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.

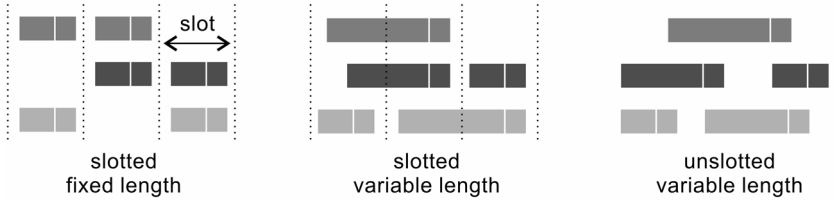


Figure A.1: Illustration of the various options for packet switching: fixed length packets, variable length packets, either in slotted or unslotted mode of operation.

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.

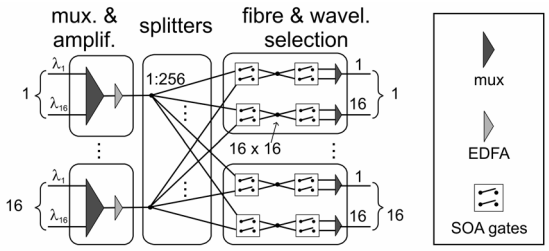


Figure A.2: A broadcast-and-select architecture as proposed in DAVID.

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.

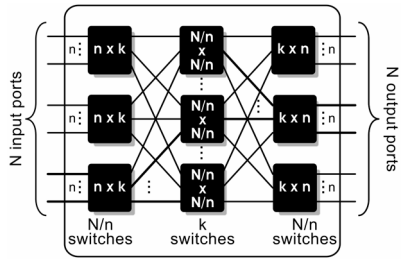


Figure A.3: A multi-stage Clos network for large switch dimensions.

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).

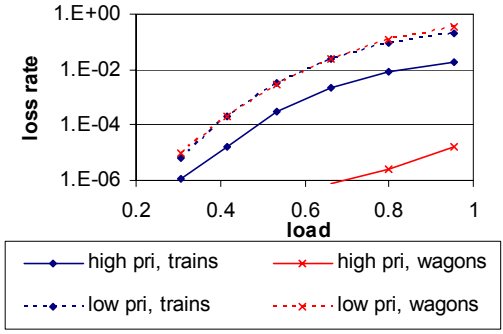


Figure A.4: Service differentiation for variable length packets in slotted OPS switches: segmenting packets into fixed size slots and treating them individually ("wagons") results in much stronger service differentiation than treating the packets as a whole ("trains").

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.

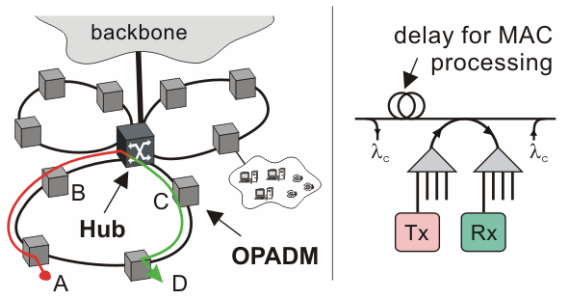


Figure B.1: Network architecture and Passive OPADM

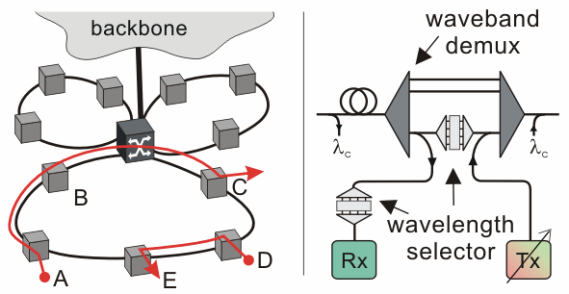


Figure B.2: Network architecture and Active OPADM

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.)

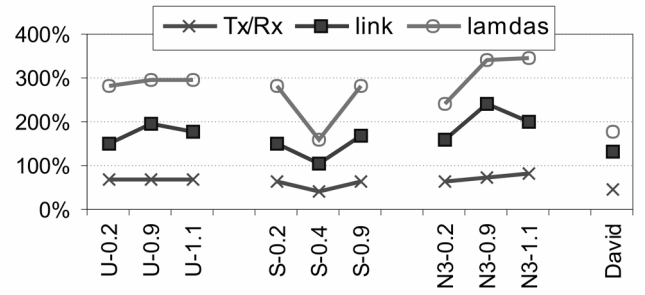


Figure B.3: Cost ratio Passive/Active; x-axis labels denote demand (U=Uni, S=Serv, N3=Neigh3; the number after the dash is the value of d).

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.

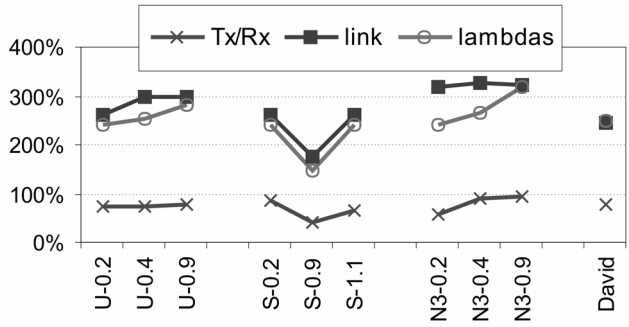


Figure B.4: Cost ratio (bands, B=4)/(no bands, B=1); x-axis labels denote demand.

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).

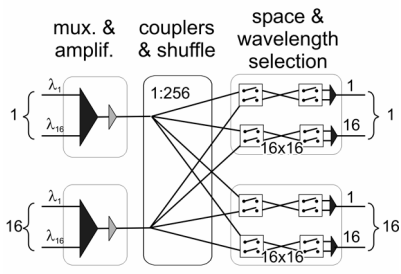


Figure C.1: A broadcast-and-select architecture as proposed in DAVID

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.

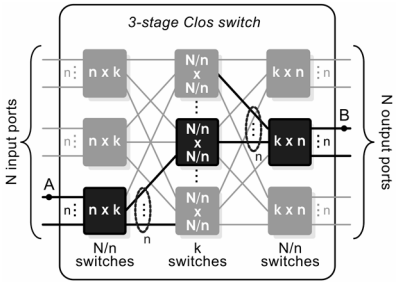


Figure C.2: Three-stage Clos architecture.

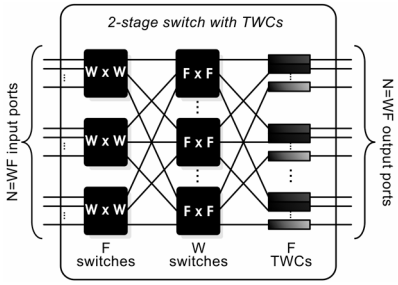


Figure C.3: Two-stage architecture with TWCs.

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).

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

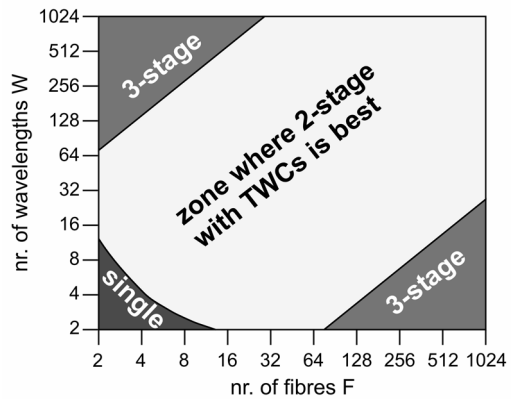


Figure C.4: Boundaries of the regions where each of the three node architectures is the most advantageous one.

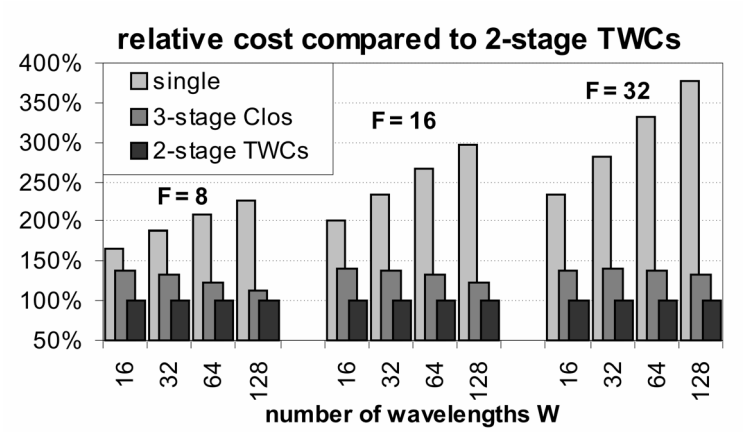


Figure C.5: Comparison of the three architectures for F=8,16,32 input/output fibres and W=16,32,64,128 wavelengths per fibre.

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.

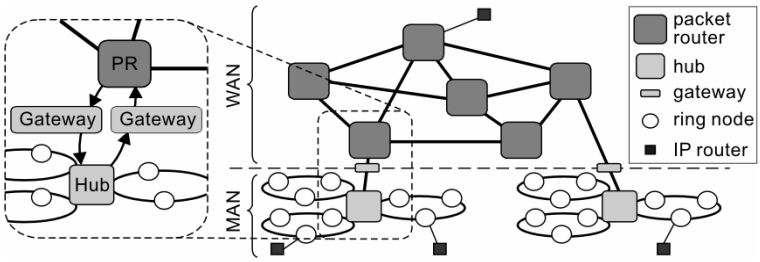


Figure D.1: The DAVID network architecture

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.

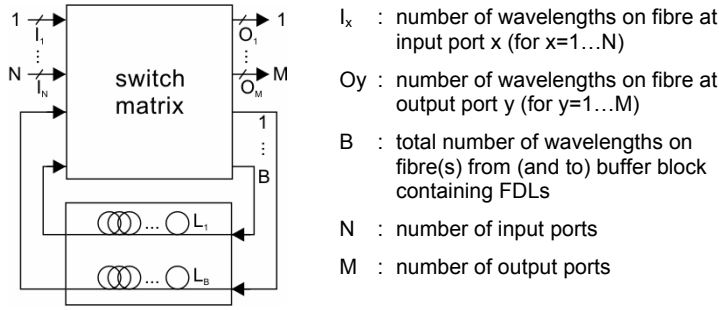


Figure D.2: The simulated node and its parameters.

