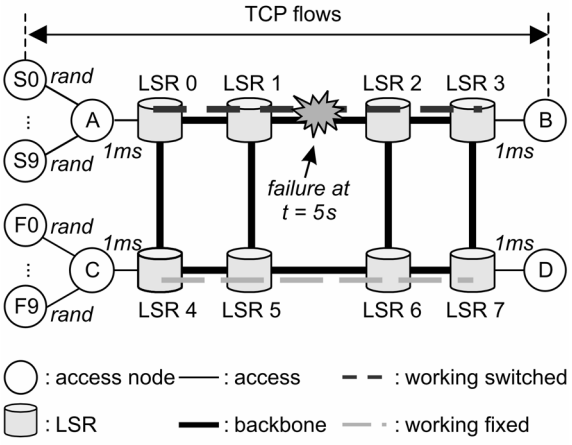


Figure I.5: Illustration of general set-up of our simulations. Each LSP carries an aggregate of 100 TCP flows, originating at nodes Sx that are connected to the backbone via access links, of which the first (to node A) has a randomly generated length (in terms of propagation delay). The TCP sources are started at different random times.

The packets flowing through this network follow predefined routes (paths), for which we exploit the MPLS provisions of the simulation tool —evidently, the exact routing paths depend on the protection mechanism under study. Along these paths, an aggregate of many TCP flows is sent. In order to reflect that each of these flows usually will be starting and/or continuing outside the backbone, and therefore may experience different delays, we have added different access nodes. Each of these access nodes will act as the source of a set of TCP connections. The access nodes will be connected to the rest of the network through access links having randomly generated propagation times uniformly distributed in the range [10ms,100ms]. In addition, to avoid unnatural synchronization between flows originating at the same access node in our simulated topology, we also use the random generator to determine the starting times of the TCP flows. Consequently, the network will be loaded with a mixture of TCP flows with uncorrelated starting times. Furthermore, by experiencing different RTTs, the flows will have different reaction times to

network changes affecting their RTTs; indeed, both the RTO timeout mechanism and the ACK-clocking mechanism are largely dependent on this RTT.

Table I.1: Common simulation parameters.

Parameter	*Value*
TCP	TCP NewReno [23]
No. of source nodes	10
No. of TCP flows	10 per source node (thus per LSP 100 in total
Access bandwidth	80% of backbone bandwidth; (90% for the timing effects in I.5)
rand delay	randomly generated propagation delay using uniform distribution in [10ms,100ms]

The two main parameters that will be varied for the scenarios used to address the topics outlined before are:

- ***Protection speed:*** This is the time that elapses between the occurence of the link failure, and the switch of the flows to the alternative path. It will be denoted by δ.
- ***Change in RTT:*** This is related to the extra propagation time the TCP flows will experience when they are switched from the primary to the backup path. For propagation times, we will use roman letters, usually d. The resulting difference in RTT between primary and backup paths will be denoted by ΔRTT.

Table I.2: Common simulation scenario.

Time	*Event*
t = 0s	Start of simulation
0.5s < t < 1 s	TCP sources start to generate traffic
t = 5s	Link LSR1–LSR2 goes down
t = 5s + δ	Protection switch is carried out
t = 10s	Link LSR1–LSR2 is up again
t = 15s	End of simulation

The scenario used to investigate the effect of changeing these parameters will consist of three periods of five seconds each, as summarised in Table I.2. During the first period, the TCP sources will start and gradually fill the network with traffic. At t = 5s, the link LSR1–LSR2 will go down. Subsequently, at t = 5s + δ, the protection switch will be carried out, rerouting the affected flows along the backup path. In order to be able to precisely influence the timing, we perform the switch ``manually'' exactly δ (delta) seconds after the link went down. That means that all packets in transit on link LSR1–LSR2 at the time of the failure (t = 5s) and arriving at LSR1

(or LSR2 for the ACKs) between t = 5s and 5s + δ will be lost. The last 5s period of our simulation, the link LSR1–LSR2 will be up again.

Each scenario was simulated 150 times different random seeds to generate the start times of the TCP sources, and the $rand$ propagation delays of the access links.

I.4.3 TCP goodput

The criterion we will use to evaluate the protection mechanisms is TCP goodput. This is the number of unique bytes successfully transmitted end-to-end (i.e. from TCP source to TCP destination) within a certain time interval, expressed in, for example, bytes per second. Clearly, this is what end-users of the network employing TCP will care about. In the graphs we present, we usually will express goodput relatively to the link bandwidth available to the (aggregate of) TCP flow(s). This means that we take the total of successfully transmitted bytes, as recorded in a variable *bytes* (see further), at fixed times $t_i = i \cdot T$, with an interval of duration T, and divide it by the maximum number of bytes that could have been transmitted (i.e. T multiplied by the bandwidth): the plotted values are given by Equation (4).

$$Good(t_i) = \frac{bytes(t_i) - bytes(t_{i-1})}{T \cdot bandwidth}, \quad \forall t_i = i \cdot T \tag{4}$$

Using ns-2, we record the goodput at the TCP receiver, which is called the Sink. We have slightly extended the ns-2 program to make a Sink keep track of the number of bytes that it has received in order; for this we use a variable named *bytes*. This variable, associated with a Sink, is updated upon the receipt of a packet, sent by the corresponding TCP source. Based on the sequence number contained within the TCP header, the Sink is able to determine whether the packet contains data that (i) hasn't been sent before, and (ii) is the data packet it expected to receive next (i.e. causing a new ACK to be sent out, which may be a partial ACK or a complete ACK). Only when both conditions are fulfilled, the variable *bytes* tracing goodput is incremented. Another way of describing this, is that upon each receipt of a packet, *bytes* is incremented with the newly ACKed data: the byte sequence number sent in the ACK minus the one sent in the previous ACK.

Consequently, when a packet loss has occurred, there will be no advance in goodput (as traced by *bytes*) for the TCP flow it is part of, until the packet has been successfully retransmitted. With the Fast Retransmit/Fast Recovery, this will occur only when a partial or complete ACK would be sent out. However, in the meantime new packets may have been sent out and successfully received by the Sink, causing duplicate ACKs, before the lost packet arrives. Hence, when finally the lost packet arrives, the *bytes* variable will suddenly increase with the difference between the highest received sequence number and that of the predecessor of the lost packet (assuming that since the losses triggering the Fast Retransmit/Fast Recovery, no other packets have been lost). When plotting the goodput evolution relatively to the available link bandwidth, this increase can obviously cause the goodput value for the corresponding time interval (given by Equation (4)) to be bigger than 100%.

I.5 The effect of timing: the influence of the speed of protection switching

I.5.1 Purpose and simulation scenario

The first issue we address is the impact of the speed at which the protection switch is carried out. As outlined in Section I.4.1, in a packet-switch-capable (PSC) GMPLS domain, the impact of the protection switch will be the biggest. In a PSC domain, the flows switched to a backup LSP will go into competition with other flows already present on (parts of) the backup LSP. The observed behavior will depend for a large part on the timing of this protection switch. Indeed, when the switch is performed very fast, the TCP flows being switched to an alternative path will still be sending at a relatively high rate (as they will not have detected any packet losses yet) when joining other flows on the links that are part of the backup path. This may cause severe buffer overflows, resulting in excessive segment loss. Therefore, it is not obvious that making a protection mechanism act as fast as possible is the best thing to do.

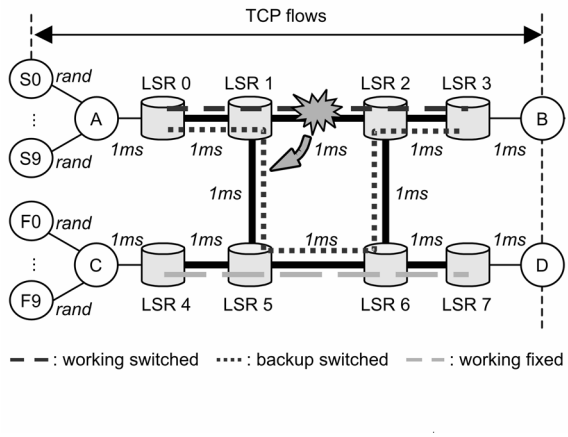


Figure I.6: Simulation set-up used to study the impact of protection switch speed on TCP.

