Chapter 4

Multi-PON architecture

4.1 State-of-the-art

The network concept (Figure 4.1) is based on a star topology presented in [24], which has been adapted to a metro environment with a target throughput of 1 Terabit/s per second. An AWG Hub interconnects several nodes through Passive Optical Networks (PONs). Each PON has a number of nodes attached to it that may be routers, access networks' head-ends, gateways to/from core networks, or any other kind of network node with a proper interface to this optical packet network.

The switching functions (time switching and lambda switching) are distributed between the nodes and the AWG. The nodes have electronic buffers, where electrical packets are stored and aggregated into longer packets before entering the optical domain, and rapidly tuneable transceivers (Tx/Rx). At a bit rate of 10 Gbit/s the tuning speed should be a few nanoseconds. Nodes buffer the incoming packets electronically. For every packet, the node sends a request to the Network Controller (NC). The NC is hence responsible of allocating the network resources scheduling the requests. In this way, the network can be thought of as a single, big, distributed optical packet router where the complexity sharing with the nodes allows the most use of edge buffers and Tx/Rxs. The central node of the star network uses an AWG; an $N \times N$ multiplexer based on a waveguide grating providing static space routing dependent on the input (port, wavelength)-tuple and which offers frequency periodicity. The same pool of wavelengths is available to each PON in the network. The AWG has Wavelength Converter (WC) arrays, positioned between extra dummy ports, which, in combination with the tuneable Tx/Rxs in the nodes, in effect can provide a fully non-blocking, distributed 3-stage switch fabric between nodes if required. Each WC array has the same number of converters as the product of the number of PONs and the number of grating frequency periods used. The number of WC arrays required depends upon the traffic level and pattern.

4.1.1 MAC protocol

The network employs resource sharing based on WDMA/TDMA, where the time is further organized in frames, each frame containing F slots, as illustrated in Figure 4.2.

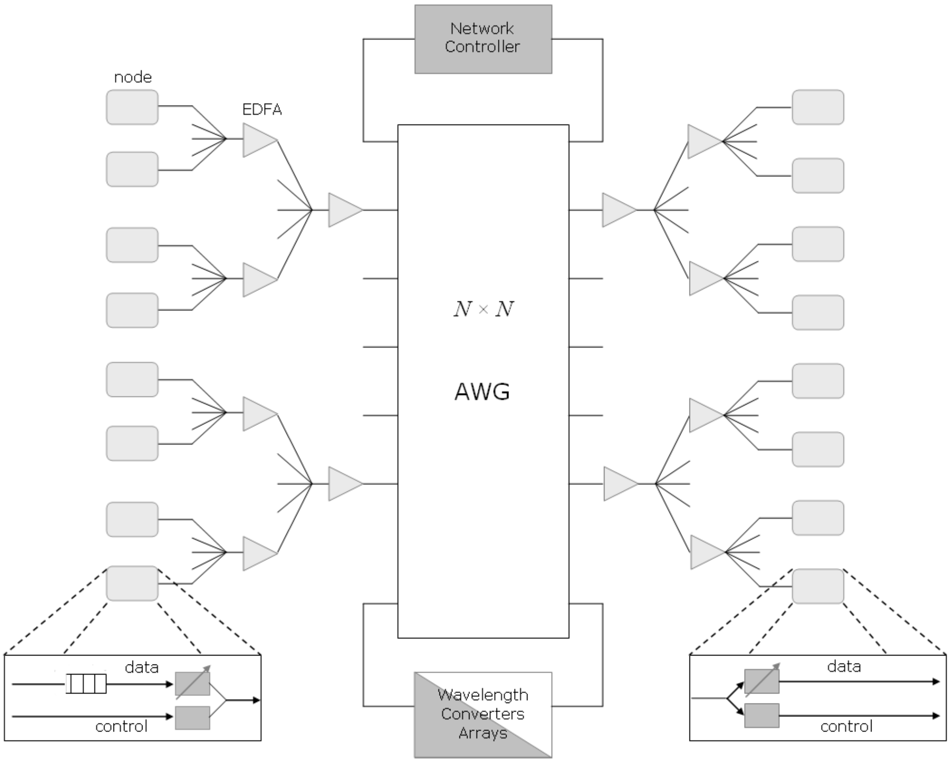

Figure 4.1: Multi-PON architecture

Slots run in parallel and are synchronized among all wavelengths in the same PON, so that a *multi-slot* (a slot per each wavelength plus the control slot) is available at each node in each time slot. The control slot is further partitioned into M *mini-slots*.

