

Impact of Assembly Algorithms on End-to-End Performance in Optical Burst Switched Networks with Different QoS Classes *

Maurizio Casoni, Enrica Luppi and Maria Luisa Merani

Department of Information Engineering

University of Modena and Reggio Emilia - Italy

Via Vignolese, 905 - 41100 - Modena - Italy

Tel.: +39-059-2056167; Fax.: +39-059-2056129

email: {casoni.maurizio, luppi.enrica, merani.marialuisa}@unimore.it

Abstract

This paper investigates the performance of a complete OBS (Optical Burst Switched) network that implements the JET (Just Enough Time) reservation mechanism. The network under examination adopts a core node architecture with no fiber delay lines and a limited set of wavelength converters, while featuring an edge node architecture with a mix of input, output and shared buffers. We investigate the overall network performance and design, specifically focusing on burst delay and end-to-end TCP throughput. In order to study the OBS network behavior, we develop a modular and flexible simulation tool, that we call MOBSSIM (Modular OBS Simulator). MOBSSIM is the means to accurately build an arbitrary topology OBS network via its main functional blocks, edge and core routers: its sound degree of development enables us to accurately evaluate several figures of merit, e.g., burst blocking probability and delay, and also allows for a critical comparison of alternative design solutions in terms of assembly algorithms, reservation strategies and QoS oriented routing.

1 Introduction

Optical networking has been boosted by the advances in coherent optical transmission, which led to DWDM (Dense Wavelength Division Multiplexing) systems able to accommodate hundreds of wavelengths per fiber, and by the advances in integrated optics, for passive and active optical components design. The ultimate goal is the development of a fully optical Internet, where signals carried within the network never leave the optical domain [1]. A first important step in this direction is to have optical networks transparent

at least for data, with the control part converted and processed in electronics. In the OBS solution [2, 3] data never leave the optical domain: for each data burst assembled at the network edge, a reservation request is sent as a separate control packet, well in advance, and processed within the electronic domain. The key idea behind OBS is to dynamically set up a wavelength path whenever a *large* data flow is identified and needs to traverse the network: a separate control packet, carrying relevant forwarding information, therefore precedes each burst by a basic offset time. This offset time is set to accommodate the non-zero electronic processing time of control packets inside the network. In an OBS network, nodes can be classified as either edge or core routers. The main task of edge nodes is the burst assembly function: as they represent the border between "traditional" electrical LAN/MAN IP networks and a high speed optical transport network, they must collect incoming IP datagrams and assemble them into bursts according to suitable algorithms. Core routers, on the other hand, deal with data bursts and the related control packets; they have to set-up on the fly internal optical paths for switching bursts and take them hop-by-hop closer to their final destination. The offset time allows the core switches to be bufferless, avoiding the employment of optical memories, i.e., fiber delay lines, on the contrary required by optical packet switching [4].

In this paper the focus is on the performance of a complete OBS network, where all its functional blocks have been implemented and integrated into the MOBSSIM tool, so that it is now possible to accurately evaluate the effects of all typical parameters (e.g., assembly time, offset time) on different choices of algorithms, such as assembly mechanisms, reservation, as well as routing strategies. Limited range wavelength converters [5] have been investigated in [6] and [7]. Our current contribution investigates the network behavior in the presence of core nodes equipped with a limited set of full range optical wavelength converters. The main metric figures we focus onto are the delays

*This work has been partially supported by MIUR, Italy, within the framework of the national project "INTREPIDO: Traffic Engineering and Protection for IP over DWDM".

introduced by the edge nodes and by the network. Finally, a widely-accepted TCP model is employed to investigate the end-to-end performance and bandwidth provided to the end users of the optical network.

The rest of the paper is organized as follows. In Section 2 the edge router architecture and the burst assembly strategies adopted are described. In Section 3 the core network with the related routing algorithms is shown. Section 4 provides the framework for the end-to-end performance evaluation when a transport protocol like TCP is employed. Numerical results are then collected in Section 5 while the concluding remarks are in Section 6.

2 Edge Routers and Burst Assembly

2.1 Traffic Profiles

This work assumes that the OBS nodes support JET and that traffic is mapped into three (but in general N_{QoScl}) QoS classes. Each class can be described by a different statistical traffic profile and can feature distinct QoS requirements, specifying a different upper bound on the burst blocking probability and/or edge-to-edge delay it can bear.