The set-up used to investigate these matters is depicted in Figure I.6. We set up two sets of TCP flows: the "switched flows", originating at nodes S0–S9 following an LSP that will experience a failure and subsequent protection switch, and the "fixed flows"" originating at nodes F0–F9 that keep following the same primary LSP (unaffected by the failure); the ACKs will follow the reverse paths. The simulation period we focus on is when the link LSR1–LSR2 goes down, thus the interval [5s,10s].

I.5.2 Results for the electrical case

In Figure I.7, the evolution of goodput over time is depicted for different values of δ in the scenario presented before. There the heavy impact of the immediate buffer overflow on the fixed flows for $\delta = 0$ is clearly visible. Also, note that the time it takes for the interacting TCP flows to stabilize is in the order of a second or more.

We compared the different values of δ by considering $f(\delta) = \text{Good}(\delta)/\text{Good}(0)$, where Good(δ) is the total goodput, attained by the whole of fixed and switched flows, during the first 1.5 seconds after the link failure for switching time δ, as listed in Eq. (5).

We chose 1.5s as "integration interval", it being the relevant period for the differences in behavior for smaller values of δ. Using the data of our 150 runs, we constructed the histograms and corresponding normal fits as depicted in Fig. I.8. That graph shows that, on average, all cases of $\delta > 0$ result in a better goodput than having an immediate protection switch ($\delta = 0$). Numerical results from the comparison are summarized in Table I.3.

$$\begin{aligned} Good(\delta) &= \text{goodput for } \textit{all} \text{ flows in (5s , 6.5s] for switching time } \delta \\ &= \sum_{\forall \text{ flows}} \frac{bytes(6.5 \text{ ms}) - bytes(5 \text{ ms})}{1.5\,\text{s} \cdot bandwidth} \end{aligned} \tag{5}$$

These results seem to lead to the conclusion that pushing fast protection to the limit (i.e. extremely fast) may not be the wisest thing to do. From a qualitative point of view, the influence of changing δ can be explained as follows: if δ is set to zero (which corresponds to an immediate detection and subsequent triggering of the protection mechanism), the switched flows will join the fixed ones at LSR5 at a time when they are both sending at a quite high rate (limited only by the rate of the access links A–LSR4, resp. C–LSR8). This will result in an immediate buffer overflow at LSR5, causing a burst of losses affecting both flow categories. When introducing a certain delay (δ strictly positive), the switched flows will experience a higher number of losses (as packets cannot be forwarded along the primary path during that time), forcing the TCP sources to back off before they are switched to the backup path. The immediate buffer overflow at LSR5 will be avoided, and the fixed flows will be approached more "gently". We can indeed not avoid link LSR5–LSR6 becoming the bottleneck, but the buffer overflow at LSR5 will occur at a later time, and will cause fewer losses compared to the $\delta = 0$ case. Indeed, carrying out the protection switch as fast as possible in the considered case of electrical MPLS is not the most advantageous thing to do: it may be better to have a slightly slower protection action.

However, to decide what exactly is the "best" time to perform the protection switch, is not obvious. It at least depends on the link load (in the case presented above, when all links are up, backbone links are loaded for max. 90% due to the limits in the access part, but a protection switch results in a sudden load on link LSR5–LSR7 of almost 180%), the RTT experienced by the TCP sources (larger RTT means slower response to topology changes), and the number of concurrent TCP flows (larger number results in faster stabilization, up to a certain limit).

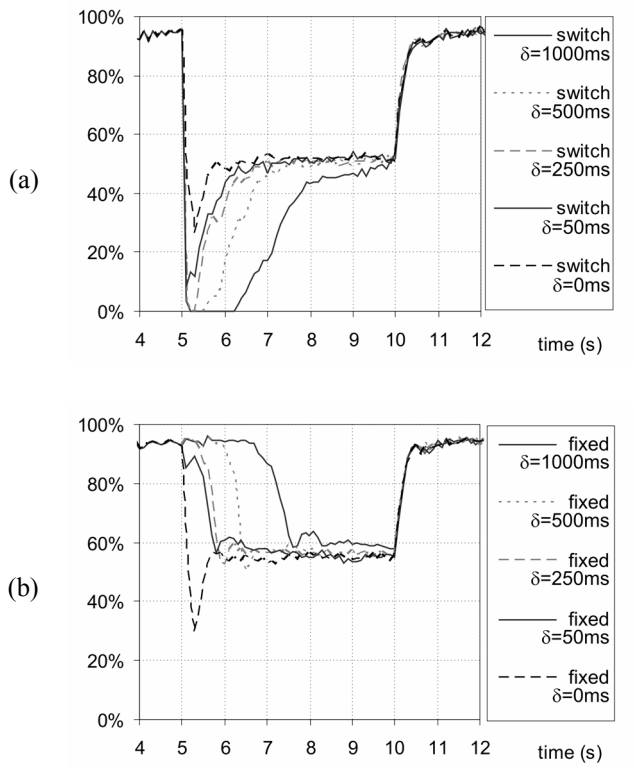


Figure I.7: TCP goodput evolution over time for different values of the switching time (delay) δ for the whole of the "switched" resp. the "fixed" flows. The goodput is expressed in % of backbone link bandwidth and was measured with a resolution of 10ms (i.e. T = 10ms in Equation(4)).

From a practical point of view, the results seem to indicate that from a TCP goodput perspective, having fast protection (order of tens of milliseconds) is not that bad —despite the sudden overload. This conclusion is probably even more true in cases where backbone links carry a vast amount of concurrent TCP flows (cf. faster stabilization than small number of flows, and therefore optimal delay shifts towards $\delta = 0$) and/or are fairly underloaded. Indeed, when backbone links do not form the bottleneck for TCP flows, interaction between switched and fixed flows will be limited. Indeed, repetition of the simulation scenario discussed above showed that —all other parameters left apart— for an access link bandwidth being a smaller fraction of the backbone bandwidth (e.g. 60% instead of 90%, thus resulting in maximal load on link LSR5–LSR6 of 120%), the optimal protection switch delay clearly shifts to lower values (towards $\delta = 0$). The simulations carried out seem to indicate that only if the 1 of protection switching is well below 50 ms, TCP effects may call for a stop to the efforts to minimize it.

All this however does not imply that extremely fast protection switching is a must for TCP: the differences in goodput for switching times δ in the range 0–250ms do not differ all that much, especially when the number of TCP flows is large.

Table I.3: Comparison of different protection switch delays. The left column represents the x-value corresponding to the average of f(δ), i.e. the peak of the normal fit in Figure I.8, minus 1 (this is Good(δ) /Good(0 ms)–1). The second column indicates the percentage of simulation results where f(d) < 100% (or, equivalently, Good(δ) < Good(0ms), whereas the rightmost column gives the number of simulation results where Good(δ) was maximal (i.e. compared to other delays).

δ	*Relatve difference in goodput compared to δ = 0*	*Fraction of random cases where δ is worse than δ = 0*	*Fraction of random cases where delay δ is best*
0.000 s	0.00%	0.00%	20.00%
0.050 s	+ 9.85%	24.00%	64.67%
0.250 s	+ 4.99%	36.67%	9.33%
0.500 s	+ 4.39%	42.67%	5.33%
1.000 s	+ 1.75%	49.33%	0.67%

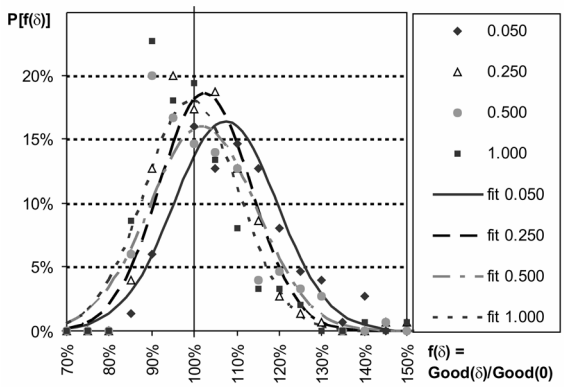


Figure I.8: Histograms (with a resolution of 5%) and normal fits for relative amount of goodput. A marker at (x,y) for a particular δ means that y% of the simulation results had f(δ) within [x,x+5%).