As a consequence of this scheme, the nodes see the network as a set of dynamic TDMA sub-networks (one per wavelength) in a way that a MAC protocol is required to avoid transmission collisions (multiple transmission in the same time slot) and receiver contention (more than one node trying to transmit to the same receiver at the same time). The MAC protocol is based on a bandwidth demand scheme. A node wanting to transmit sends a request (using a mini-slot) to the NC indicating the address of the destination node and the amount of slots required to accommodate the packet. A static TDMA protocol is used in the control channel, i.e., the node i has m_i reserved set of mini-slots available in each frame for allocating its requests (as shown in Figure 4.2). Therefore, the NC schedules the requests end-to-end between nodes. When a request has been successfully scheduled, the NC advises the node of the time slot and wavelength channel it has allocated to the packet. The following section discusses this issue.

4.1.2 Scheduling algorithms

The high targeted throughput of this network and the large number of nodes make impractical the implementation of optimum algorithms; it is proved in [6] that the

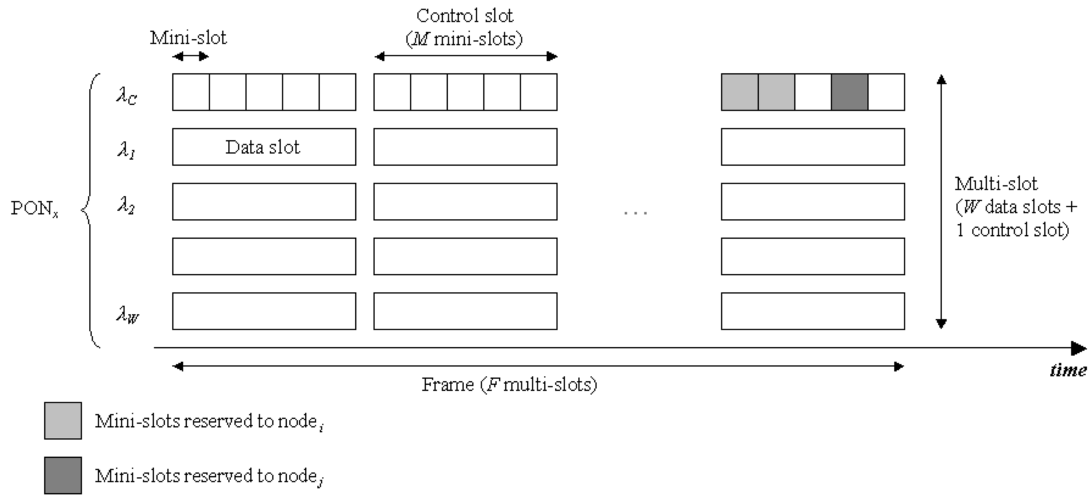

Figure 4.2: Timing structure of the wavelength channels

scheduling problem is NP-hard. Among all possible solutions, in DAVID project two heuristic scheduling algorithms have been considered:

- *Greedy algorithm* [6].
- *Frame-based algorithm* [31].

The Greedy algorithm is based on the separation of the time domain from the wavelength domain using two different algorithms:

- The *Slot Allocation* (SA) algorithm that operates always on a given fixed wavelength topology configuration based on a set of matrices, which provide a map of the network's available resources. According to the contents of these matrices, the algorithm selects the requests to be accepted based on a priority rotation scheme.
- A *Logical Topology Design* (LTD) algorithm that aims to obtain an efficient allocation of available wavelengths in order to reduce the slot allocation failure rate.