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.

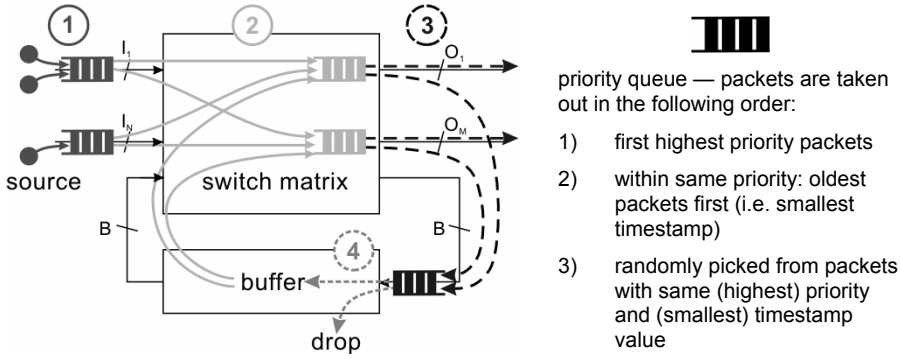


Figure D.3: Implementation of the scheduling strategy in the simulation program.

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).

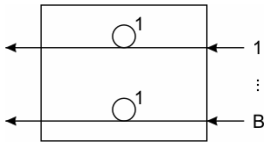


Figure D.4: Fixed loop size.

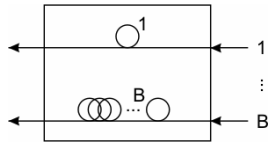


Figure D.5: Increasing loop size.

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.

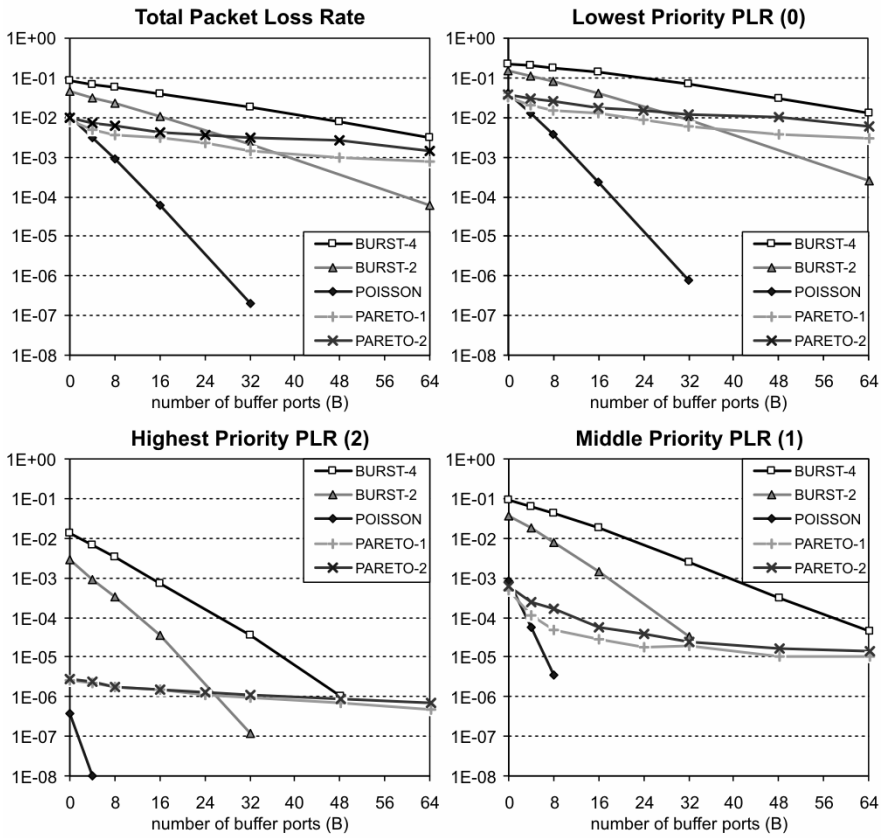


Figure D.6: PLR for increasing number of loops of length 1 (i.e. delay of single slottime).

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.

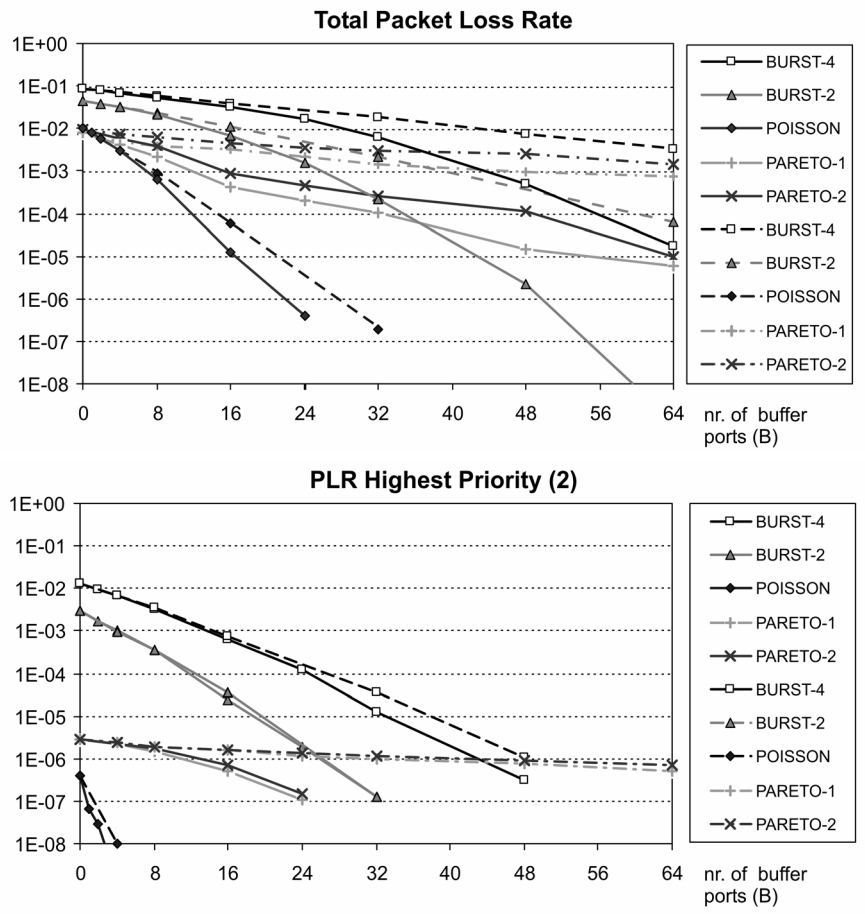


Figure D.7: Comparison of PLR for FDLs of fixed size (dashed lines) versus increasing size (full lines) for an increasing number of buffer wavelengths.

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.

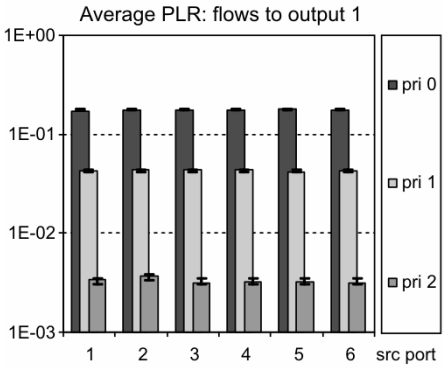


Figure D.8: PLR for individual sources generating traffic for output port 1; the deviation markers indicate the interval (avg-stdev, avg+stdev) taken over the six traffic sources per priority class.

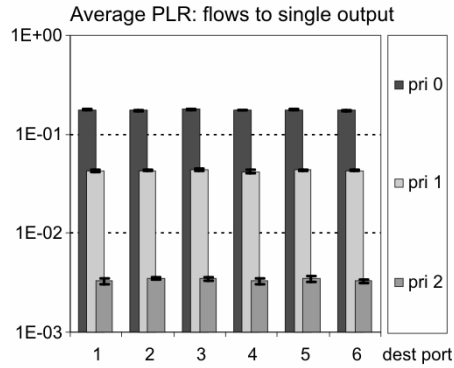


Figure D.9: Average PLR over all sources generating traffic for a particular output port; the deviation markers indicate the stdev on those averages.

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).

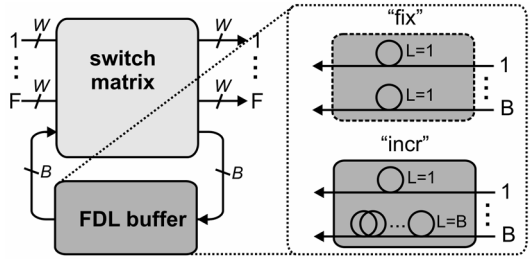


Figure E.1: Logical structure of the OPR with two sample FDL configurations.

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.

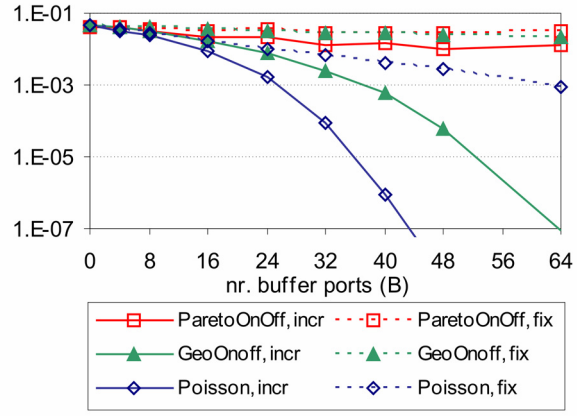


Figure E.2: Comparison of using a single FDL length (fix, dashed lines) and increasing FDL lengths (incr, full lines).

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.