I.5.3 Results for the optical case

The simulation discussed above considered fast protection at an electrical MPLS layer. However, if fast protection is offered by an optical MPλS layer (or any other "circuit" layer, e.g. using TDM channels as "labels", as in SDH networks), we are in an altogether different situation. Indeed, then we will have no interaction between competing TCP flows: in that case we assume that the capacity for protection is reserved, and is fully available from the very instant the protection switch is carried out. Clearly, dynamic behavior of TCP in response to packet losses will still occur.

In this optical case, the intuitively clear conclusion we have drawn from our simulations is: the faster the protection switch at the optical layer is performed, the better (from a TCP goodput point of view). The simulations performed for this case had a link going down for a certain amount of time δ, without any protection actions taken at the MPLS level. For 140 random cases (random RTTs, etc., as before) and δ in {0, 5, 10, 20, 30, 40, 50, 250, 500, 1000 ms} we saw that in 94% of the cases, $\delta = 0$ was the best (only packets in transit on failing link are lost); in the remaining 6% of the cases, $\delta = 5$ms was the best (which is due to details in dynamic TCP behavior in some rather peculiar cases). Thus, the avoidance of TCP interactions is an advantage of protection at the optical layer with respect to protection at the merge-capable electrical MPLS layer. At the optical layer, even extremely fast protection switching does not seem to pose any problem (at least from a TCP point of view); clearly, the price paid for this is a higher cost in terms of network capacity to install (see, for example, Colle et al. [4]).

I.6 The effect of changing RTTs

I.6.1 Purpose and simulation scenario

Now we will focus on the effect of the sudden increase in RTT experienced by TCP flows when switched to a longer backup path. The simulation topology is depicted in Figure I.9. The total propagation delay of the span LSR1–LSR4–LSR5–LSR2 was set to d, with d in {1ms, 2ms, 10ms, 50ms, 100ms}. The original working path crossed the link LSR1–LSR2 with a propagation delay of 1ms. Thus, the increase in RTT is given by $\Delta RTT = 2 \cdot (d - 1\text{ms})$.

The average RTT of the original paths between the sources Sx and destination B, was 120ms, as can be derived from Figure I.9. So, when setting d to e.g. 100ms, this results in almost tripling the RTT (increase with $2 \cdot 99 = 198$ms).

This is what could happen when considering e.g. local loop-back protection for a failure of a link close to the egress node of the LSP (esp. in a network with a low connectivity degree): in that case the backup path will indeed be considerably longer than the working path, as it will be made up of almost the entire original path, its reverse, plus the link-disjunct alternative route to the egress (recall Figure I.2).

As the comparison of the different increases in RTT will surely depend on the speed of protection, we have repeated the experiment with a protection switch delay δ in {5ms, 10ms, 20ms, 100ms}.

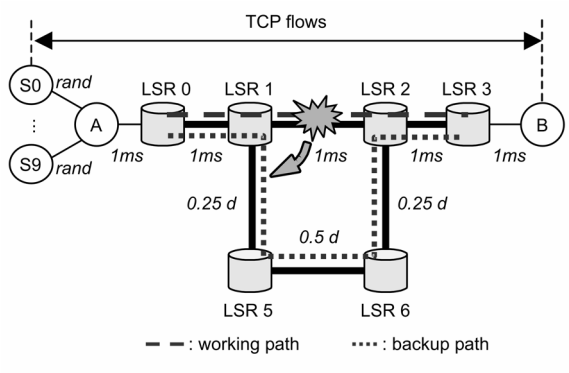


Figure I.9: Simulation topology used to investigate the influence of changes in RTT, caused by a protection switch, on TCP behavior. The access links had a bandwidth that was 80% of that of the backbone links.

I.6.2 Results

When we consider TCP flows switched to a backup path that is longer than the original working path, the increase in RTT they thus experience reinforces the temporary drop in the goodput evolution (due to lost packets and their retransmission).

Indeed, the switch to a longer path may cause TCP's retransmission timer to expire even when the protection switch is carried out quickly enough (small δ). Moreover, since the speed of increase in TCP window size (which will be reduced due to detected packet losses) is related to the RTT (cf. ACK-clocking property of TCP), a larger RTT also means slower recovery: it will take longer for the flows to fully exploit the available bandwidth again. Intuitively, we expect that the larger the difference in RTT, the more severe the penalty will be.

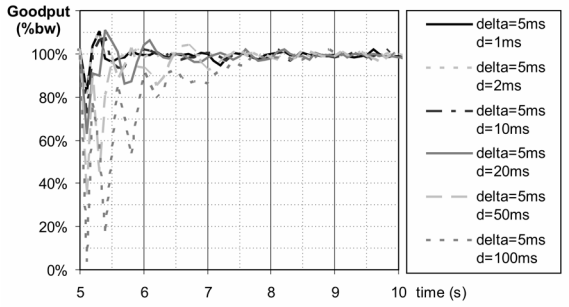


Figure I.10: Goodput evolution in interval (5s,10s] for δ = 5ms, and d in {1ms, 5ms, 10ms, 20ms, 50ms, 100ms}. It is plotted here with a resolution of 100ms, which means that every 100ms, the number of bytes successfully transported end-to-end in (t–100ms, t] was measured.

In Figure I.10 we have plotted the goodput evolution for the case $\delta = 5$ms, which shows the expected drop right after the link failure, and the subsequent gradual recovery. It confirms our qualitative discussion: the larger the difference in RTT, the more severe the drop in goodput is, and the slower the recovery. If we concentrate on the smaller differences (say δ up to 50ms), and compare the total goodput achieved within the first 1.5 seconds after the failure (denoted by Good(δ,d), with the same definition as in Equation (5)), we can construct a histogram for the relative goodput compared to $\delta = 1$ms (no change in RTT). This results in the graph presented in Figure I.11}, and the accompanying numerical data in Table I.4 with similar interpretations as those in Section I.5.

Table I.4: Comparison of different changes in RTT. The second column represents the x-value corresponding to the average of f(d), i.e. the peak of the normal fit in Figure I.11, minus 1 (this is Good(d)/Good(1 ms) – 1). The middle column indicates the percentage of simulation results where f(d) < 100% (or, equivalently, Good(d) < Good(1ms)), whereas the rightmost column gives the number of simulation results where Good(d) was maximal (i.e. compared to other delays)

d	*ΔRTT*	*Relative difference in goodput compared to d=1ms*	*Fraction of random cases where d is worse than d = 1ms*	*Fraction of random cases where delay d is best*
1 ms	0.00%	0.00%	0.00%	38.67%
2 ms	+1.67%	+0.13%	42.00%	46.00%
10 ms	+15.00%	–1.03%	75.33%	15.33%
20 ms	+31.67%	–3.13%	94.00%	0.00%
50 ms	+81.67%	–13.66%	100.00%	0.00%
100 ms	+165.00%	–31.48%	100.00%	0.00%

From these numerical data, we may conclude that for the considered switching time $\delta = 5$ms, the penalty of having a longer backup path is probably acceptable if the increase in RTT is limited to around 50% (resulting in a drop in goodput of less than 10%). One may wonder whether we can accept larger differences in RTTs when the reaction time of the protection mechanism is slower, i.e. when δ is larger. If it is sufficiently large, we expect to have the same drop in goodput for all cases of δ, immediately after the failure: it will completely fall back to zero, whatever the change in RTT is. Yet, the speed of recovery will be dependent on the value of the RTT change, as discussed before.

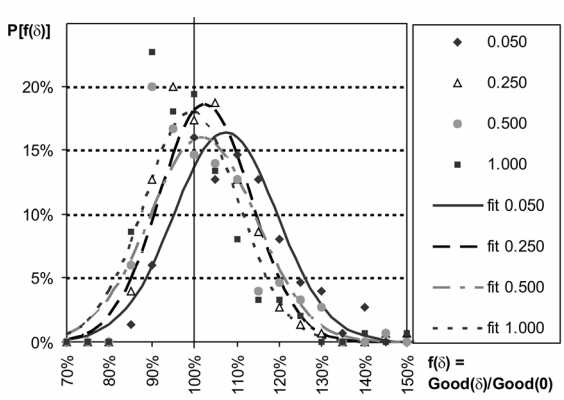


Figure I.11: Histograms (with a resolution of 1%) and normal fits for relative amount of goodput. A marker at (x,y) for a particular d means that y% of the simulation results had f(d) within [x,x+1%). Good(.) is the goodput attained by the total of all TCP flows in the period (5s, 6.5s].