Traffic incoming into the edge nodes is supposed to be *M/Pareto*: equivalently, OFF periods are exponentially distributed, ON periods are Pareto distributed. Indicating by X_{on_i} the random variable which represents the duration of the ON period for the i -th input class, its cumulative distribution function is

$$F(x) = P[X_{on_i} \leq x] = 1 - \left(\frac{k_{on_i}}{x}\right)^{\alpha_{on_i}}, 0 < k_{on_i} \leq x, \quad (1)$$

where the constant k_{on_i} is the smallest possible value of the random variable X_{on_i} , expressed as multiple of a basic unit (here the size of the i -th class datagram); $0 < \alpha_{on_i} < 2$ in order to have a heavy-tailed distribution. Note that under the hypothesis $0 < \alpha_{on_i} < 2$, X_{on_i} has infinite variance.

Given that \bar{x}_1 , \bar{x}_2 and \bar{x}_3 are the mean duration times of the ON periods due to the aggregation of, for instance, short, medium and long-sized datagrams, respectively, on each incoming link the offered load ρ is the sum of three distinct contributions,

$$\rho = \sum_{i=1}^3 \rho_i = 3 \sum_{i=1}^3 p_i \lambda \bar{x}_i, \quad (2)$$

where p_1 , p_2 and p_3 represent the occurrence probabilities of the three classes and λ is the mean arrival rate.

Without loss of generality, bursts labelled class 1, class 2 and class 3, are defined: class 1 is created with time-sensitive data and with data that allow the management of closed-loop flow control, i.e., ACKs of end-to-end TCP flow

control. In the burst assembly process class 1 bursts are attributed the highest priority and thus given the longest extra-offset time. Class 2 and 3 bursts are supposed to be loss-sensitive and no extra-offset is given to them. It is worth noting that this last choice has simply been made to set a possible traffic scenario; MOBSSIM guarantees the a complete freedom in setting the extra-offset parameter.

An alternative traffic description, that MOBSSIM encompasses as well, although not considered in this paper, features medium and large-size datagrams that are 576 and 1500 byte long [8]; a third IP datagram size, equal to 200 bytes, is introduced to account for Voice over IP applications (20 bytes of IP header, 8 bytes of UDP header, 12 bytes of RTP header and 160 bytes of G.711 payload).

2.2 Edge Node Architecture

The edge node architecture investigated in this paper is reported in Figure 1. It can be logically divided into three main blocks. The first includes the input interface cards, equipped with buffers for storing datagrams that cannot be currently forwarded to the shared memory where bursts are generated. The second is the burst assembly block, given by a control unit operating as router and classifier and by a shared memory for burst formation. The third block is given by the output interfaces, provided with the scheduler for managing burst transmissions and with the output buffers for contention resolution when all wavelengths are busy. Incoming datagrams are grouped according to their destination (among N_{dest} possible) and class of service, being N_{QoScl} the number of such classes. Generated bursts may then be classified as belonging to one out of $N_{virt-queues} = N_{dest} \times N_{QoScl}$ queues. The electronic shared memory can therefore be divided in $N_{virt-queues}$ virtual queues and it is on them that the burst assembly algorithm operates.

Incoming datagrams may also be buffered into input electronic memories in case of multiple arrivals of packets directed towards the same virtual queue. The transit delay from input to shared memory is supposed negligible compared to assembly time. It is worth noting that this architecture impacts the delay each datagram experiences in the edge node and the outgoing burst length: these parameters will be investigated and discussed next.

2.3 Assembly Algorithms

Three distinct burst assembly algorithms are investigated. Namely, T_{max} [10], $T_{max-L_{min}}$ version 1 and $T_{max-L_{min}}$ version 2 that we newly propose. Observe that, depending on the nature of information flows, whether time-sensitive or loss-sensitive, two different goals are to be pursued. For the former, since the edge-to-edge delay has to be bounded, a maximum delay T_{max} can be tolerated in the assembly phase, but after that the burst must be transmitted. For the second type of flows, delay is not a tight