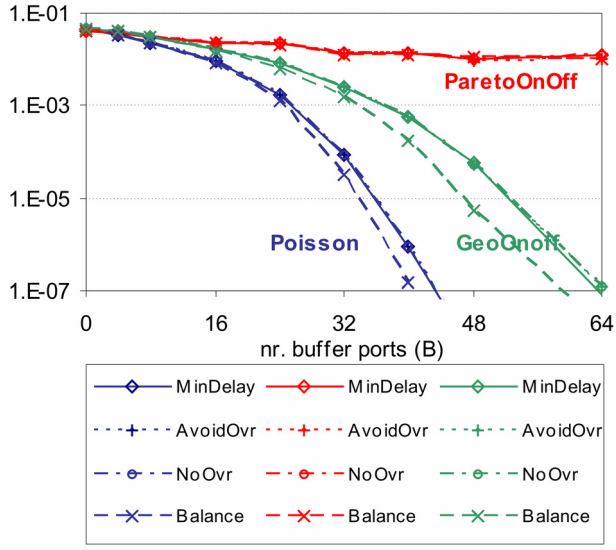


Figure E.3: Comparison of four buffer strategies for a feed-back buffer with increasing FDL lengths (configuration incr).

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.

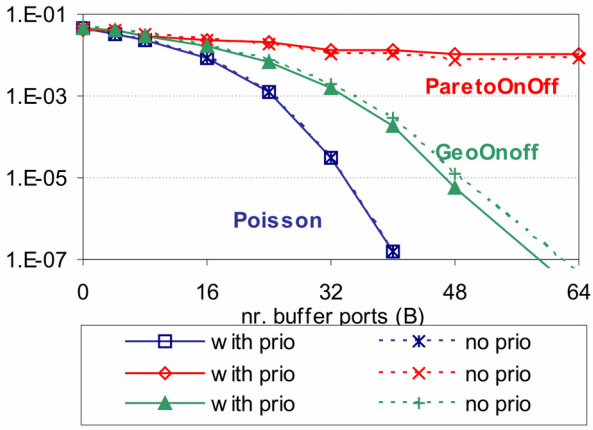


Figure E.4: Overall PLR when using service differentiation (with prio) vs. when ignoring traffic priority class information (no prio) for the buffer with increasing FDL lengths (incr) using the Balance strategy.

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.

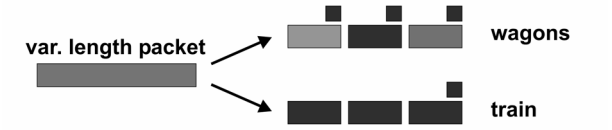


Figure F.1: Dealing with variable packets in a slotted switching concept: train versus wagons.

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.

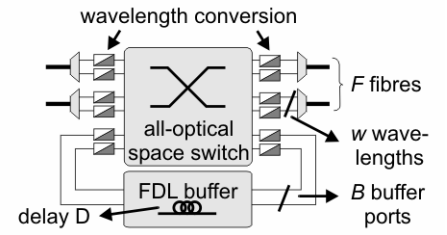


Figure F.2: The OPS switch architecture under study.

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.

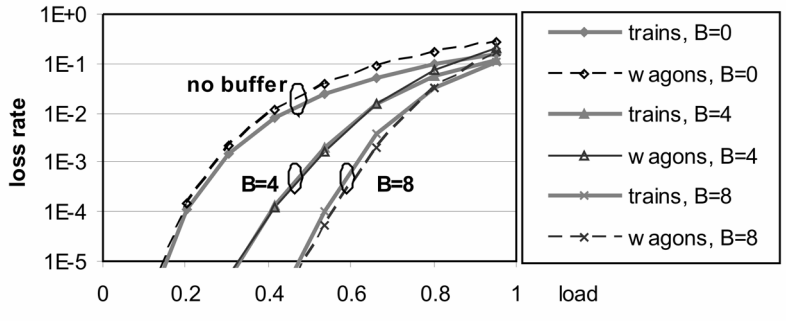
F.4 Trains or wagons

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)

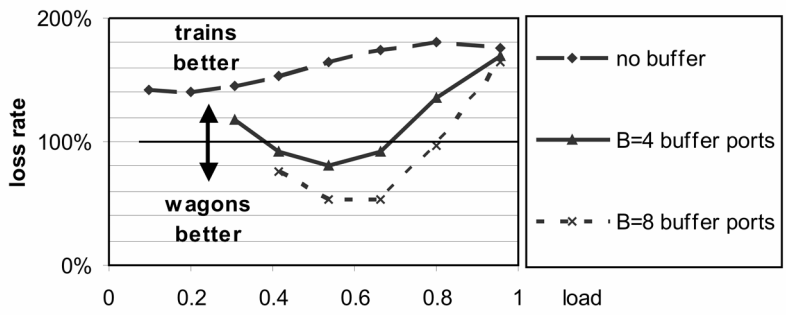


Figure F.3: Wagons vs trains: loss rates for increasing load, with B=0,4,8 buffer ports: (a) loss rate, (b) ratio of loss rate: wagons / trains.

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.

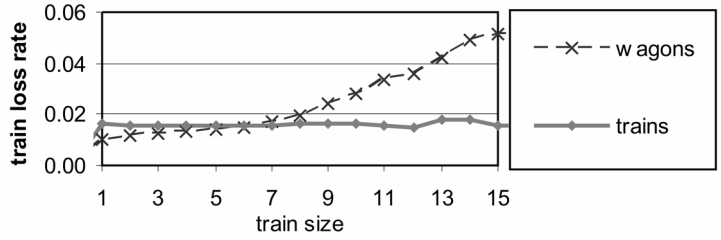


Figure F.4: Illustration of unfairness in loss rates for the wagon approach: loss rate per train length, for B=4 and a load of 0.62.

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.

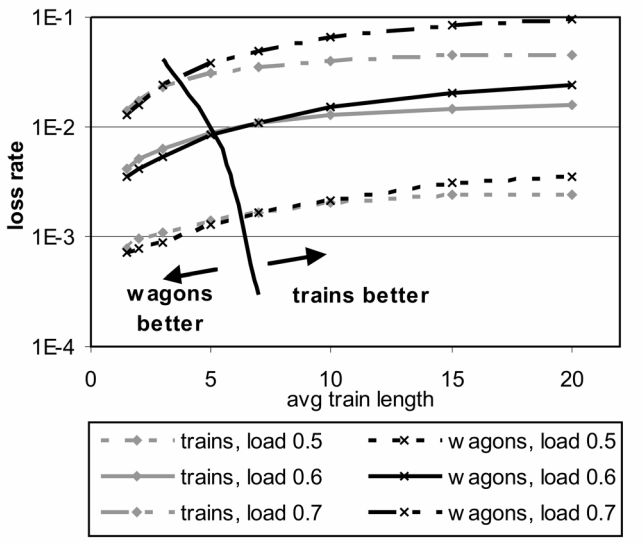


Figure F.5: Wagons vs trains: loss rates for increasing train lengths, with B=4 buffer ports, loads=0.5, 0.6, 0.7

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).

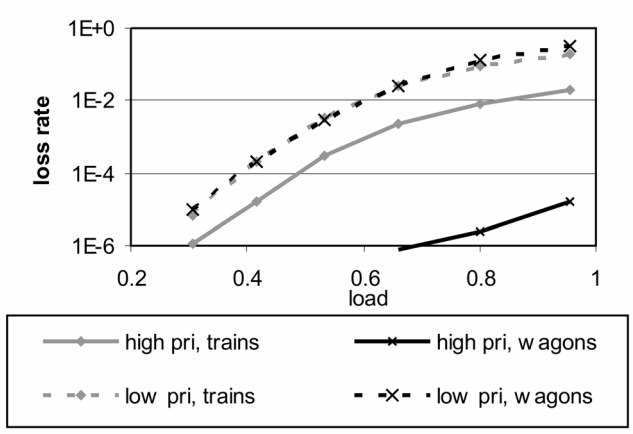


Figure F.6: Service differentiation of wagons vs trains: loss rates for increasing loads, with B=4 buffer ports.

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.

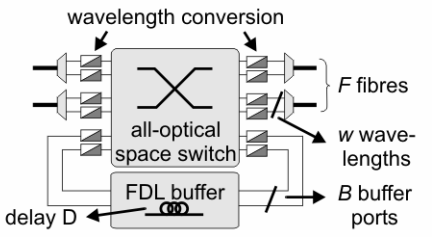


Figure G.1: Switch architecture under study.

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.

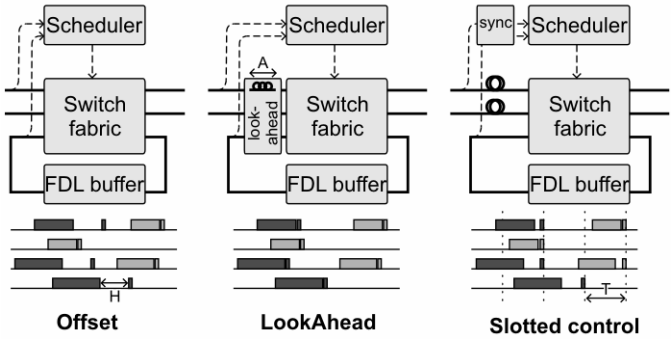


Figure G.2: The three QoS approaches we focus on.

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.

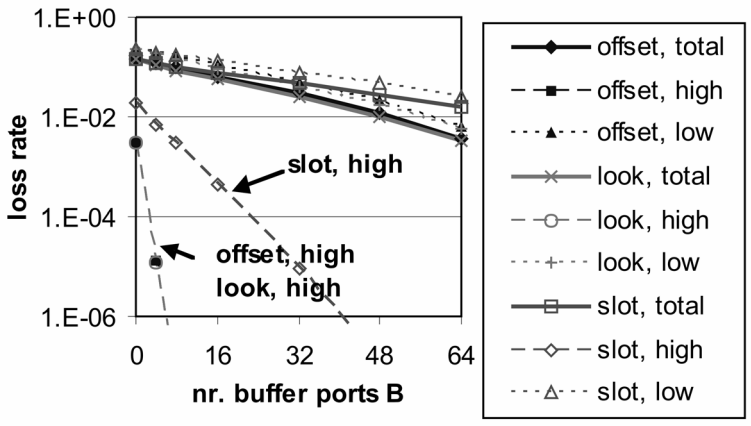


Figure G.3: Loss rates for increasing number of recirculating buffer ports B (load=0.8, F=6, W=8, 40% high priority traffic).

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).