The experiments with δ in {10ms, 20ms, 50ms, 100ms} resulted in the goodput evolution graphs during [5s, 10s] as plotted in Fig. I.12. These graphs confirm our qualitative predictions: when increasing δ, the drop in goodput is the same for all cases of d. For small values of d, the drop in goodput is significantly bigger when δ is increased. For larger d values the effect of a slower protection switch (larger δ) is not that the initial drop in goodput is larger, but rather that TCP recovers more gradually from it: the time to stabilize is about the same for different protection switch times δ (see Fig. I.12 for d = 100ms, where it takes about 3s for every δ), but the total goodput during this stabilization period is lower for larger δ.

In order to make a more accurate comparison, we again computed the total goodput during the first 1.5 seconds after the failure (as in Equation (5)). Note that this is smaller than the stabilization period for some cases, but taking a larger integration interval does not change the relative positions, in terms of better goodput figures, of the different (d,δ)-cases. Figure I.13 shows that, when increasing δ, the "best" corresponding d value shifts towards lower values. This indicates that when we wait longer to switch to the protection path, we can be more certain that a lower change in RTT performs better than a large one.

This is confirmed by Figure I.14, showing the total goodput attained by the total of all flows in (5s, 6.5s] for all the considered combinations of d and δ. From the same goodput data, we can calculate the penalty of increasing the change in RTT (i.e. increasing d) compared to having no change (i.e. d = 1ms), for different values of δ as listed in Table I.5. This all shows that the larger δ, the larger the relative penalty is of increasing the RTT (compared to keeping the same RTT).

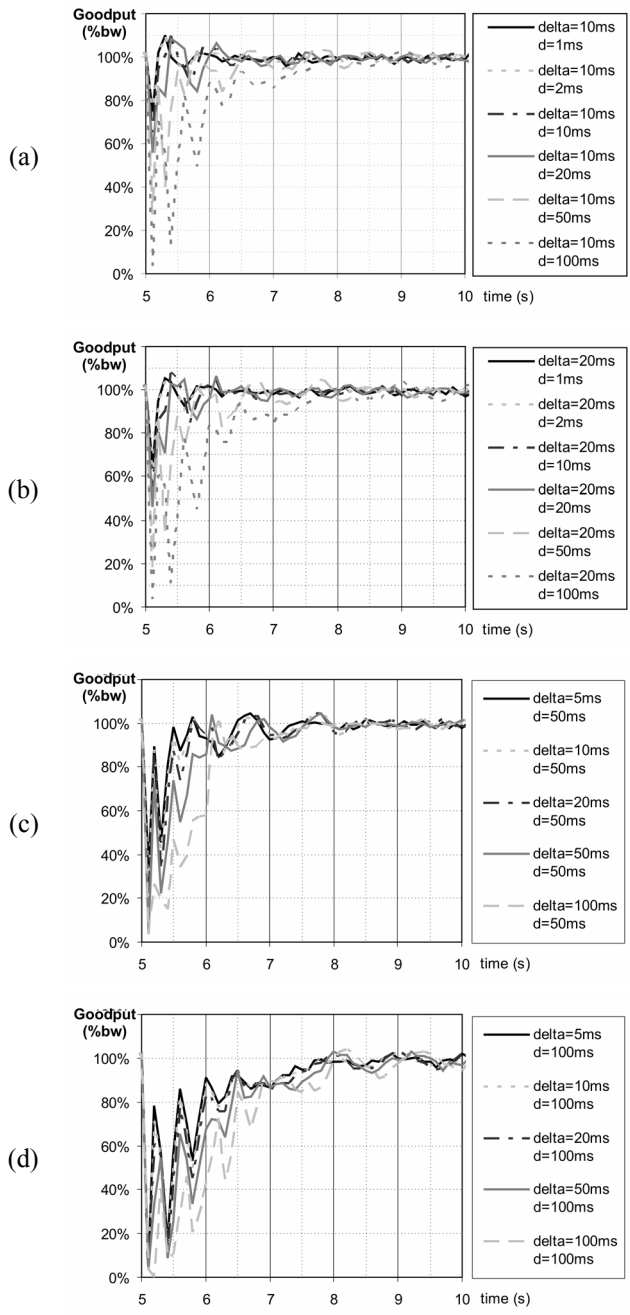


Figure I.12: Goodput evolution in interval [5s,10s] for all simulated combinations of (d,δ), grouped per δ.

Note also that Figure I.14 indicates that an increase in path length (and hence RTT) seems to be as damaging as increasing the protection switch time δ with the same order of magnitude (but keeping the change in RTT minimal), e.g. compare points (d, δ) = (5ms, 100ms) and (100ms, 5ms). This suggests that it is not obvious what is better: performing a fast protection switch to a longer (thus non-optimal) path —as e.g. with the local loop-back scheme— or rather delaying the switch a bit in order to find a route of about the same length as the original path (supposing that it exists) —which could be the case for path protection.

Table I.5: Comparison of different changes in RTT. An entry for (d,δ) gives the difference in goodput between d and d = 1ms for that particular δ, i.e. it is Good(d,δ)/Good(1ms, δ)–1, where Good(.,.) is the total goodput in (5s,6.5s]. Each entry is the average of 150 simulation runs using different random seeds.

d	*ΔRTT*	*δ =5ms*	*δ = 10ms*	*δ = 20ms*	*δ = 50ms*	*δ = 100ms*
1 ms	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
2 ms	+1.67%	+ 0.13%	– 0.20%	– 0.11%	– 0.23%	– 0.71%
10 ms	+15.00%	– 1.03%	– 1.65%	– 2.01%	– 3.43%	– 6.65%
20 ms	+31.67%	– 3.13%	– 4.20%	– 5.10%	– 7.82%	– 13.25%
50 ms	+81.67%	– 13.66%	– 15.43%	– 17.55%	– 22.56%	– 29.04%
100 ms	+165.00%	– 31.48%	– 33.83%	– 36.43%	– 43.10%	– 52.83%

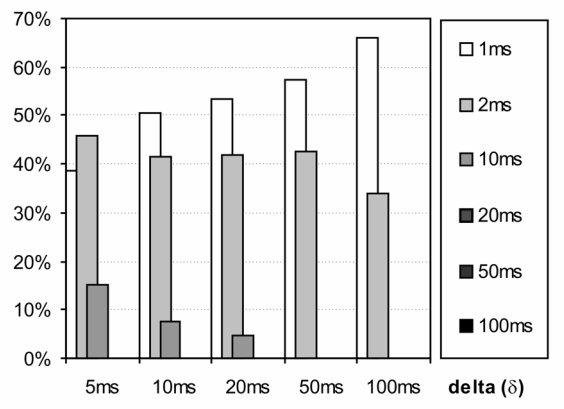


Figure I.13: Bar graph showing, for each δ, the percentage of simulation cases where a particular d was best. A bar with height y% for (d,δ) means that for y% of the simulations with δ, those for d resulted in the best value of Good(d,δ) (i.e. the goodput during (5s, 6.5s]).

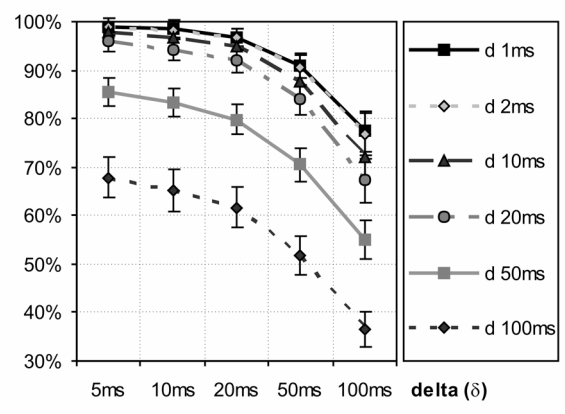


Figure I.14: Comparison of different values of d and δ. A marker with value y% means that averaged over 150 simulation runs, Good(d, δ) was y% of the maximal achievable goodput, i.e. that would be reached when every packet would be sent only once, and the access bandwidth would be fully used all the time. Good(d,δ) is goodput measured in (5s,6.5s].