constraint: the main goal is to achieve the highest efficiency by transmitting bursts only after they have reached a minimum length L_{min} . This nevertheless implies to keep the burst length under control, avoiding the creation of bursts that are too long and may adversely impact network performance. For the $T_{max}-L_{min}v1$ algorithm, the condition on L_{min} for class 2 and 3 is enough to consider the burst ready to be scheduled for transmission. On the other hand, for $T_{max}-L_{min}v2$, all overlapping incoming datagrams directed to the same virtual queue are assembled into the same burst. To make clear the way these algorithms work, consider Figure 2, that shows an example of datagram pattern entering the edge node. If these datagrams are directed to the same virtual queue, the T_{max} algorithm assembles all four datagrams in the same burst. With the $T_{max}-L_{min}v1$ scheme, if $L_1 + L_2 > L_{min}$, datagrams 1 and 2 are assembled in a burst and datagrams 3 and 4 in the next. Last, with the $T_{max}-L_{min}v2$ strategy, datagrams 1, 2 and 3 go into a burst and only datagram 4 falls in the next. This is worth underlining, because in case of multiple arrivals towards the same output virtual queue, several bursts might be ready for transmission during the transmission time of the previous burst. Since wavelengths in output fibers are assigned with no regards to the class the bursts belong to, it might happen that “newborn” bursts are blocked because there is no free wavelength available for transmission. If this is the case, these bursts have to be further delayed, rather than lost, and must be stored somewhere. This is one of the tasks of the scheduler S , that takes the bursts from the shared electronic memory as soon as they are ready and sends them out to the electronic output queue associated to the selected optical output interface. As soon as a wavelength becomes available, the next burst is taken from the queue, converted in optics and transmitted.

In Section 5, several numerical comparisons among the burst assembly algorithms will be reported, together with some design remarks.

3 Routing and Network Topology

3.1 Core Node Architecture

The core optical router examined is equipped with $M \times M$ optical interfaces capable of supporting N wavelengths each. We suppose that the size of the pool of converters is limited and that the optical node is bufferless, i.e., no fiber delay lines (FDLs) are present in order to resolve contention for an output fiber, output wavelength. It is however reasonable to assume the use of a set of input FDLs [9], whose exclusive task is to re-align the OBS data burst and its control packet, so as to guarantee a minimal offset time at any intermediate node (Figure 3). Let w_c be the number of wavelength converters the node is equipped with. Whenever an incoming burst enters the node on the w_i wavelength

($i \in 1, 2, \dots, N$), an attempt is made to forward it to the desired output fiber on that same wavelength, w_i . If this is not available, the core router might look for a free output wavelength w_j , $j \neq i$, on that same output fiber, but has to resort to a converter, in order to successfully forward the burst. In order to enforce distinct QoS levels, in this study we only allow class 1 and class 2 bursts to resort to wavelength conversion.

3.2 QoS Routing and Network topology

Regarding the network, information flows are made of bursts routed by means of the Dijkstra algorithm; for the sake of simplicity, all links are supposed to be of equal cost. Since class 1 bursts carry time-sensitive data, class 2 and 3 loss-sensitive data, routing is modified in order to better meet the performance required by each class and a simple QoS routing criterion is introduced: class 2 and 3 bursts are allowed to be deflected and thus re-routed in case of unavailability of a wavelength in the desired output fiber. This leads to a variable edge-to-edge delay that needs to be evaluated; however, this solution is better than increasing the burst blocking probability these classes experience. As previously outlined, variable delays can be managed by the core nodes, because they employ input FDLs for offset realignment. Overall, the strategy for service differentiation is the following: class 1 bursts are given the highest priority through an additional extra-offset and the use of wavelength converters in the core nodes; class 2 bursts have medium priority by just using the converters; class 3 bursts have the lowest priority since they have no extra offset and cannot exploit wavelength conversion; on the other hand, class 2 and 3 can use alternative sub-optimal variable delay paths through deflection.

As far as the network topology is concerned, the network covers most European countries (Figure 4). London, Oslo and Stockholm nodes are sources of information flows and operate as edge routers; Madrid, Rome and Athens nodes are possible destinations. All the other nodes work as core routers. Moreover, no flow is supposed to enter or leave the network at intermediate nodes. Section 5 will show the performance of the three classes of bursts, in terms of overall edge-to-edge burst delay and burst blocking probability. These performance figures will also be discussed when the different assembly algorithms previously described are employed.