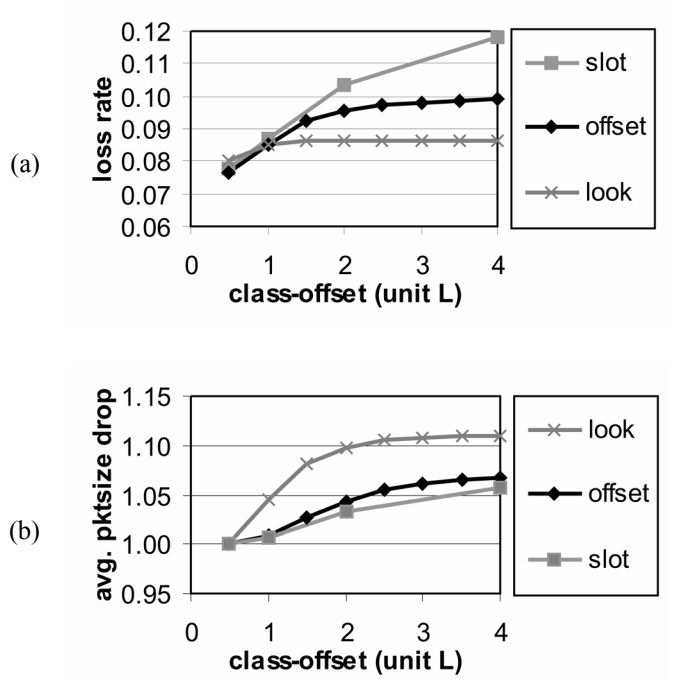


Figure G.4: (a) Overall loss rate and (b) avg. size of dropped packets for increasing class-offset O=H=T. (load=0.8, F=6, W=8, 40% high priority traffic; L=avg. packet length.).

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.

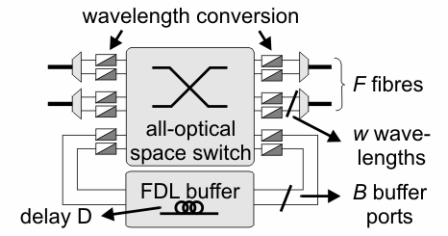


Figure H.1: Optical Packet Router architecture.

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.

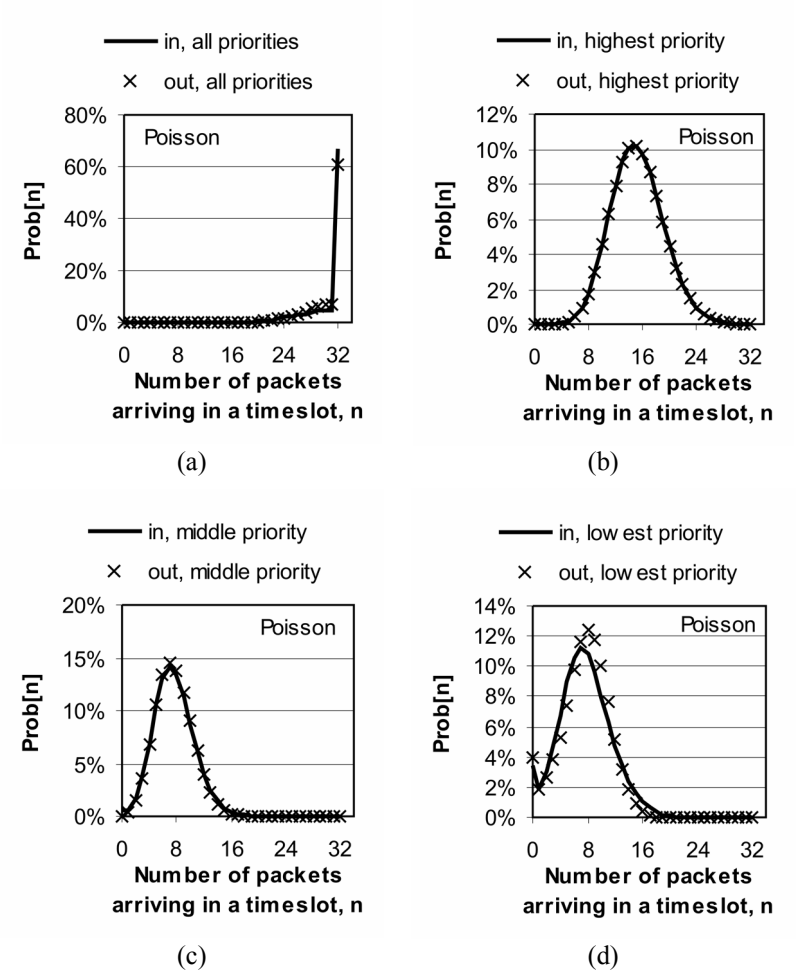


Figure H.2: Comparison of the histograms of the number of packets per timeslot (w = 32) arriving on a particular input port (solid lines) and leaving the OPR on a particular output port (marked dashed lines).

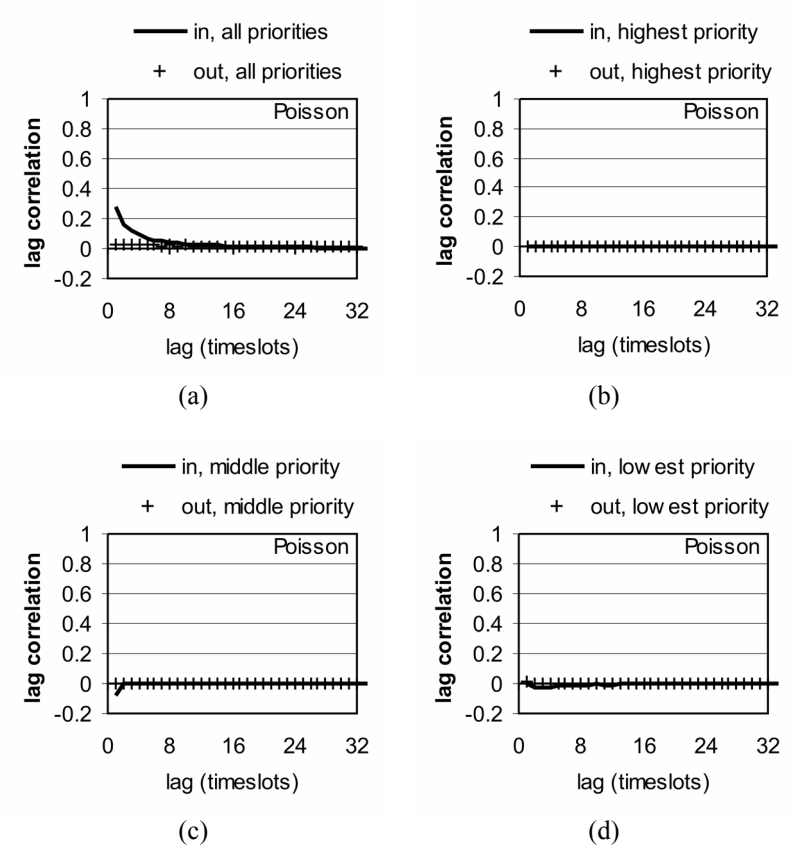


Figure H.3: Comparison of the correlation between the number of packets per timeslot (w = 32) arriving in timeslots τ and τ+lag on a particular input port (solid lines) and leaving the OPR on a particular output port (marked dashed lines).

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).

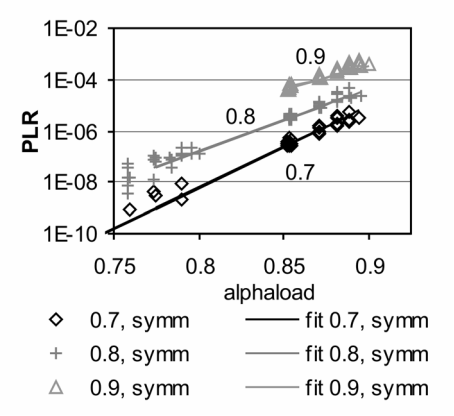
$$L_{i,f} = \begin{cases} L_f \,/\, 2^{1+(i-f) \bmod F} & , \quad i \neq f \bmod F \\ L_f \,/\, 2^{F-1} & , \quad i = f \bmod F \end{cases} \tag{2}$$

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.

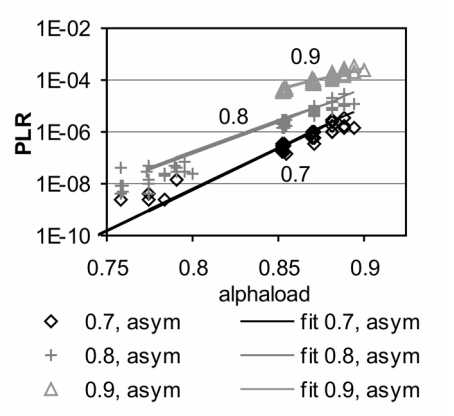
$$PLR(L_f \mid L_0 \ldots L_{F-1}) = p(L_f) \cdot 10^{r(L_f)\cdot(L_\alpha - L_f)} \tag{3}$$

$$p(L_f) = PLR(L_f \mid uniform) = a \cdot {L_f}^b \cdot 10^{c \cdot L_f} \tag{4}$$

$$r(L_f) = u \cdot L_f + v \tag{5}$$



(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.

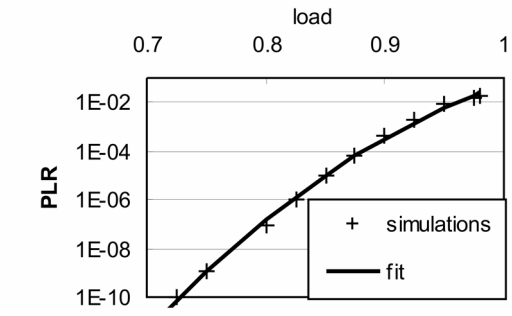


Figure H.5: Packet loss rates in case of a uniform traffic matrix for increasing loads.

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.