The Frame-based algorithm requires two steps for each frame: F-matching and Time Slot Assignment (TSA). The F-matching problem consists in finding the maximum subset of admissible packets that can be scheduled in a frame of length F slots. The TSA problem consists of scheduling through the switch the set of accepted non-conflicting slots in the frame and allocating wavelength channels to them.

For an $N \times N$ switch, the F-matching algorithm selects, at the end of the current frame, a set of up to $N \times F$ non-conflicting packets that will be transmitted in the slots belonging to the next frame. The matching algorithm accepts the set of non-conflicting packets always satisfying the following two non-overbooking properties:

- the number of accepted packets from each input port cannot be higher than F,
- the number of accepted packets to each output port cannot be higher than F.

According to these constraints, the matching algorithm runs through the well-known three phases [77] adapted to the frame length [8]:

1. *request*: each queue requests a number of slots from the corresponding appropriate output port in the frame,
2. *grant*: each output port issues up to F grants distributed amongst the queues destined for that output,
3. *accept*: each input port accepts up to F of the grants received at the port, where each acceptance received by a queue gives the right to use one slot in the next frame.

The selection of requests to be granted, and of grants to be accepted is based upon a rotating priority scheme, which is implemented using two sets of pointers, one for each input and one for each output. Several pointer use/update rules were defined in [7] that led to the definitions of different variants of the frame-based matching algorithm. Since the adoption of any variant of the frame-based matching algorithm does not affect the techniques we propose, we use the rules called NOB8, where the input (output) ports move their pointer to the output (input) port following the last one to which it gave an acceptance (grant).

For what regards scheduling a set of non-conflicting requests in a time frame (TSA problem), it is possible to see that the problem is equivalent to the routing of circuits in a Clos interconnection network, for which several algorithms are available in the literature; some of these algorithms are suited for parallel implementations (see, e.g., [72]), leading to complexities which can be smaller than the complexity of a typical sub-optimal F-matching algorithm. In this study we used the well-known sequential TSA algorithm proposed in [63].

4.1.3 QoS provisioning

Two different classes of services are considered in previous works: a *persistent* and *non-persistent* service [6]. The former is a connection-oriented approach which provides isochronous service: a node requiring it sends a persistent request to the NC indicating the amount of slots required in each frame (connection setup). Therefore, resources are allocated until one of the two involved terminals signals that it wants either release some slots for the connection or close it. The latter is based on a best-effort approach which provides asynchronous service; the nodes send a nonpersistent request and the resources are allocated only in a single frame; in the next frame, the NC will release all the resource previously allocated to best-effort traffic.

4.2 Contributions

Our contributions include: (i) the performance evaluation of these architectures and the identification of the weaknesses and of the open issues, (ii) the optimization of the proposed architectures and MAC protocols, and finally (iii) the proposal of a QoS mechanism to support priority and best-effort services.

4.2.1 Simulation scenario

The performances of the proposed mechanisms are evaluated in order to assess their merits. The simulation results presented in the following sections have been obtained by means of an ad-hoc event-driven simulator reproducing a real scale configuration of the multi-PON network. The parameters of the network are:

- P indicates the number of PONs;
- n indicates the number of nodes per PON;
- W indicates the number of wavelengths per fiber;
- WCA indicates the number of wavelength converter arrays;
- B_w indicates the bit-rate. It is set to 10 Gbit/s in every simulation scenario;
- P_s indicates the duration of the time-slot. It is set to 1 μs;
- F indicates the number of time-slots per frame;
- m_i indicates the number of mini-slots per control slot. It is set to 15;
- L indicates the distance between the Hub and the nodes. It is set to 20 km;
- ρ indicates the offered load;
- $\mathbf{M}$ indicates the PON-to-PON traffic matrix, whose generic element $\mathbf{M}_{i,j}$ is a real number ranging between 0 and 1 representing the percentage of traffic coming from input PON i and going to output PON j with respect to ρ. Four different traffic matrix are defined named: *uniform* $\mathbf{M}^U$, *power-of-two* $\mathbf{M}^P$, *diagonal* $\mathbf{M}^D$, and *dynamic diagonal* $\mathbf{M}^{DD}$. For the case of $P = 4$, the matrices are as follows:

$$M^U = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \qquad M^P = \frac{1}{15}\begin{bmatrix} 1 & 2 & 4 & 8 \\ 2 & 4 & 8 & 1 \\ 4 & 8 & 1 & 2 \\ 8 & 1 & 2 & 4 \end{bmatrix}$$

$$M^D = \begin{bmatrix} 0.55 & 0.15 & 0.15 & 0.15 \\ 0.15 & 0.55 & 0.15 & 0.15 \\ 0.15 & 0.15 & 0.55 & 0.15 \\ 0.15 & 0.15 & 0.15 & 0.55 \end{bmatrix} \quad M^{DD} = \begin{bmatrix} 0.4 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.4 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.4 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.4 \end{bmatrix}$$

For the dynamic diagonal traffic matrix the values on the columns shift on the right each 200 ms. This means that if the figure shows the situation at time 0, after, for example, 200 ms the new values $\mathbf{M}^{DD}_{i,j}$ are equal to $\mathbf{M}^{DD}_{i,|j-1|_4}$ of the figure.

These matrices represent a good sample of all possible traffic patterns: the classical uniform matric to evaluate a fair situation, the power-of-two matrix which demonstrates performance degradations when applied to the scheduling, the diagonal matrix to evaluate the inter-PON unbalanced situation, and finally a dynamic matrix to evaluate the behavior under fluctuations.

The electrical queues at nodes are considered infinite.

Concerning the traffic model, the packet interarrival time for any traffic class follows a self-similar process implemented as a superposition of 16 strictly alternating independent and identically distributed ON/OFF sources. The duration of each ON/OFF period was assumed to be a random variable with a Pareto distribution with shape $\alpha = 1.2$, which leads to a Hurst parameter of $H = 0.9$ [103]. All packets have the same size and fit in one slot.

Note that we assume an IP self-similar traffic model, since it is reasonable to think that IP will be the predominant traffic in metropolitan networks.

The number of simulated packets is chosen big enough to reach steady-state results and a 95% confidence interval is calculated.

We define the following measures to evaluate the performance of the metro network:

- *Throughput.* It is the usual performance measure and is calculated as the ratio between used and available slots.

- *Average and Maximum end-to-end delay.* They are, respectively, the average and maximum time needed to transmit a packet between a pair of nodes including both the service and transmission time.

- *Average Packet Loss Rate* (PLR). Assuming infinite buffer lengths, a packet is lost when the NC fails to serve the corresponding request within the QoS constraints. It is calculated as the ratio between the lost and generated packets.

In the following evaluation sections, we will only show the most significant measures according to the purposes of the study.

4.3 Performance evaluation

For this architecture, we have firstly performed an exhaustive performance evaluation in order to determine possible drawbacks of both the architecture and MAC protocol.

As an example, we show the results of the multi-PON network using the Greedy algorithms and considering $P = 4$ PONs, $n = 250$ nodes, $W = 32$ wavelengths, $WCA = 17$, and $F = 100$ slots per frame.