I.7 Effect of a switch-back to the primary path

I.7.1 Purpose and simulation scenario

The previous sections concentrated on the effects of protection switching actions taken upon a link failure. However, when a revertive mode of operation is adopted, traffic is automatically switched back from the recovery path to the original working path upon the restoration of the working path to a fault-free condition. In this section, we focus on the effects of such a switch-back operation in two cases, denoted as "electrical" and "optical", and compare them with the non-revertive mode of operation where the flows keep following the backup paths (a strategy denoted as "none"). The "electrical" case corresponds with protection switching on a GMPLS level where merging of flows is possible, which tallies with protection actions in a PSC domain. The "optical" case reflects the behavior where merging is not possible. The results of the presented simulations assist in a comparative study of electrical and optical protection mechanisms.

The simulation topology is the same as the experiments for the switch-over (see Figure I.9). However, we now focus on the interval [10s,15s], where the link is up again. The actions taken in this interval will be one of the following:

- ***Electrical:*** the routes followed will be switched back to the original path, as indicated in the figure below, at t = 10s. In the electrical scenario, packets still underway on the backup path will be merged with the new ones forwarded along the original path. This merging operation takes place at LSR2 for the data packets, and LSR1 for the ACKs.

- ***Optical:*** merging is not possible. From the moment the first data packet arrives at LSR2 along the original path (coming from LSR1), the packets coming from LSR5 are discarded. The same goes, mutatis mutandis, for LSR1. As an aside, note that in our simulations using ns–2, we have used a kind of approximation: from t = 10s on, the paths are switched back to the original working path —as in the electrical case— but at t = 10s + x we have made the links LSR5–LSR1 and LSR4–LSR0 go down (fail), with x the propagation delay on link LSR1–LSR2 (1ms in our example).
- ***No action ("none"):*** the last option we consider is leaving the routing as it was (the flows continue using the backup path).

We have again considered the effects for d in {1ms, 2ms, 10ms, 20ms, 50ms, 100ms}.

I.7.2 Results

Before looking at the simulation results in more detail, we are already able to predict the following differences between the scenarios:

- ***Electrical:*** after the link has come up again, both flows are merged at LSR2. When d gets large, this means we have a fairly long period where the link LSR2–LSR3 is overloaded. Indeed, both the incoming interfaces LSR5–LSR2 and LSR1–LSR2 offer a continuous stream of packets for some time (max. duration = d – 1ms; less if TCP reduces its sending rate before). Therefore, we expect some packet losses as long as this overlap exists, at least if d is big enough (big with respect to the buffer size for the interface to LSR2–LSR3). In any case, also when d is small (but larger than the delay of 1ms on LSR1–LSR2), we will get out-of order delivery at the destination. This will result in duplicate ACKs, to which TCP will respond, as if it were the result of losses, by retransmitting some packets. So we certainly do expect a lower goodput than the "none" strategy, where we keep sending along the backup path.
- ***Optical:*** Here we effectively will have packet losses: as soon as LSR2 detects a signal again coming from LSR1, it will forward this and discard whatever is still coming from LSR5. The bigger d is, the more packets will get lost, but clearly no duplicate ACKs will be generated due to out-of-order delivery. Still, we again expect lower goodput values than with the non-revertive strategy ("none").

From this qualitative discussion alone, it is not clear yet which of the strategies (electrical or optical) will be the better.

Obviously, for strategy "none" the value of d has no impact on the goodput evolution.

When the RTT difference is negligible (d = 1ms, 2ms), there is hardly any noticeable difference between the three cases: for the "electrical" case, we will have no losses at the merging point LSR and the number of out-of-order deliveries will be very limited; for the "optical" case, the number of losses at the merging point will be very small. However, for d in {10ms, 20ms}, we see already a different behavior for the "electrical" and "optical" cases. The electrical case shows a tiny drop in TCP goodput (around 5% of the link bandwidth in the evolution graphs), due to some TCP flows going temporarily to the fast retransmit/fast recovery phase. However,

since it's due to out-of-order delivery rather than packet losses, TCP recovers rather quickly. For the "optical" case, the drop is more pronounced, as it is the result of packet losses rather than out-of-order delivery. When increasing the difference in RTT to bigger values (d in {50ms, 100ms}), drops also occur in the electrical case (due to buffer overflow). Additionally, there are some out-of-order deliveries of packets. At least for 100ms, the joint effect of these phenomena apparently results in a worse goodput behavior than in the optical case.

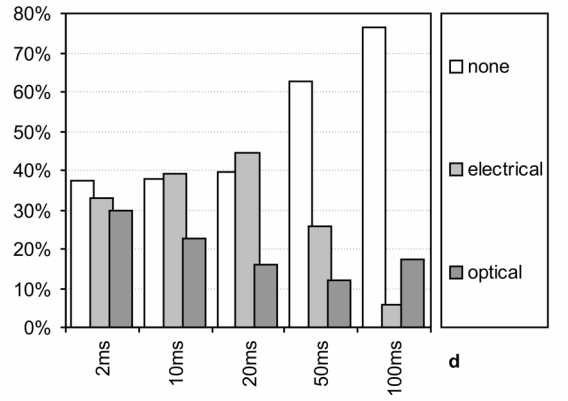


Figure I.15: Bar graph showing, for each d, the percentage of simulation cases where a particular strategy was best. E.g. a bar with height y% for "none" at d = 1ms means that for y% of the simulations with d = 1ms, "none" resulted in the best value of Good(.,d) (i.e. the goodput during (10s, 11.5s]).

To allow a more concise comparison, we again have calculated the goodput during the first 1.5 seconds after the switch-back operation (which covers the entire period where "optical" and "electrical" cases differ significantly). In Figure I.15, we show the fractions of the simulation cases where a particular strategy proved to be the best one. When the difference in round trip time is small (small d), it is not easy to distinguish what strategy is best. However, for big differences, the trend is clear: the strategy "none" is best. This means that only when the difference in RTT becomes large, the penalty of having a switch-back operation seems to be large enough. To distinguish between the "optical" and "electrical" cases is not evident from this figure, yet it seems to confirm that for big RTT changes (d = 100ms), the optical scenario is slightly better, but for smaller RTT increases the electrical merging protection switch seems to be preferable (and not much worse than adopting a non-revertive strategyA more detailed comparison between the optical and electrical scenarios is possible by comparing the ratio of the goodput attained by the two cases: Good (opt,d)/Good(el,d), where Good(s,d) is the goodput attained by all TCP flows in (10s,11.5s].

This comparison is presented in Figure I.16 and the accompanying Table I.6. For small RTT changes (d = 1ms, 2ms) the electrical and optical cases are more or less equivalent. For somewhat larger differences in RTT (d = 10ms, 20ms, 50ms)

there is a slight advantage in favor of the electrical case, and for even larger differences (d = 100ms) the optical strategy proves to be preferable. Note that the exact point where optical strategy becomes the better one depends on the buffer size for interface from LSR2 to LSR3: if this buffer were smaller, the electrical case would get worse for smaller changes in RTT (cf. smaller period of overlap when merging packet streams coming from working and backup paths will be sufficient to cause buffer overflow).