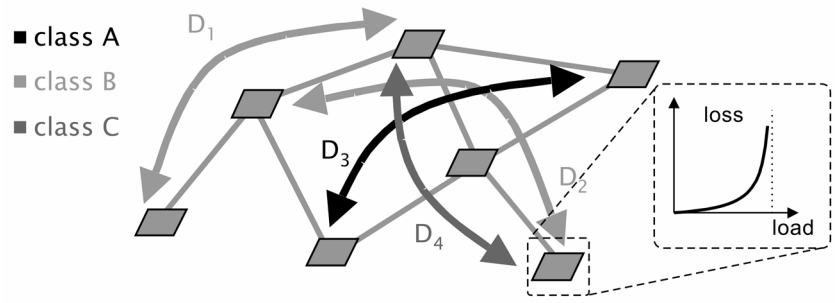


Figure H.6: Routing problem.

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.

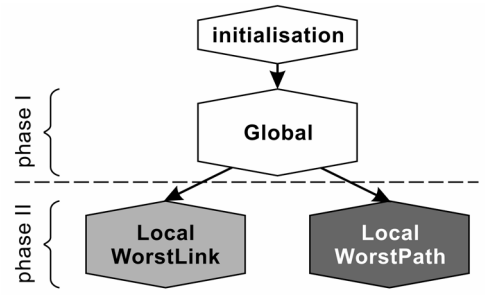


Figure H.7: Phases of the algorithm.

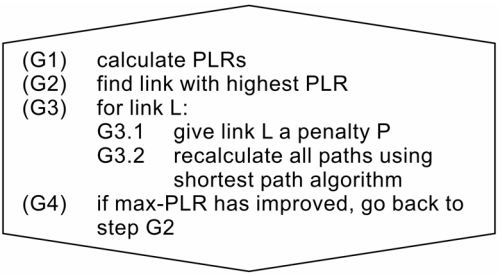


Figure H.8: Routing algorithm: algorithmic description of global phase.

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.

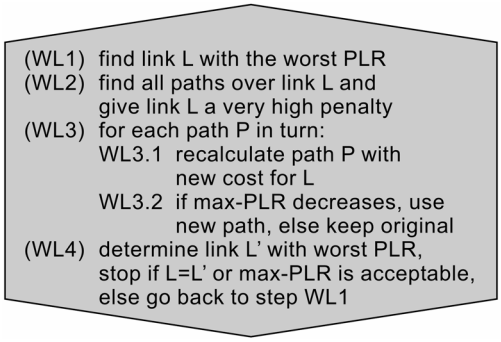


Figure H.9: Routing algorithm: algorithmic description of local phase – WorstLink variant.

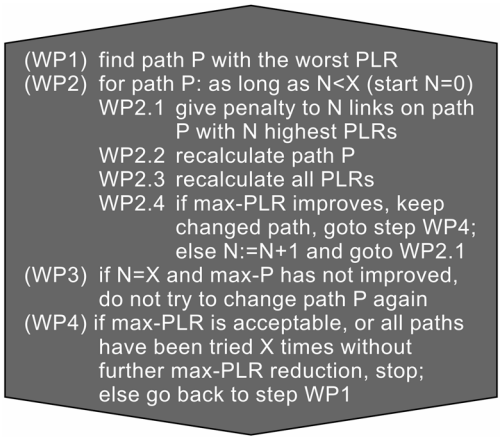


Figure H.10: Routing algorithm: algorithmic description of local phase – WorstPath variant.

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.

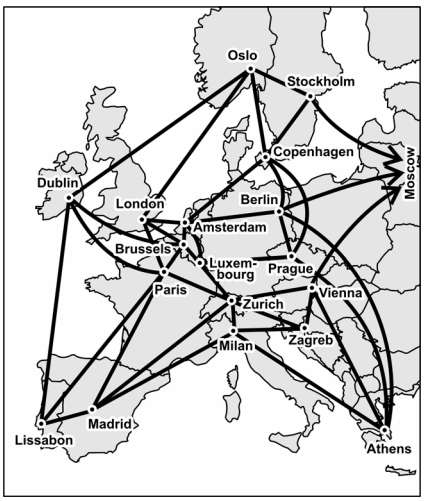


Figure H.11: The dense 19-node network.

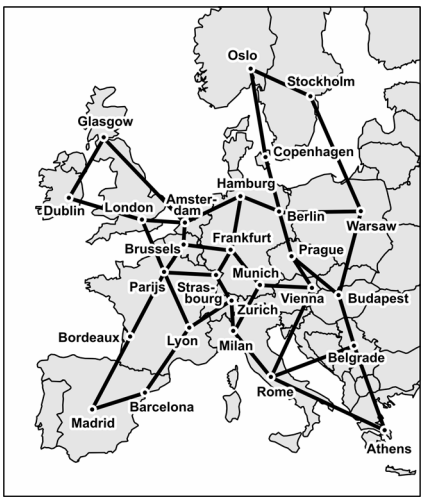


Figure H.12: The sparse 27-node network.

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.

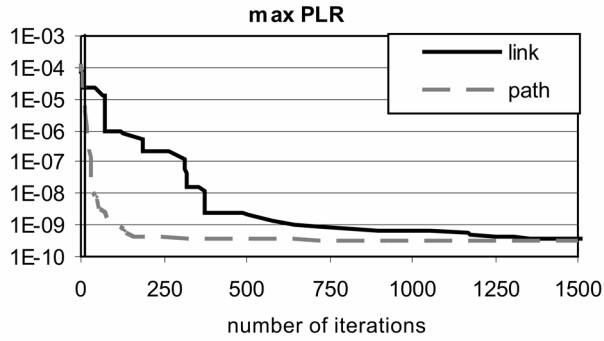


Figure H.13: Evolution of max-PLR (in function of number of iterations) for a random demand on the second pan-European network. The full line is the WorstLink version of the algorithm, the dashed line the WorstPath version. Only the first 1500 iterations are shown, since after that number the max-PLR 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.

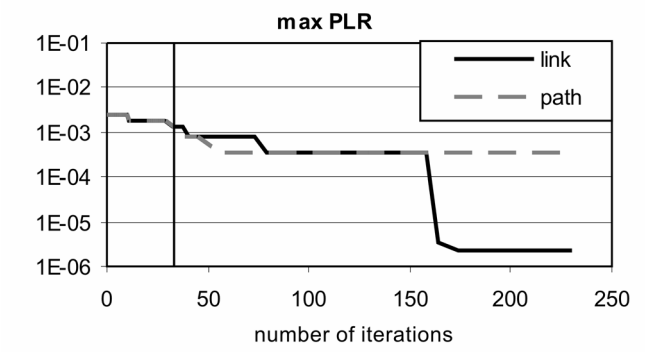


Figure H.14: Evolution of max-PLR (in function of number of iterations) for a random demand on the first pan-European network: we omitted the iterations beyond the first 250 because no further lowering of the max-PLR was attained. The vertical line marks the start of the local phase of the algorithm.

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.

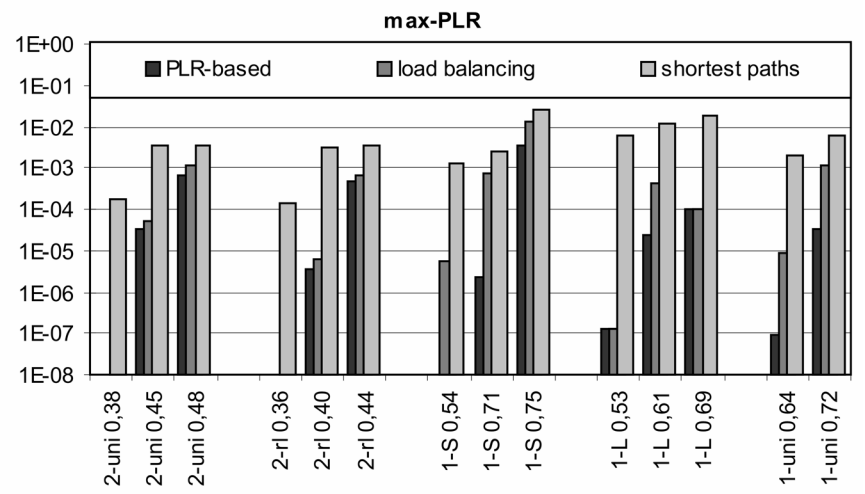


Figure H.15: Comparison of max-PLR reached with PLR-based routing, shortest path routing, and load balancing for different demands.

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.

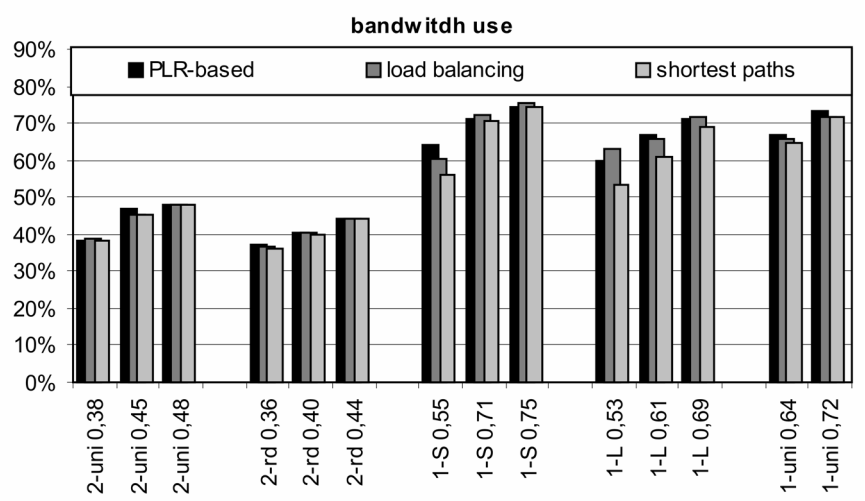


Figure H.16: Comparison of bandwidth-use (%) obtained with PLR-based routing, shortest path routing and load balancing for different demands.

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.

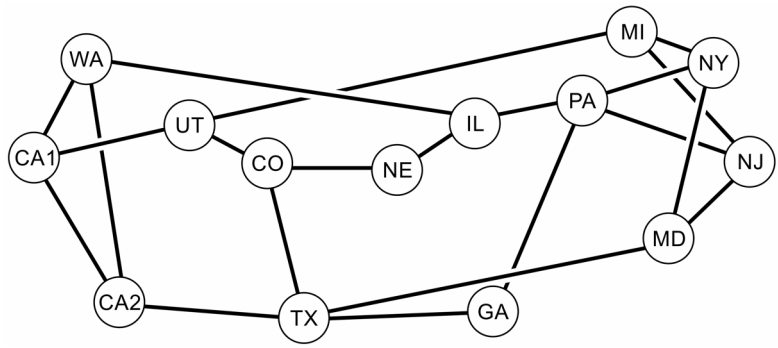


Figure H.17: The NSFNET.