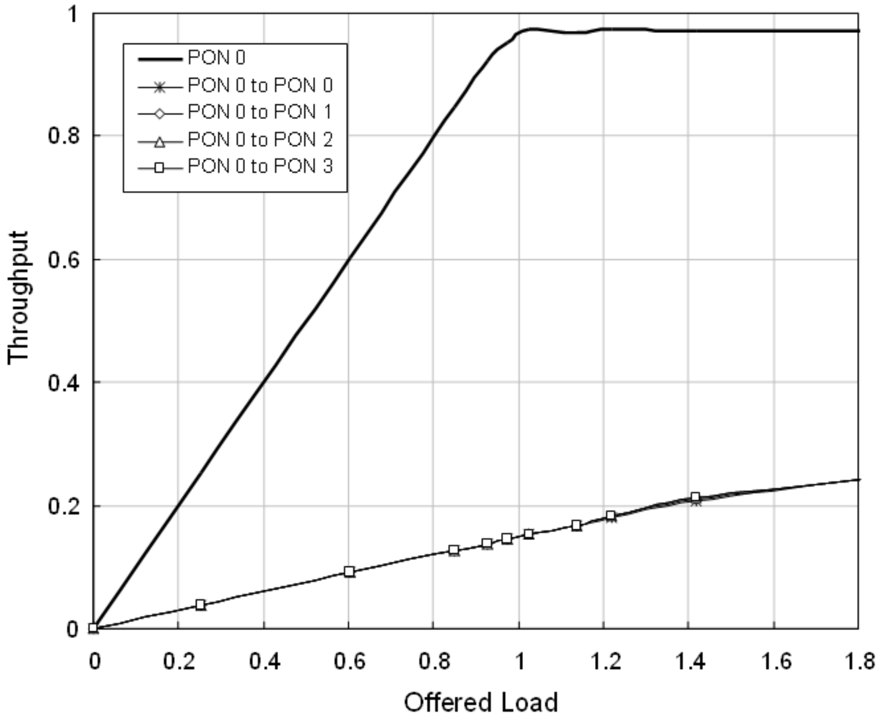

Figure 4.3: Throughput as a function of the offered load under uniform traffic matrix

Figure 4.3 shows the throughput per each destination PON reachable from PON 0 as a function of the offered load using the uniform traffic matrix. Although we report the throughput for a single PON, the same behavior holds for all other PONs due to the traffic symmetries. We can observe that the throughput increases with the offered load until it reaches the saturation. The PONs fairly use the resources as the overlapped curves indicates.

In Figure 4.4, we report the throughput per each destination PON reachable from PON 0 as a function of offered load under diagonal traffic matrix. From Figure 4.4, we can see that some degree of fairness between competitive traffic is enforced. For low value of offered load (ranging from 0.1 to 0.95), the throughput of each PON grows proportionally to their level. As soon as the offered load increases to values that create congestion, the network treats the PON-to-PON connections fairly according to a max-min like fairness criteria; i.e. the amount of admitted PON 0-to-PON 0 traffic is decreased to equalize the amount of traffic transmitted from each PON.

Next experiment is set up to evaluate the capacity of the network to cope with the rapid traffic fluctuations using the dynamic diagonal traffic matrix.

In Figure 4.5a, we examine the throughput from PON-0 to PON-0, and from PON-0 to each of the others possible destinations (PON-1, PON-2 and PON-3), as a function of time under dynamic unbalanced traffic near the saturation point ($\rho = 0.92$).

Figure 4.5a shows that the throughput per destination PON is always close to optimal one. That is, when the load of PON i-to-PON j interconnection is $\mathbf{M}_{i,j}^{DD} = 0.4$, the throughput is $0.4 \times \rho$; also when the $\mathbf{M}_{i,j}^{DD} = 0.2$, the throughput is $0.2 \times$