Table I.6: Comparison of different changes in RTT. The second column gives ratio of RTT using backup paths vs. that for the primary paths. The third column represents the x-value corresponding to the average of f(d), i.e. the peak of the normal fit in Figure I.16, minus 1 (this is Good(opt,d)/Good(el,d)–1). The rightmost column indicates the percentage of simulation results where f(d)=Good(opt,d)/Good(el,d) is greater than 100%.

d	*ΔRTT*	*Average change in goodput optical vs. electrical*	*Percentage of cases where optical is better than electrical*
1 ms	0.00%	– 0.01%	50.00%
2 ms	+1.67%	+ 0.02%	46.67%
10 ms	+15.00%	– 0.43%	36.00%
20 ms	+31.67%	– 0.62%	29.33%
50 ms	+81.67%	– 0.39%	41.33%
100 ms	+165.00%	+ 0.73%	59.33%

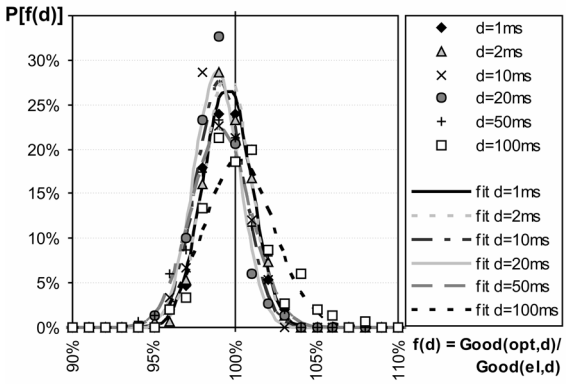


Figure I.16: Histograms (with a resolution of 1%) and normal fits for relative amount of goodput attained using "optical" strategy, compared to "electrical". A marker at (x,y) for a particular d means that y% of the simulation results had f(d) within [x,x+1%). Good(.) is the goodput attained by the total of all TCP flows in the period (10s, 11.5s].

All the results confirm the conclusions of the qualitative discussion at the beginning of this section: when the difference in RTT between working and backup paths is small, the electrical approach is preferable compared to the optical scenario. However, if the change in RTT is so large that in case of electrical MPLS it results in buffer overflows at the merge point, then providing protection at the optical layer may lead to better results (TCP-wise speaking).

I.8 Case study: comparison of MPLS protection mechanisms

I.8.1 Purpose and simulation scenario

With the previous simulation studies, we have gained insight in the effects of protection speeds and changing path lengths on TCP behavior. In the case study presented here, we apply that knowledge to the main GMPLS protection mechanisms.

The topology used for those simulations is depicted in Figure I.17. We will consider four scenarios, corresponding with the three MPLS recovery techniques discussed in Section I.2, and a loop-back variant. The protection actions that will be taken after the failure of link LSR1–LSR2 (at t = 5s, cf. scenario of Table I.2) and when the link has come up again (at t = 10s) for each of the four scenarios, are the following:

- ***local:*** This scenario will use local protection actions. After the link failure, at $t = 5s + \delta$, a protection switch will be carried out, as depicted in the upper part of Figure I.17, to a backup path that is d_l longer than the original working path. After the link has come up again, LSR1 (resp. LSR2) again forward packets along link LSR1–LSR2, and both flows are merged at LSR2 (resp. LSR1).
- ***path:*** When using path protection, signaling is necessary to inform the LSRs at the edges that they should switch. Therefore, the switch at LSR0 (resp. LSR3) will be carried out later than in the previous scenario: at $t = 5s + \delta + a$ (resp. at $t = 5s + \delta + c$). Again, traffic will be flowing along a path that is longer than the original path (increase with d_p), and flows will be merged at LSR3 (resp. LSR0). When the link is up again, the switch-back operation again needs signaling; this implies also the switch-back operation will be carried out later than in the local protection case. Note that this is less of a problem than in the case of the switch-over case: packets still sent along the backup path some time after 10s will not get lost (as opposed to those sent along the working path right after the failure).
- ***loopback:*** This is short for "local loop-back", the protection mechanism as presented in Section I.2. In this case, no signaling is required: at $t = 5s + \delta$, the protection switch is carried out. The backup path now is $d_p + 2a$ longer than the original path for flows from LSR0 to LSR3 (see Figure I.17); the reverse path LSR3–LSR2–LSR3–LSR7–LSR6–LSR5–LSR0 is $d_p + 2c$ longer than the corresponding original. When the link has come up again, LSR1 (and LSR2 for the reverse direction) resume forwarding along link LSR1–LSR2.

- ***loopbackvar:*** A disadvantage of local loop-back clearly is the presence of loops in the backup path. With this variant scenario, we combine local loop-back and path protection: at t = 5s + δ, a switch to the loop-back path is carried out, and a signal is sent to the end nodes of the path (LSR0 resp. LSR3). Thus, at t = 5s + δ +a the LSR0 stops forwarding along LSR0–LSR1, but switches to the backup path as in path protection. The revertive actions after the link failure has been resolved (after t=10s) are the same as in path protection.

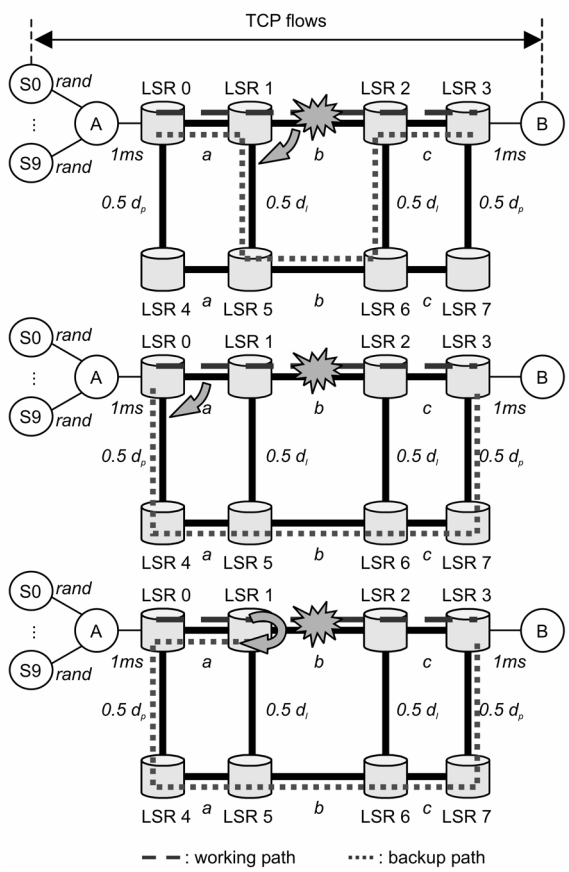


Figure I.17: Simulation topology and scenarios used to study the MPLS protection mechanisms; from top to bottom: local protection, path protection, local loop-back (for simplicity, only the path from LSR1 to LSR3 is shown, and not the reverse one). The access links had a bandwidth that was 80% of that of the backbone links. The times next to the links are the propagation delays used; for each of the access links Sx–A, it was independently chosen, using a uniform distribution, from [10ms,100ms].

Each of the above scenarios has been simulated for four sets of topology parameters, as listed in Table I.7. With a+c = 50ms and b = 1ms, this resulted in an average

propagation RTT of 216ms. The other parameters used are the common parameters listed in Section I.4.2.

Table I.7: Topology parameters used for the different scenarios.

Scenario	d_l	d_p	*a*	*b*	*c*
A	1ms	3ms	2ms	1ms	48ms
B	5ms	15ms	10ms	1ms	40ms
C	10ms	30ms	20ms	1ms	30ms
D	20ms	60ms	40ms	1ms	10ms

I.8.2 Switch-over

From the description of the scenarios above, it is clear that for a given topology parameter set, local protection will perform better than any of the other protection schemes. Indeed, the switching time (δ) will be the smallest, and so will the increase in RTT (given by $2 \cdot d_l$). A comparison of those two parameters for the various schemes is summarized in Table I.8.

Table I.8: Comparison of parameters influencing TCP behavior for different MPLS protection mechanisms.

Protection-mechanism	*Protection switch delay*	*ΔRTT*
Local	δ	$2 \cdot d_l$
Path	δ + a	$2 \cdot d_p$
Loopback	δ	$2 \cdot d_p + 2 \cdot a + 2 \cdot c$

For path protection, the increase in RTT d_p has been chosen larger than for local protection, which we expect to lead to worse goodput figures (see Section I.7). Moreover, due to an extra signaling delay, the switching time δ_P will be bigger as well (δ+a for path from LSR0 to LSR3), resulting in more losses than with local protection, again with a negative impact on TCP goodput. The local loop-back scheme solves the issue of faster switching time, but enlarges the increase in RTT by 2a+2c, i.e. twice the propagation delays along links LSR0–LSR1 and LSR2–LSR3. From the simulations presented in Section I.6, an increase in RTT has a comparable diminutive impact on goodput as an increase in switching delay δ of the same order of magnitude (e.g. compare (δ = 5ms, d = 100ms) and (δ = 100ms, d = 10ms) in Figure I.14. Therefore, we expect that local loop-back in the simulated topologies will perform not much better than path protection, in terms of goodput.