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]).

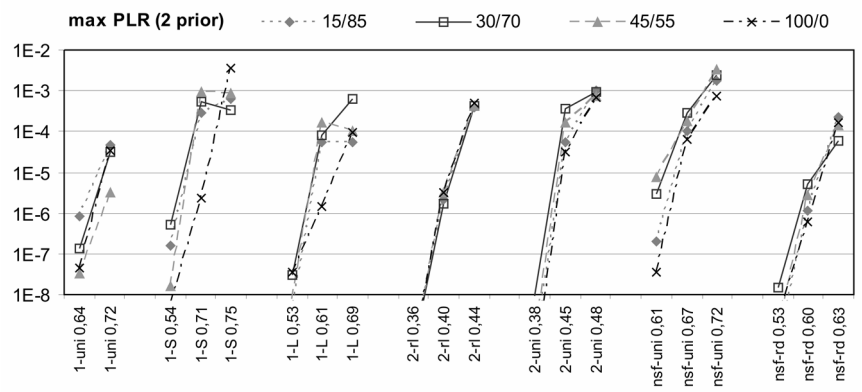


Figure H.18: Comparison of max-PLR values reached with the WorstLink variant of the PLR-based algorithm, when (i) considering traffic of a single priority (100/0 ratio highest/lowest priority ratio) and (ii) when considering different ratios of traffic of two priorities (15/85;30/70;45/55 ratio high 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).

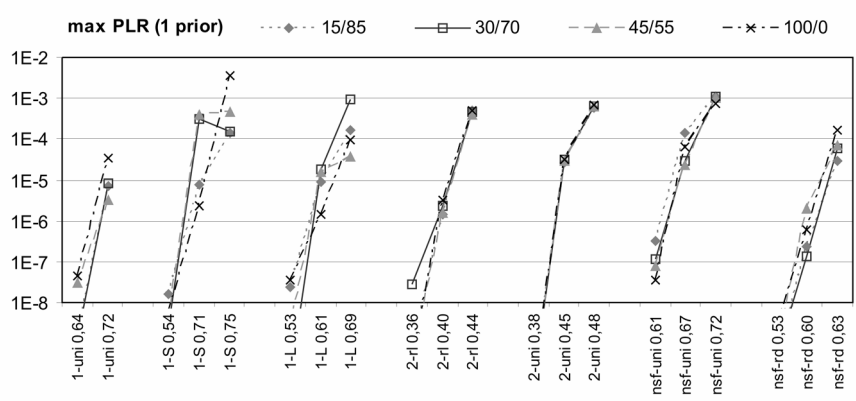


Figure H.19: Comparison of max-PLR values reached with the WorstLink variant of the PLR-based algorithm for traffic of one priority (i) for demands of 100% (100/0) and (ii) for the same demands split into two parts: 15/85; 30/70; 45/55.

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).

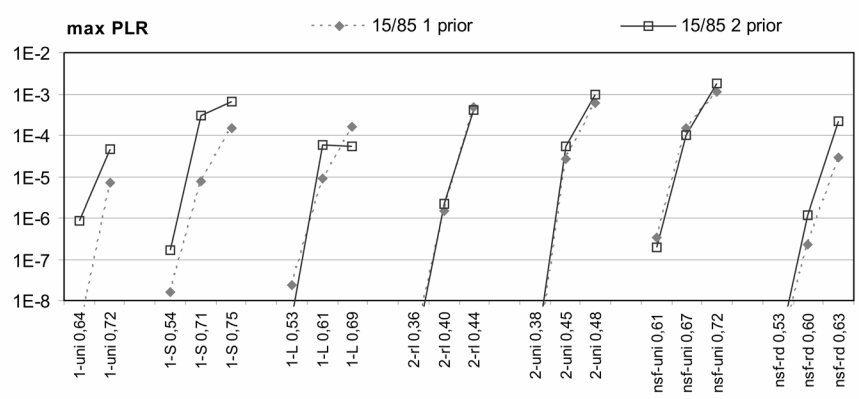


Figure H.20: Comparison of the max-PLR values reached for the traffic demand ratio 15/85 when the two parts the demands are split into are (i) of the same priority, (ii) of a different (high/low) priority.

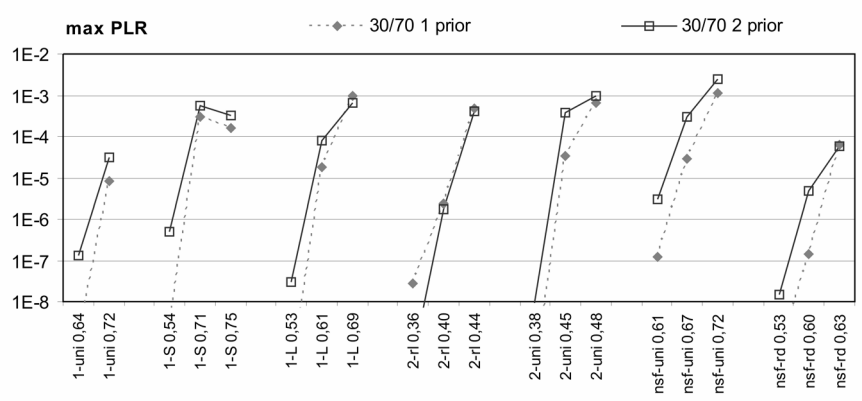


Figure H.21: Comparison of the max-PLR values reached for the traffic demand ratio 30/70 when the two parts the demands are split into are (i) of the same priority, (ii) of a different (high/low) priority.

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.

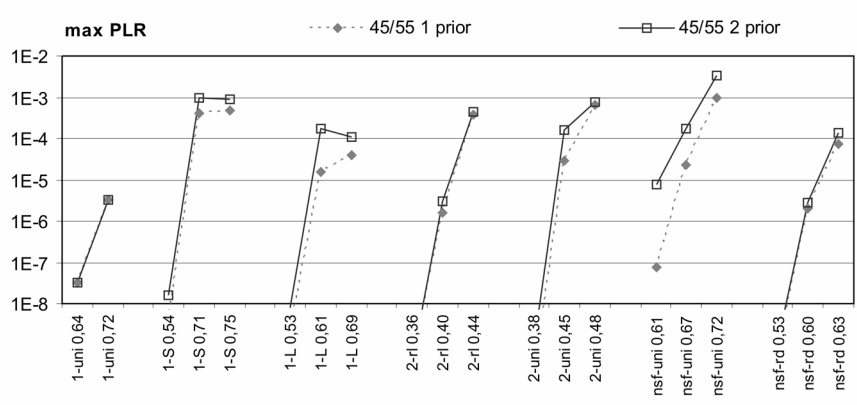


Figure H.22: Comparison of the max-PLR values reached for the traffic demand ratio 45/55 when the two parts the demands are split into are (i) of the same priority, (ii) of a different (high/low) priority.

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.

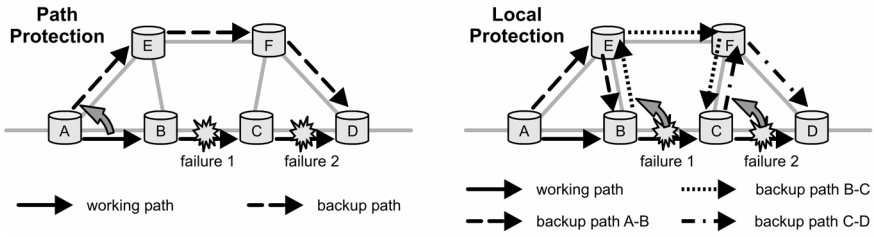


Figure I.1: Illustration of path and local protection, under two different failure scenarios.

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.

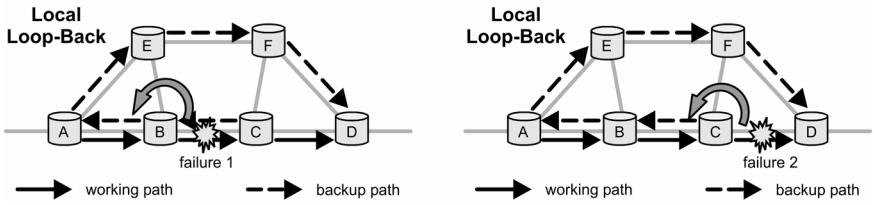


Figure I.2: The local loop-back technique combines the advantage of Path Protection (single backup LSP) and Local Protection (protection switch performed locally in an LSR adjacent to the failure).

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.

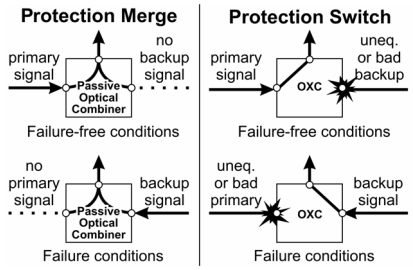


Figure I.3: A protection merge (left) can be realized by a passive optical combiner, if and only if backup and primary signals never are received simultaneously. If this condition cannot be met, a protection switch (right) is needed instead of a protection merge [13].

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.

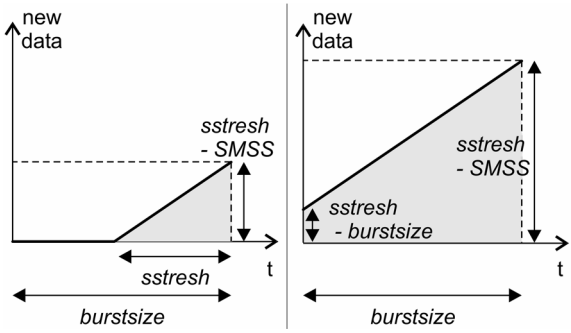


Figure I.4: Evolution of new data sent during successive RTTs in the Fast Retransmit/Fast Recovery phase of TCP NewReno. The left part illustrates the case where *burstsize* > *sstresh*, whereas the right part illustrates *burstsize* < *sstresh*. Note that the scales of the axes are not the same for 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).