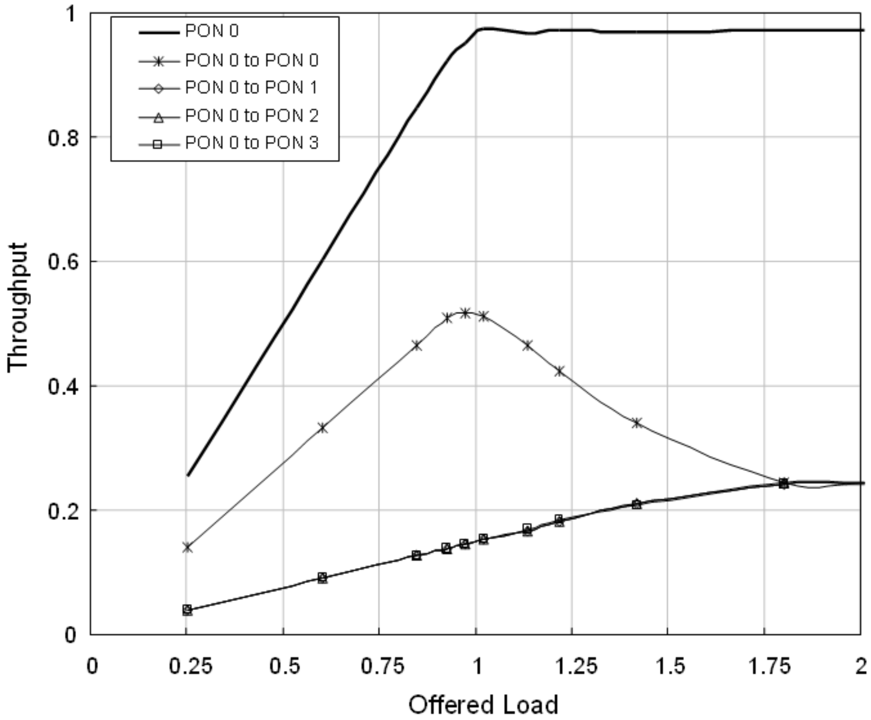

Figure 4.4: Throughput as a function of the offered load under diagonal traffic matrix

ρ. Moreover, Figure 4.5a shows that the throughput per destination PON reacts very quickly to the traffic fluctuations. To confirm this point, Figure 4.5b reports an enlargement around $200\,ms$ where a fluctuation occurs; with this figure, we can evaluate a transient behavior of less than $5\,ms$.

In next figures we compare the performance of the Greedy algorithm and the Frame-based algorithm considering a network with $P = 4$ PONs, $n = 32$ nodes, and $W = 32$ wavelengths. The power-of-two traffic matrix is used for this study.

The comparison is made in terms of throughput (Figure 4.6), and average delay (Figure 4.7). All traffic is treated as Best-Effort and packets are discarded only when requests fail to be served regardless of the buffer backlog, i.e. there are zero losses within the network and no packets are dropped within the nodes because of buffer overflow as lengths are assumed infinite. The frame length that maximizes the matching performance is the mean burst length times the number of switch ports [8]. As this figure can be very large sometimes, the chosen frame length is usually a compromise between throughput performance and maximum delay. For this study we have chosen a frame length of $F = 100$ slots, which provides a good performance with a reasonable delay below $1ms$ (1000 slots) for traffic loads under 80% (Figure 4.7). Figure 4.6 shows that the Frame-based algorithm yields a better performance even with just one WC array, reaching 90% throughput.