4 End-to-End Performance

Throughput is studied here on an end-to-end basis, when TCP is assumed as the transport layer protocol. In the system under investigation we assume that the hosts of the LANs/MANs connected to the OBS network employ TCP Reno. Following the model developed in [11], we approxi-

mate its throughput (in bit/s) as

$$Thr_{TCP} = \frac{MSS}{RTT\sqrt{\frac{2bp}{3}} + T_0\min\left(1, 3\sqrt{\frac{3bp}{8}}\right)p(1 + 32p^2)} \quad (3)$$

MSS being the maximum segment size expressed in bits, RTT the round trip time, p the segment loss probability, T_0 the retransmission timeout and b the number of packets acknowledged by every ACK.

The performance of TCP over OBS networks has been studied in some previous works,[12]-[13], but it can still be considered an open issue and thus a challenge for the research community. Here we want to add some thoughts to the discussion. We begin by observing that for the OBS network under study, where the core nodes are bufferless, the process of burst assembly/disassembly at edge nodes does have a great impact on end-to-end performance, as the delay it introduces must be carefully evaluated and taken into account. To this regard, if we indicate by $T_{assembly}$ and T_{disass} the assembly and disassembly times, by N_{hops} the number of traversed hops and by T_{hop} the propagation delay of each hop, we have that the one-way delay $T_{e2e1way}$, measured edge-to-edge, from the time the TCP segment enters the OBS ingress edge to the time it leaves the OBS egress edge, is

$$T_{e2e1way} = T_{assembly} + N_{hops} \times T_{hop} + T_{disass} \quad (4)$$

When working under the hypothesis that the network has a symmetric behavior, RTT in (3) can be determined as $RTT = 2T_{e2e1way}$. Actually, RTT should also include the delays given by the access networks that the TCP connection traverses: equivalently, $RTT = 2T_{e2e1way}$ can be interpreted as a lower bound. Now, considering the network topology reported in Fig.4, the average link length is 800 km and N_{hops} is in the [3 – 5] range. Since the light propagation speed in the fibers is roughly 70% the speed of the light in the vacuum, the propagation delay for each hop, T_{hop} , is on average 4 ms. It follows that $N_{hops} \times T_{hop}$ is upper bounded by 20 ms, while $T_{assembly}$ and T_{disass} remain to be estimated: their actual values depend on the specific assembly algorithm the edge nodes adopt.

Regarding p , the second parameter that needs to be properly determined in order to estimate the end-to-end TCP throughput, we assume, as classified in [12], to have only slow TCP sources that emit at most one segment during the interval $(0, T_{max})$. This implies that at most one segment for each connection is contained in a burst generated by the edge node and injected into the OBS network. It follows that for this type of sources the segment loss probability p can well be approximated by the burst blocking probability.

Next Section will report the TCP throughput results obtained for the above assumptions on RTT and p , and for the

different assembly algorithms we have examined.

5 Numerical Results

The presented results refer to edge nodes, where different assembly algorithms are compared, to the entire network, where the burst loss probability and the overall delay are the figures of merit, and to end-to-end performance experienced by a TCP connection traversing the OBS core network. The performance of the complete burst switching network has been determined by simulation, relying upon the MOBSSIM tool, an *ad-hoc* event-driven C++ object oriented simulator. In order to describe the burst duration, as well as the interarrival time between bursts, the simulator adopts an ON/OFF model with exponentially or Pareto distributed ON (burst duration) and OFF (interarrival time) periods. Incoming traffic into edge nodes is supposed to be $M/Pareto$. with $\bar{x}_1 = 218$ bytes, $\bar{x}_2 = \bar{x}_3 = 10$ kbytes. The extra offset for class 1 bursts is set at $8\mu s$. p_1 , p_2 and p_3 are set equal to 0.5, 0.2 and 0.3 respectively. Regarding core routers, various symmetric $M \times M$ OBS core nodes featuring a different number of incoming and outgoing fibers have been investigated where each of the outgoing fibers is chosen with equal probability $1/M$. It is set $wc = 32$, $M = 2$ with $N = 16$ wavelengths per fiber. T_{max} has been set equal to 1 ms for class 1 burst assembly and to 20 ms for class 2 and 3. L_{min} has been set to 640 kbytes. Figure 5 shows the average input queue delay of the three classes of service for the T_{max} - L_{min} version 1 and 2 algorithms. This delay has been computed only on datagrams which experience delay. The number of delayed datagrams increases when the offered load increases, but this average delay is not affected by it. T_{max} - L_{min} version 2 exhibits better performance than $v1$ for class 2 and 3 while are similar for class 1. Since all overlapping datagrams with $v2$ are assembled in the same burst, this implies a shorter time storage in the input queue than for $v1$, for which overlapping datagrams can be assembled into different adjacent bursts. Figure 6 reports the average assembly time for the T_{max} - L_{min} version 1 and 2 algorithms for the three classes of service. First of all, the general behavior is decreasing with load and this is expected: the more numerous datagrams arrive in a time period, the shorter the interval to reach L_{min} . While for class 1 bursts both algorithms give the same delay, bounded by the imposed 1 ms value, for the other two classes $v1$ performs better. This was expected as well, since building a burst with all overlapping datagrams takes longer than building with just the one which makes the burst be longer than L_{min} . Here we are quantifying in terms of delay the behavior as a function of load. Table 7 reports the average number of class 2 and 3 bursts ready for transmission per virtual queue as a function of the offered load. It is worth noting that during transmission of class 1 bursts no other burst of this class