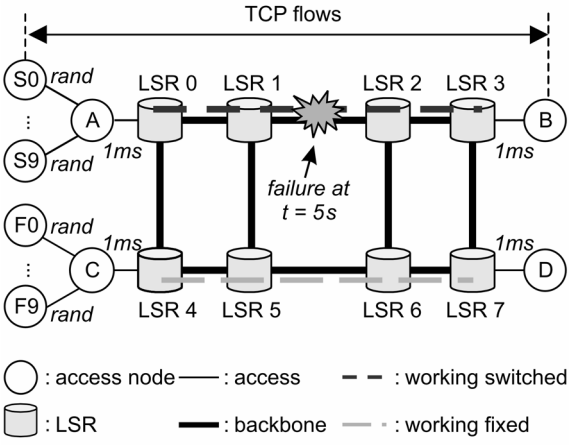


Figure I.5: Illustration of general set-up of our simulations. Each LSP carries an aggregate of 100 TCP flows, originating at nodes Sx that are connected to the backbone via access links, of which the first (to node A) has a randomly generated length (in terms of propagation delay). The TCP sources are 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.

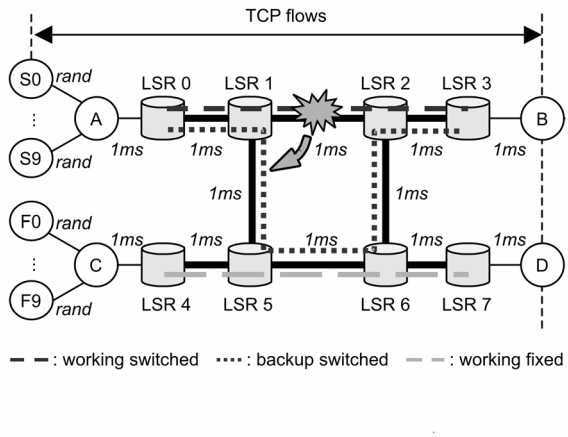


Figure I.6: Simulation set-up used to study the impact of protection switch speed on TCP.

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).

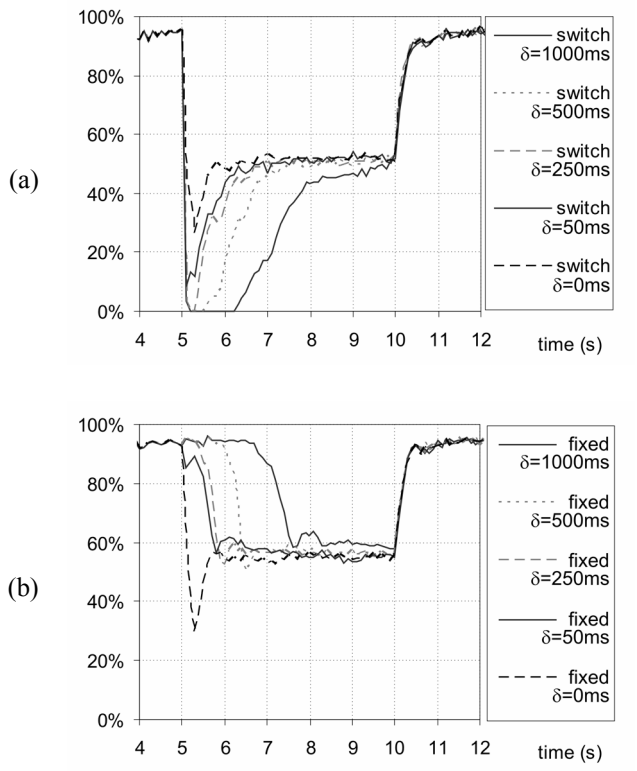


Figure I.7: TCP goodput evolution over time for different values of the switching time (delay) δ for the whole of the "switched" resp. the "fixed" flows. The goodput is expressed in % of backbone link bandwidth and was measured with a resolution of 10ms (i.e. T = 10ms in Equation(4)).

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%

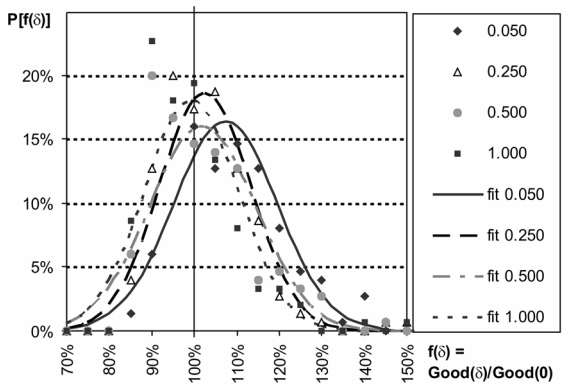


Figure I.8: Histograms (with a resolution of 5%) and normal fits for relative amount of goodput. A marker at (x,y) for a particular δ means that y% of the simulation results had f(δ) within [x,x+5%).

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}.

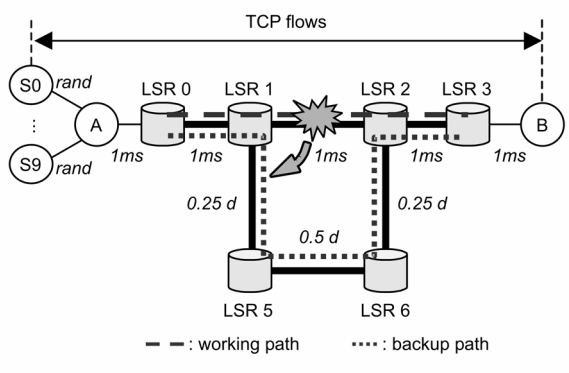


Figure I.9: Simulation topology used to investigate the influence of changes in RTT, caused by a protection switch, on TCP behavior. The access links had a bandwidth that was 80% of that of the backbone links.

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.

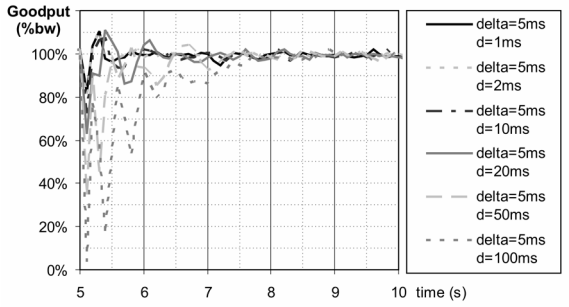


Figure I.10: Goodput evolution in interval (5s,10s] for δ = 5ms, and d in {1ms, 5ms, 10ms, 20ms, 50ms, 100ms}. It is plotted here with a resolution of 100ms, which means that every 100ms, the number of bytes successfully transported end-to-end in (t–100ms, t] was measured.

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.

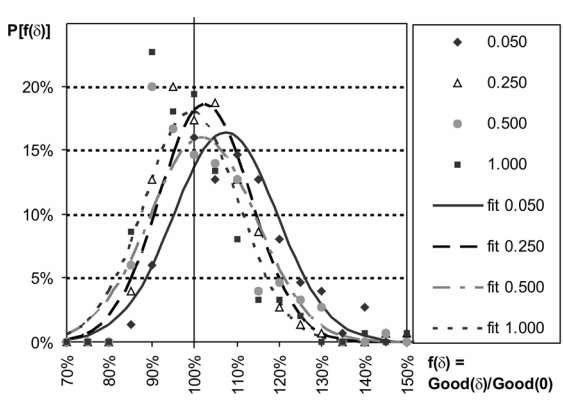


Figure I.11: Histograms (with a resolution of 1%) and normal fits for relative amount of goodput. A marker at (x,y) for a particular d means that y% of the simulation results had f(d) within [x,x+1%). Good(.) is the goodput attained by the total of all TCP flows in the period (5s, 6.5s].

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).

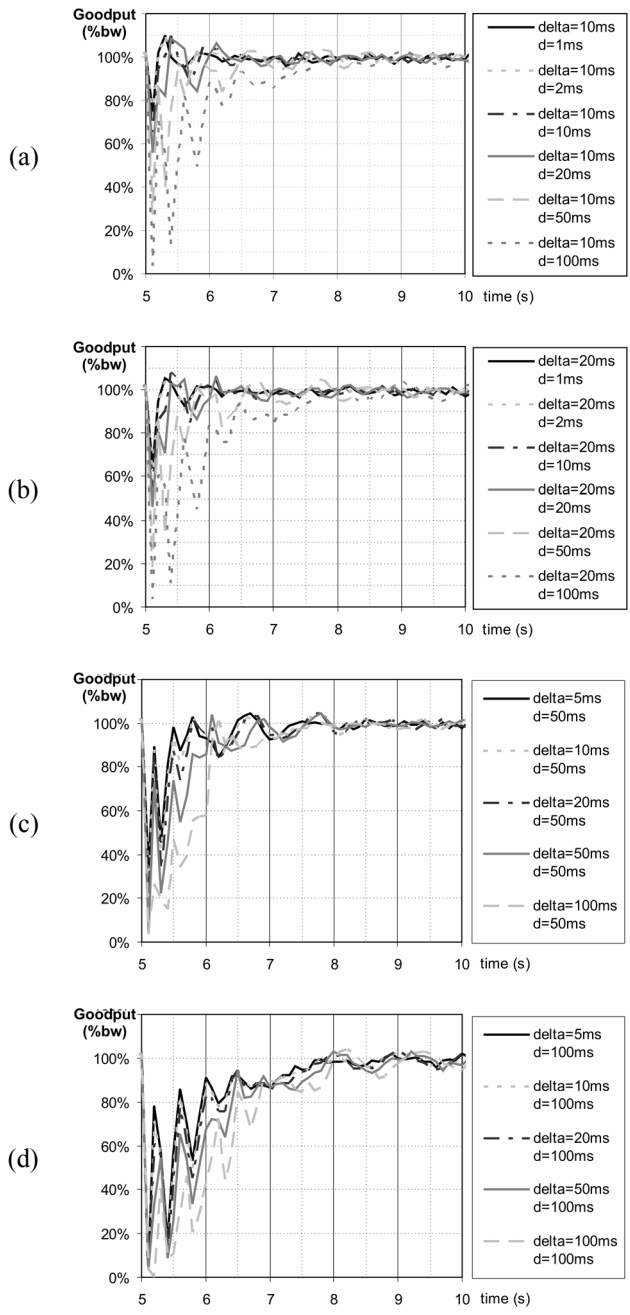


Figure I.12: Goodput evolution in interval [5s,10s] for all simulated combinations of (d,δ), grouped per δ.