The loop-back variant, which keeps the fast switching time δ of local loop-back, but removes the superfluous loops in the paths, may perform slightly better than local loop-back, as the eventual RTT will be smaller. However, the switch at

LSR0 from the loop-back path to the path without loop is similar to the switch-back operation discussed in Section I.7. Consequently, it is foreseeable that the net advantage will be diminished.

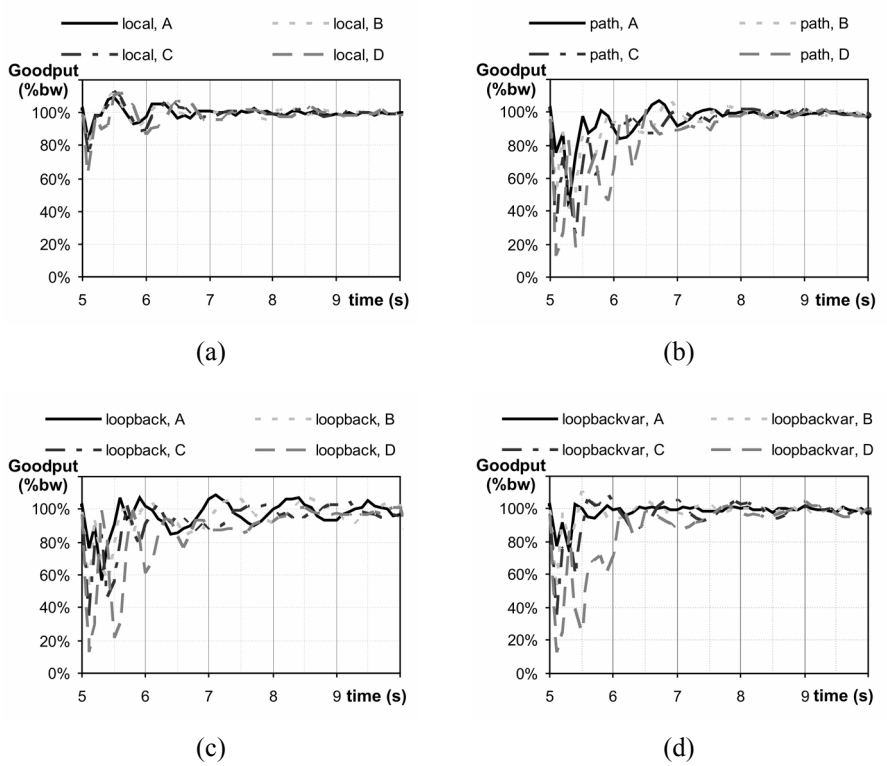


Figure I.18: Goodput evolution comparison for different protection mechanisms for each set of topology parameters (Table I.7), resulting in different protection switching time (Table I.8, second column). Goodput was plotted with a resolution of 100ms (meaning that value at t gives goodput in (t–100ms,t].

This qualitative discussion is confirmed by the simulation results. In Figure I.18 we plot the goodput evolution for the different MPLS protection mechanisms for each of the different topology parameters used. As expected, the different topologies do not result in big differences in goodput for the local protection scheme (Figure I.18(a)): the differences are similar to those observed in Figure I.12 (Section I.6) for comparable changes in RTT. For path protection, we observe larger drops in goodput and slightly slower recovery when changing the topology parameters. Again, this is in accordance with our previous findings. For the loop-back scenario, we see a similar evolution: the goodput evolution graphs alone do not show remarkable differences between the loop-back and path protection schemes either.

The curves for the loop-back variant shows a more surprising behavior. For the smallest changes in RTT (scenarios A, B, C), we see a drop that is very similar to the standard loop-back mechanism, but the recovery happens faster, which is due to the reduction in RTT at the time of switching to the backup path without loops. This behavior is in accordance with our qualitative discussion, as the goodput indeed lies between local protection and local loop-back. For the last scenario D, the recovery from the drop does not seem to happen faster than in the local loop-back case. This is because in this case the overlap between the packets still flowing along backup path and those directly forwarded along LSR0–LSR4 lasts too long: packets go lost. This is the same effect we saw in Section I.7 (see Figure I.15): there the penalty of the longer RTT also showed only when it was large enough to cause real losses instead of just out-of-order deliveries.

A more crisp overview of this comparison is presented in Figure I.19(a), where we have compared the goodput attained by the whole of all TCP flows during (5s,7.5s], which is the interval where the different mechanisms show distinct behavior. In conclusion, this figure shows that for the considered topologies local protection is the best thing to do (from a TCP point of view). As choosing between local loop-back and path protection is choosing between larger switch time or longer backup paths, they do not differ that much. The local loop-back variant, which eliminates the unnecessary loop in the backup path proves to be useful only when the loop is not very big (in terms of packets that can be in transit on it) with respect to the buffer capacity in the LSR that has to merge loop-free and loop-back paths.

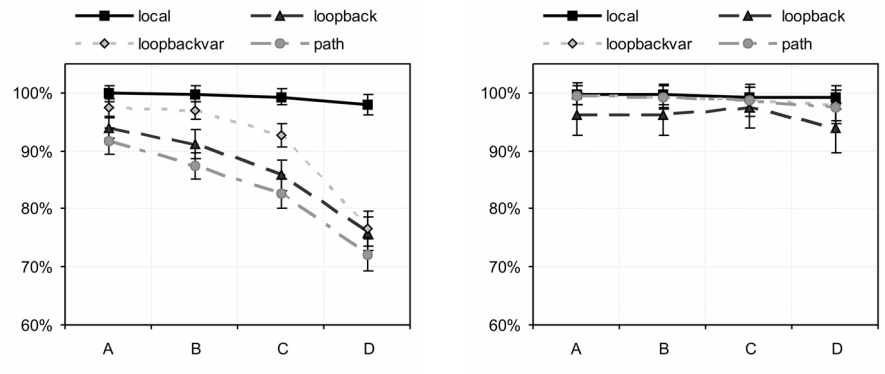


Figure I.19: Comparison of MPLS protection mechanisms. A marker for protection mechanism p and scenario s with a value of y% means that averaged over our 150 runs, Good(p,s) was y% of the maximal achievable goodput, i.e. the goodput that would be reached when every packet would be sent only once, and the access bandwidth would be fully used all the time.

I.8.3 Switch-back

In the previous section, we focussed on the protection switch itself. As discussed before, when a switch-back is performed when the failed link has been restored to a

fault-free condition, this has again impact on the TCP flows. As discussed in Section I.7, the only parameter that has an impact here is the difference in RTT between working and backup paths. This is the smallest for local protection, and thus we expect it to perform best. Path protection (and the local loop-back variant) has a lower difference in RTT than local loop-back, and —as opposed to the switch-over operation— therefore will perform better in this case. This is all confirmed by the results presented in Figure I.19(b).

I.9 Conclusion

In future IP-over-WDM networks, GMPLS may prove to be an excellent tool to administer and control the network. This network will be layered, in the sense that it will be built using technologies having different granularities of the traffic flows, and use different switching techniques (cf. packet, timeslot, lambda, fiber, … switching). GMPLS also opens the way to implement different resilience mechanisms, including various protection switching techniques. In this paper, we have focussed on the effects of protection switching on TCP, one of the most important protocols used by the clients of the network, to assist in a comparative study of the various mechanisms, and the layers at which they may be implemented (esp. optical or electrical).

We have considered interaction between switched flows and other flows already present along (parts of) the backup route, and analyzed the influence of the speed of protection. From this analysis, we have concluded that only if fast protection switching is pushed to the limit (sub-50ms), this may become a problem. Therefore, it is probably not advisable to push fast protection switching to the limit: not only is it not very useful (at least from a TCP perspective) to have extremely fast protection switching, a somewhat longer delay may even prove to be better (again from a TCP user's point of view).