comes to be completed and ready for transmission. It is also interesting to observe that when $T_{max}\text{-}L_{min}v1$ is employed, the average number is roughly twice the value obtained for the T_{max} algorithm: this is as expected, since the L_{min} condition implies shorter but many more bursts than T_{max} . When the $T_{max}\text{-}L_{min}v2$ is used the average number is less than $T_{max}\text{-}L_{min}v1$ and just greater than T_{max} . This average number plays a crucial role in system design: it represents the number of resources, i.e., wavelengths, per virtual queue to be installed in order to limit blocking during transmission. As previously said, $N = 16$ and in general it has to be properly dimensioned. Figure 8 and 9 show the output queue delay experienced by bursts for the different assembly schemes. T_{max} generates the longest bursts and several of them are on average ready for transmission and are then buffered. With $T_{max}\text{-}L_{min}v1$ shorter bursts are generated but there are many more ready for transmission: as a consequence, the output delay for class 2 and 3 are similar to T_{max} , but class 1 improves since they can interleave better with class 2 and 3. It is interesting to note that $T_{max}\text{-}L_{min}v2$ provides a good trade-off between burst length and average number of completed bursts: as a matter of fact, it gives the shortest output queue delay. Furthermore, class 1 output delay is always below 1 ms, while it can reach 90 ms for T_{max} and this value may be unacceptable for real time applications. Summarizing, an incoming datagram undergoes a cascade of delays given by input, shared and output buffers. Let us assume 0.6 as the reference load. If $T_{max}\text{-}L_{min}v1$ is employed, the edge router gives for class 3 a total delay of $10 + 0.5 + 250 \simeq 260$ ms, for class 2 of $8 + 1 + 250 \simeq 260$ ms and for class 1 of $0.1 + 1 + 2 \simeq 3$ ms. If, on the other hand, $T_{max}\text{-}L_{min}v2$ is used, the edge router gives for class 3 a total delay of $2 + 1 + 2 = 5$ ms, for class 2 of $1 + 1.2 + 2 \simeq 4$ ms and for class 1 of $0.1 + 1 + 1 \simeq 2$ ms. Recall that by using the T_{max} algorithm just the average delay in output queue is roughly 80 ms. In conclusion we believe that $T_{max}\text{-}L_{min}v2$ is the best choice. Figure 10 reports the burst loss probability as a function of load for the Stockholm-Rome couple when the $T_{max}\text{-}L_{min}v1$ and $T_{max}\text{-}L_{min}v2$ algorithm are used at edge nodes. The difference between the two algorithms is evident: the latter guarantees a remarkable improvement. More comments now about the curves of $T_{max}\text{-}L_{min}v2$. Interestingly, class 2 bursts never experience loss because they can be deflected in case of output contention. On the other hand, class 1 bursts may experience some loss because they may incur in congested nodes and cannot be deflected in order not to increase their edge-to-edge delay. As expected, class 3 bursts, having the lowest priority, experience the worst performance, since they cannot use the set of converters and no extra-offset is provided. However, these results show that a satisfying network design with service differentiation can be achieved by properly limiting the traffic load: if it is bounded by $\rho = 0.4$,

burst loss is negligible for class 2 bursts, 10^{-3} for class 1 and 10^{-2} for class 3.