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%

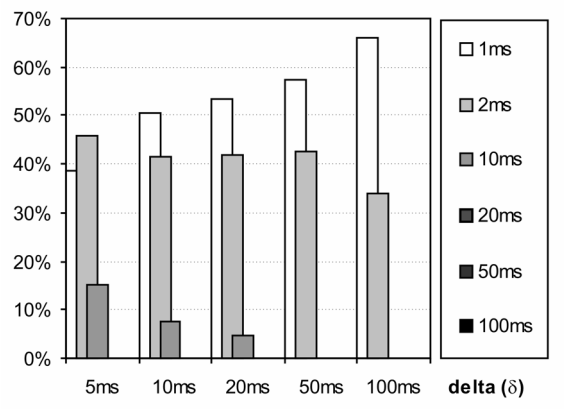


Figure I.13: Bar graph showing, for each δ, the percentage of simulation cases where a particular d was best. A bar with height y% for (d,δ) means that for y% of the simulations with δ, those for d resulted in the best value of Good(d,δ) (i.e. the goodput during (5s, 6.5s]).

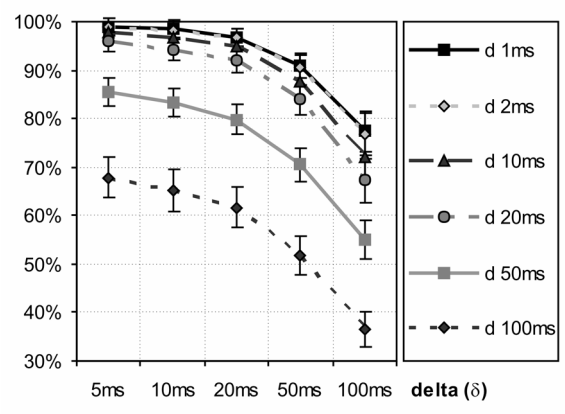


Figure I.14: Comparison of different values of d and δ. A marker with value y% means that averaged over 150 simulation runs, Good(d, δ) was y% of the maximal achievable goodput, i.e. that would be reached when every packet would be sent only once, and the access bandwidth would be fully used all the 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.

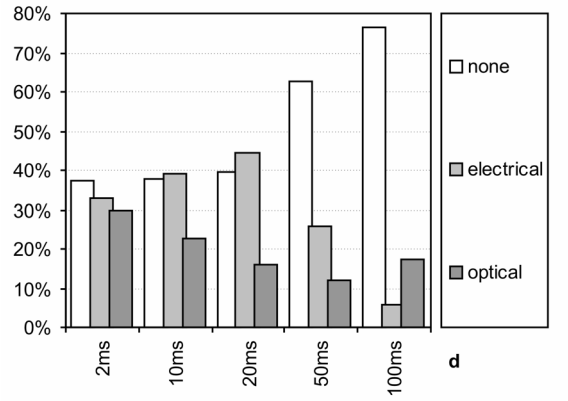


Figure I.15: Bar graph showing, for each d, the percentage of simulation cases where a particular strategy was best. E.g. a bar with height y% for "none" at d = 1ms means that for y% of the simulations with d = 1ms, "none" resulted in the best value of Good(.,d) (i.e. the goodput during (10s, 11.5s]).

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%

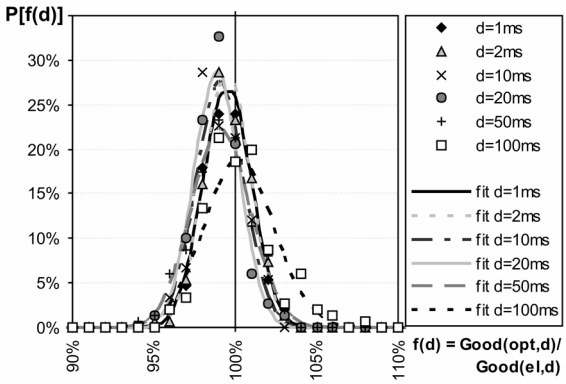


Figure I.16: Histograms (with a resolution of 1%) and normal fits for relative amount of goodput attained using "optical" strategy, compared to "electrical". A marker at (x,y) for a particular d means that y% of the simulation results had f(d) within [x,x+1%). Good(.) is the goodput attained by the total 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.

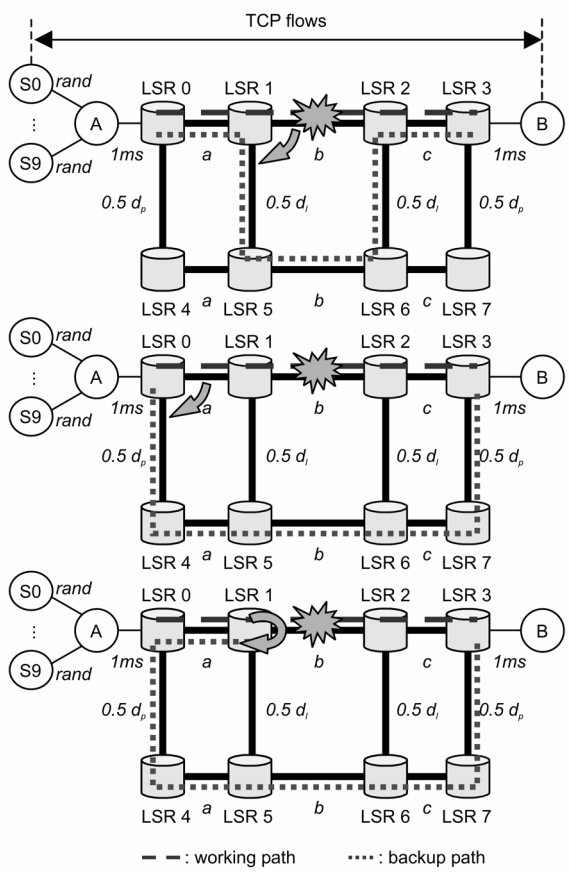


Figure I.17: Simulation topology and scenarios used to study the MPLS protection mechanisms; from top to bottom: local protection, path protection, local loop-back (for simplicity, only the path from LSR1 to LSR3 is shown, and not the reverse one). The access links had a bandwidth that was 80% of that 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.

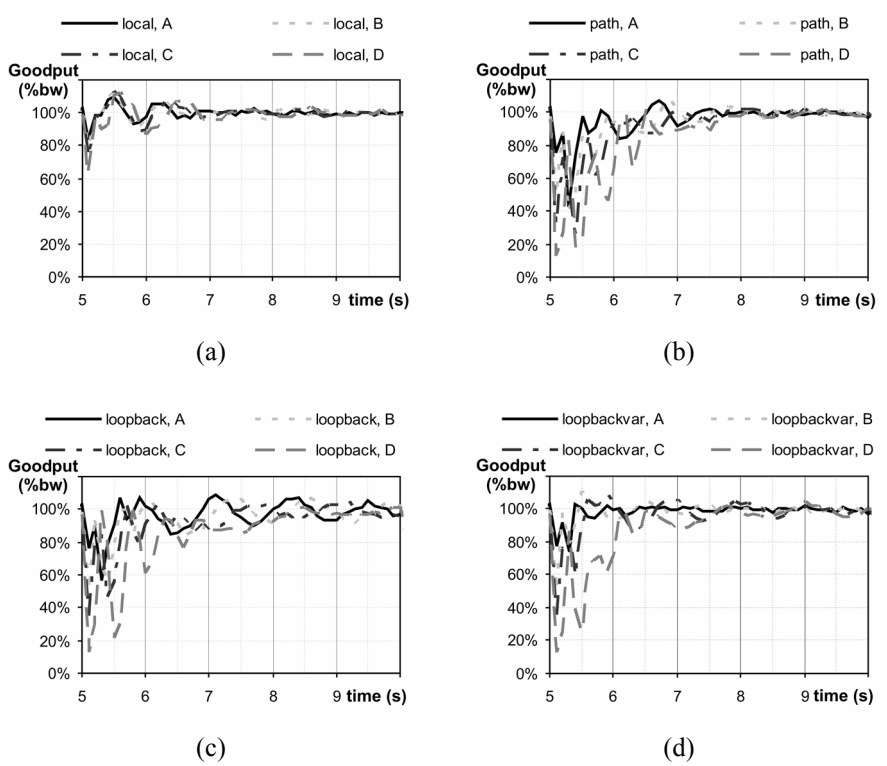


Figure I.18: Goodput evolution comparison for different protection mechanisms for each set of topology parameters (Table I.7), resulting in different protection switching time (Table I.8, second column). Goodput was plotted with a resolution of 100ms (meaning that value at t gives goodput in (t–100ms,t].

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.

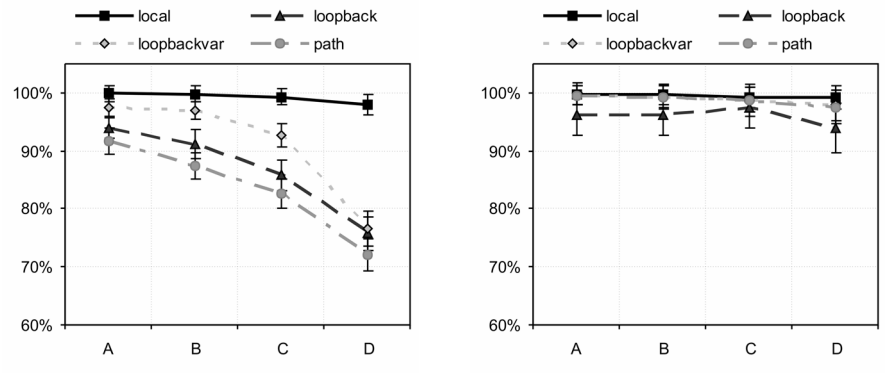


Figure I.19: Comparison of MPLS protection mechanisms. A marker for protection mechanism p and scenario s with a value of y% means that averaged over our 150 runs, Good(p,s) was y% of the maximal achievable goodput, i.e. the goodput that would be reached when every packet would be sent only once, and 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