Furthermore, we compared the effect of the changing RTT (stemming from a longer backup path) on TCP behavior for different switching times. The change in RTT indeed has a negative impact on TCP, and the more when switching times increase. The results also pointed out that no straightforward answer can be given to the question whether it is best to have a fast protection mechanism using longer (non-optimal) backup paths, or rather a slower mechanism that finds a backup path that is of about the same length as the working path: the differences in TCP goodput are small, and depend on the exact timing and topology parameters.

We also investigated the effect of a switch-back operation performed once the network failure has been restored. We compared the cases of optical (non-merge-capable) and electrical (merge-capable) protection switching. When the difference in path lengths between backup and recovery path gets large, the advantage of merging disappears: the optical non-merge-capable technique performs slightly better.

From the joint results of these case studies, we may conclude that providing protection at the optical layer has the advantage that it avoids interaction between TCP flows between different endpoints. When working and backup paths show substantial differences in length (compared to buffer sizes at electrical interfaces), the penalty of the technology being non-merge-capable on the TCP goodput seems

to be negligible. Yet, this comparison needs to be put in perspective by the study of e.g. the bandwidth requirements associated with the various mechanisms, as reported upon in [15] (showing that recovery at the optical layer suffers from higher capacity requirements, esp. for local protection).

In a final section, we presented a case study analyzing the differences between three well-known MPLS protection mechanisms. This analysis indicated that, TCP-wise speaking, local protection proved to perform best. Furthermore, local loop-back does not offer much advantage over path protection (small advantage at switch-over, slight disadvantage at switch-back). Eliminating the loop when using the loop-back mechanism is only advantageous when the loop is short enough. Again, this ranking of protection mechanisms has to be counterbalanced by criteria other than TCP goodput (e.g., Colle et al. [14] illustrating the expensiveness —in terms of bandwidth— of local protection, compared to other recovery mechanisms).

Acknowledgments

Part of this work has been supported by the European Commission through the IST-projects LION and DAVID, and by the Flemish Government through the IWT-project ITA/980272/INTEC.

The first author is a Research Assistant of the Fund for Scientific Research – Flanders (F.W.O.–Vl., Belgium).

References

[1] Evaluating the size of the Internet, "NetSizer Internet Growth Forecasting Tool", Telcordia (Online): *http://www.netziser.com.*

[2] The European Information Technology Observatory, Frankfurt, Germany (Online): *http://www.eito.com*.

[3] K. Claffy, G. Miller, K. Thompson, "The nature of the beast: recent traffic measurements from an Internet backbone", Proc. of INET 98, (Geneva, Switzerland, July 1998).
http://www.caida.org/outreach/papers/Inet98

[4] S. Leinen cited on *http://www.cs.columbia.edu/~hgs/internet/traffic.html*, Feb. 2001.

[5] A. Oram, ed., "Peer-to-Peer / Harnessing the Power of Disruptive Technologies", O'Reilly, March 2001.

[6] K. Struyve, N. Wauters, P. Arijs, D. Colle, P. Demeester, P. Lagasse, "Application, design an evolution of WDM in GTS's pan-european transport network", IEEE Communications Magazine, vol. 38, no. 3, (March 2000), pp. 114–121.

[7] Architecture for the Automatic Switched Transport Network, ITU-T G.ason, version 0.4, July 2001.

[8] N. Ghani, S. Dixit, T.-S. Wang, "On IP-over-WDM Integration", IEEE Communication Magazine, vol. 38, no. 3, (March 2000), pp. 72–84.

[9] O. Abdul-Magd, D. Awduche, C. Brownmiller, J. Eaves, R. Hoebeke, H.

Ishimatsu, M. Lazer, G. Li, M. Mayer, A. Nagarajan, L. Neir, S. Patel, E. Varma, Y. Xu, Y. Xue, J. Yates, "A Framework for Generalized Multi-protocol Label Switching (GMPLS)", Internet Draft, draft-many-ccamp-gmpls-framework-00.txt, Work In Progress, July 2001.
http://www.watersprings.org/pub/id/draft-many-ccamp-gmpls-framework-00.txt

[10] E. Mannie (ed.), et al., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", Internet Draft, draft-ietf-ccamp-gmpls-architecture-00.txt, Work In Progress, June 2001.
http://www.watersprings.org/pub/id/draft-ietf-ccamp-gmpls-architecture-00.txt

[11] P. Ashwood-Smith, A. Banerjee, L. Berger, G. Bernstein, J. Drake, Y. Fan,K. Kompella, E. Mannie, J.P. Lang, B. Rajagopalan, Y. Rekhter, D. Saha, V. Sharma, G. Swallow, Z. Bo Tang, "Generalized MPLS — Signaling Functional Description", Internet Draft draft-ietf-mpls-generalized-signaling-06.txt, Work In Progress, October 2001.
http://www.watersprings.org/pub/id/draft-ietf-mpls-generalized-signaling-06.txt

[12] X. Xiao, L. M. Ni, "Internet QoS: the big picture", IEEE Network Magazine, vol. 13, no. 2, (March 1999), pp. 8–18.

[13] D. Colle, S. De Maesschalck, C. Develder, P. Van Heuven, A. Groebbens, J. Cheyns, I. Lievens, M. Pickavet, P. Lagasse, P. Demeester, "Data-Centric Optical Networks and their Survivability", IEEE Journal on Selected Areas in Telecommunications, vol. 20, no. 1, (January 2002).

[14] D. Colle, P. Van Heuven, C. Develder, S. Van den Berghe, I. Lievens, M. Pickavet, P. Demeester, "MPLS recovery mechanisms for IP-over-WDM networks", Special Issue on IP over WDM and Optical Packet Switching of Photonic Network Communications Magazine, vol. 3, no. 1/2, (January/June 2001), pp. 23–40.

[15] D. Colle, A. Groebbens, P. Van Heuven, S. De Maesschalck, M. Pickavet, P. Demeester, "Porting MPLS-recovery techniques to the MPlS paradigm, Special Issue on Protection and Survivability of Optical Networks Magazine, vol. 2, no. 4, (July/August 2001), pp. 29–47.

[16] V. Sharma, B.-M. Crane, S. Makam , .K. Owens, C. Huang , F. Hellstrand, J. Weil, L. Andersson, B. Jamoussi, B. Cain, S. Civanlar, A. Chiu, "Framework for MPLS-based Recovery", Internet Draft, draft-ietf-mpls-recovery-frmwrk-03.txt, Work in Progress, July 2001.
http:/ /www.watersprings.org/pub/id/draft-ietf-mpls-recovery-frmwrk-03.txt

[17] K. Owens, V. Makam, V. Sharma, B. Mack-Crane, C. Haung, "A Path Protection/Restoration Mechanism for MPLS Networks", Internet Draft, draft-chang-mpls-path-protection-03.txt, Work in Progress, July 2001.
http://www.watersprings.org/pub/id/draft-chang-mpls-path-protection-03.txt

[18] R. Goguen, G. Swallow, "RSVP Label Allocation for Backup Tunnels", Internet Draft, draft-swallow-rsvp-bypass-label-01.txt, Work in Progress, November 2000.
http:/ /www.watersprings.org/pub/id/draft-swallow-rsvp-bypass-label-01.txt

[19] D. Haskin, R. Krishnan, "A Method for Setting an Alternative Label Switched

Paths to Handle Fast Reroute", Internet Draft, draft-haskin-mpls-fast-reroute-05.txt, Work in Progress, November 2000.
http://www.watersprings.org/pub/id/draft-haskin-mpls-fast-reroute-05.txt

[20] S. Floyd, "A report on recent developments in TCP congestion control", IEEE Communications Magazine, vol. 39, no. 4, (April 2001), pp. 84–90.

[21] J. Postel, "Transmission Control Protocol", RFC 793, Standards Track, September 1981.
http://www.ietf.org/rfc/rfc793.txt

[22] R. Braden, ed., "Requirements for Internet Hosts -- Communication Layers", RFC 1122, Standards Track, October 1989.
http://www.ietf.org/rfc/rfc1122.txt

[23] S. Floyd, T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm", RFC 2582, Experimental, April 1999.
http://www.ietf.org/rfc/rfc2582.txt

[24] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, H. Yu, "Advances in network simulation", IEEE Computer, vol. 33, no. 5,(May 2000), pp. 59–67.
http://www.isi.edu/nsnam/ns/ns-research.html