Let us now discuss end-to-end performance. Figure 11 shows the throughput of TCP Reno given by (3) as a function of RTT for $T_0 = 1.0$ s, $b = 2$, $MSS = 1500$ bytes and $p = 10^{-2}$, a reasonable value for the burst loss probabilities observed in the OBS network. Putting together edge node delays and network propagation delay, the RTT for class 2 and 3 results in exceeding 300 ms when $T_{max}\text{-}L_{min}v1$ and T_{max} are used, whereas it drops to roughly 45 ms for $T_{max}\text{-}L_{min}v2$. This figure clearly states that, even if a very high speed core network is employed, the best throughput we can achieve is 1.5 Mbit/s when the $T_{max}\text{-}L_{min}v2$ algorithm is employed and it decreases to unacceptably low values when the alternative algorithms are used. In addition, for class 1 bursts, carrying time sensitive data, the T_{max} strategy gives an RTT ranging from 10 ms (@ a load $\rho = 0.2$), to 90 ms, a delay that may be not tolerable for this kind of applications. Again, $T_{max}\text{-}L_{min}v2$ is the best choice, the corresponding RTT being bounded by at most a couple of ms.

6 Conclusions

This work has investigated the overall performance of an OBS network, focusing on the burst assembly mechanisms at the edge nodes and on their influence on end-to-end performance. Three different burst assembly algorithms, namely T_{max} , $T_{max}\text{-}L_{min}v1$ and $T_{max}\text{-}L_{min}v2$, have been studied and evaluated, once integrated in a complete OBS network having a specific European topology. $T_{max}\text{-}L_{min}v2$ has revealed as the best choice, satisfying the requirements of all classes of service, when the main target is end-to-end performance.

It has further been demonstrated that a key design parameter is represented by the number of wavelengths the edge nodes employ, since this value is strictly related with the node functioning and the delay they introduce. To this regard, an important index is the product $N_{virt\text{-}queues}$ times the number of concurrent-bursts, that provides a meaningful indication for the number of wavelengths to be installed, in order to limit output contention. If over-provisioning of wavelengths is pursued, the edge nodes do not have too much an impact; if not, they play a fundamental role. A trade-off must anyhow be found, among the complexity/cost of the transmission system and its optical components and the type of burst assembly algorithm employed.

References

- [1] B. Bostica, F. Callegati, M. Casoni, C. Raffaelli, "Packet=Optical Networks for High Speed TCP-IP Backbones" - *IEEE Communications Magazine*, January 1999, pp.124-129.

- [2] C. Qiao, M. Yoo, "Optical Burst Switching (OBS) - a New Paradigm for an Optical Internet," *Journal of High Speed Networks*, No.8, pp.69-84, 1999.
- [3] M. Yoo, C. Qiao, S. Dixit, "QoS Performance of Optical Burst Switching in IP-Over-WDM Networks," *IEEE Journal on Selected Areas in Communications*, Vol.18, No.10, pp.2062-2071, October 2000.
- [4] M. Casoni, C. Raffaelli, "Tandem Architecture for Photonic Packet Switches" =96 *Journal of Communications and Networks*, Vol.1, No. 3, September 1999, pp.145-152.
- [5] B.Ramamurthy, B. Mukherjee, "Wavelength Conversion in WDM Networking", *IEEE Journal on Selected Areas in Communications*, Vol.16, No.7, September 1998, pp.1061-1073.
- [6] T.Y. Chai, T. Cheng, C. Lu, G. Shen, S.K. Bose, "On the Analysis of Optical Cross-Connects with Limited Wavelength Conversion Capability", Proc. of IEEE ICC 2001, Helsinki (Finland).
- [7] Z. Zhang, Y. Yang, "Performance Modeling of Bufferless WDM Packet Switching Networks with Wavelength Conversion", Proc. of IEEE Globecom 2003, San Francisco (U.S.A.).
- [8] K. Thompson, G. J. Miller, R. Wilder, "Wide-Area Internet Traffic Patterns and Characteristics," *IEEE Network*, November/December 1997, pp.10-23.
- [9] Y. Xiong, M. Vandenhoute, H.C. Cankaya, "Control Architecture in Optical Burst-Switched WDM Networks", *IEEE Journal on Selected Areas in Communications*, Vol.18, No.10, October 2000, pp.1838-1851.
- [10] A. GE, F.Callegati, L.S. Tamil, "On Optical Burst Switching and Self-similar Traffic", *IEEE Communications Letters*, Vol. 4, No. 3, March 2000, pp.98-100.
- [11] J. Padhye, V. Firoiu, D. F. Towsley, J. F. Kurose, "Modeling TCP Reno Performance: A Simple Model and its Empirical Validation", *IEEE/ACM Trans. on Networking*, vol.8, no.2, pp.133-145, April 2000.
- [12] A. Detti, M. Listanti, "Impact of Segment Aggregation on TCP Reno Flows in Optical Burst Switching Networks", Proc. of IEEE Infocom 2002, 23-27 June 2002, New York (U.S.A.).
- [13] S. Gowda, R. K.Shenai, K. M.Sivalingam, H.C. Cankaya, "Performance Evaluation of TCP over Optical Burst-Switched (OBS) WDM Networks", Proc. of ICC 2003, 11-15 May 2003, Anchorage (U.S.A.).

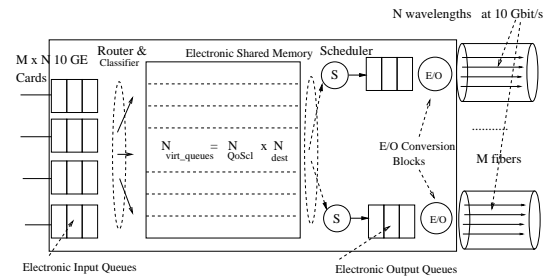


Figure 1: Edge node architecture.

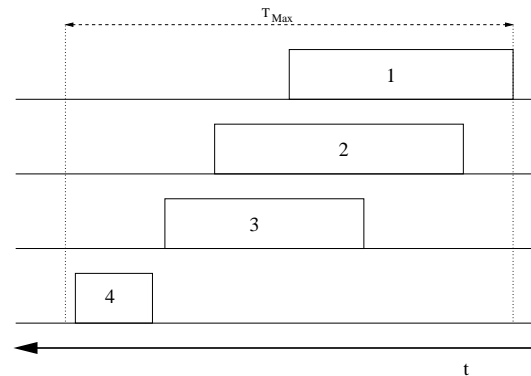


Figure 2: An example of datagram arrival pattern to the edge node.

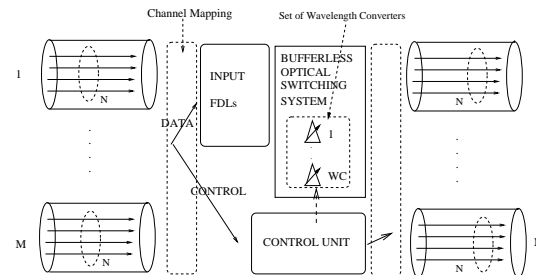


Figure 3: Core router architecture.

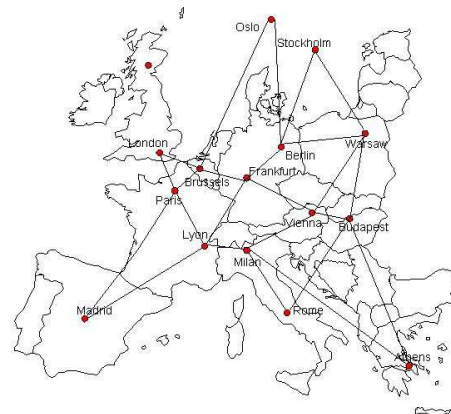


Figure 4: The reference network.

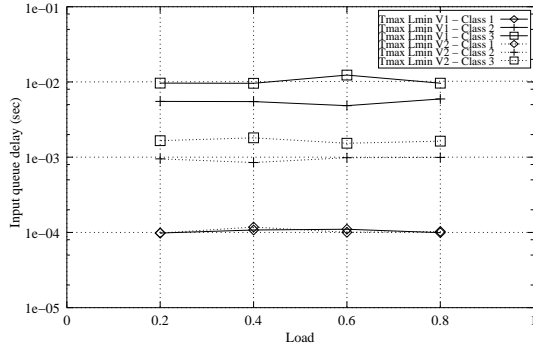


Figure 5: Average input queue delay for the T_{max} - L_{min} version 1 and 2 algorithms.

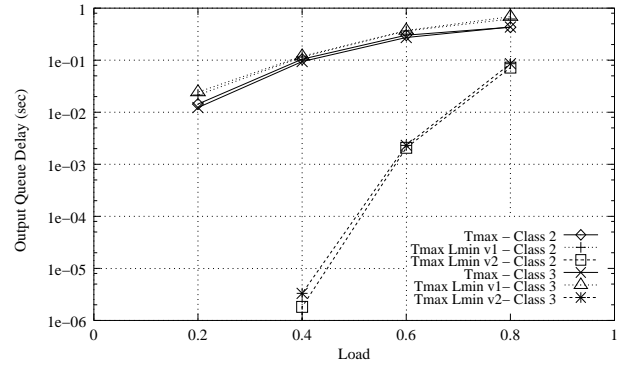


Figure 9: Average output queue delay for class 2 and 3 bursts.

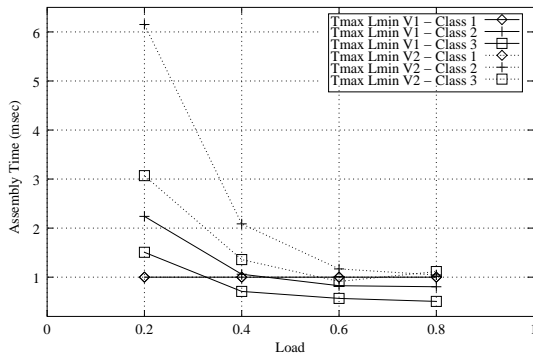


Figure 6: Average assembly time for the T_{max} - L_{min} version 1 and 2 algorithms.

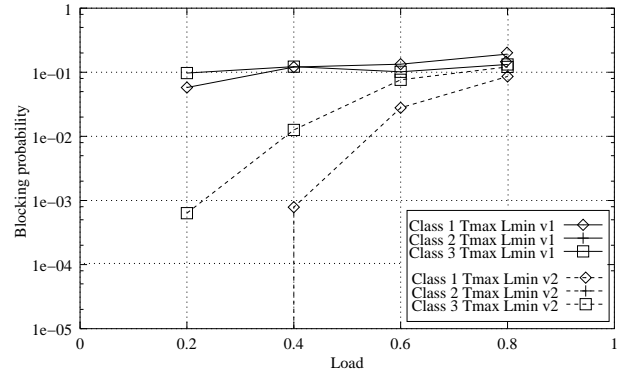


Figure 10: Average burst blocking probability for the couple Stockholm-Rome for the three classes of service and for two different burst assembly algorithms.

		0.2	0.4	0.6	0.8
Class 2	Tmax	1.73	2.17	2.27	2.55
	Tmax Lmin v1	2.98	2.97	4.94	4.98
	Tmax Lmin v2	1.56	2.20	2.69	3.68
Class 3	Tmax	2.61	3.29	3.34	3.66
	Tmax Lmin v1	4.84	6.08	6.70	6.99
	Tmax Lmin v2	1.93	2.64	3.57	5.51

Figure 7: Average number of concurrent bursts ready for transmission for the different assembly algorithms.

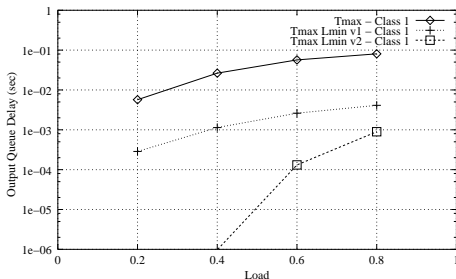


Figure 8: Average output queue delay for class 1 bursts.

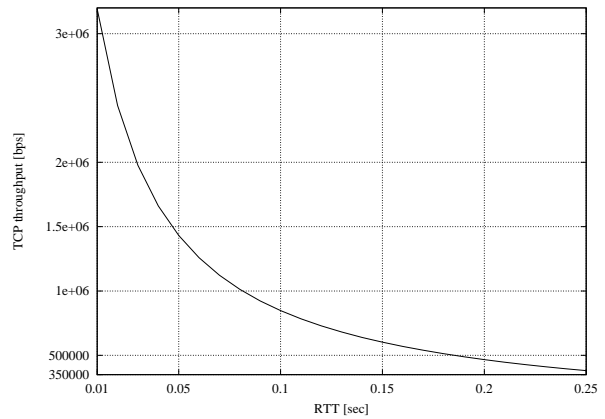


Figure 11: TCP throughput as a function of RTT with $p = 10^{-2}$, $b = 2$.