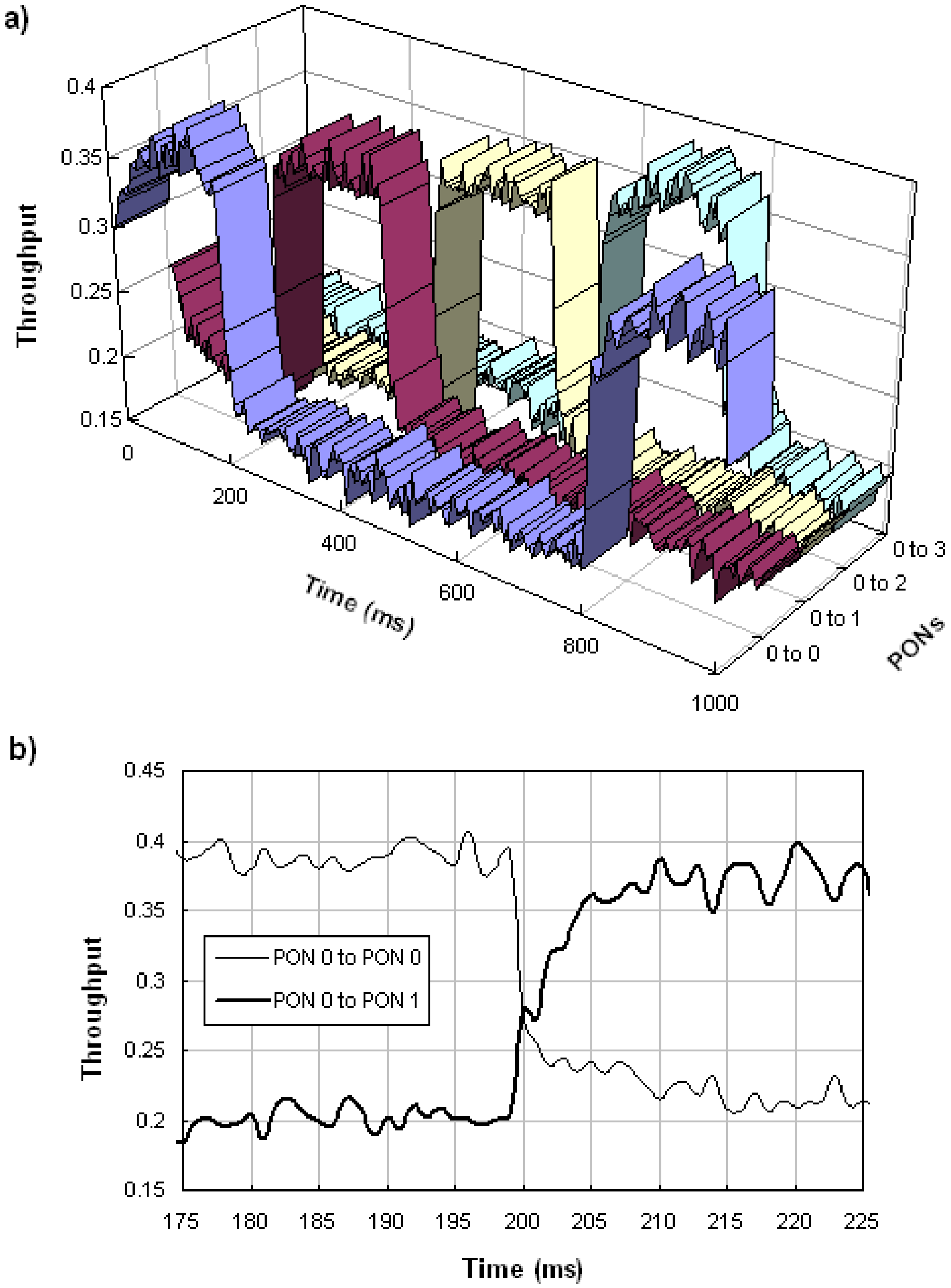

Figure 4.5: Throughput as a function of the offered load under dynamic diagonal traffic matrix

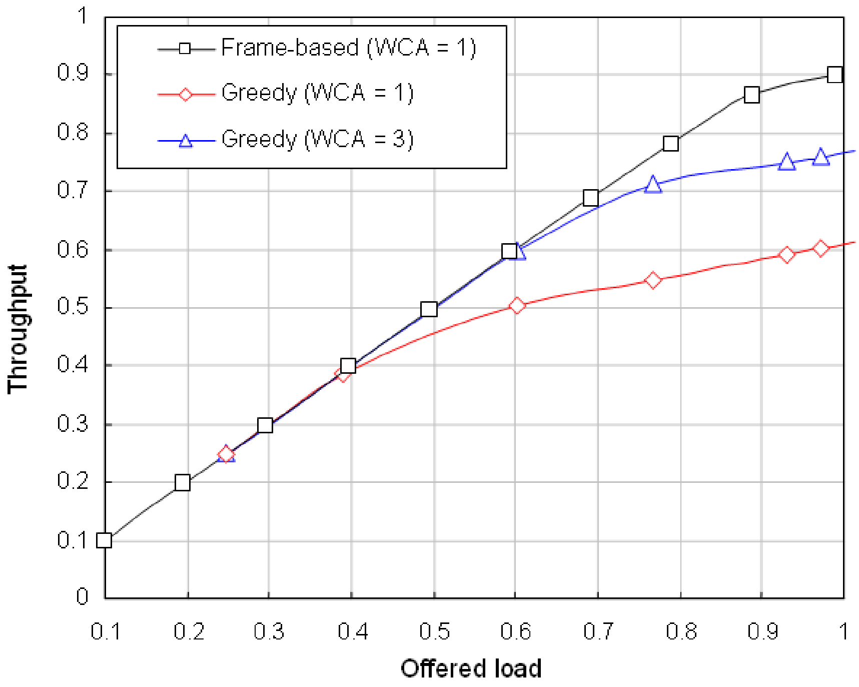

Figure 4.6: Throughput as a function of the offered load comparing the Greedy and the Frame-based algorithm

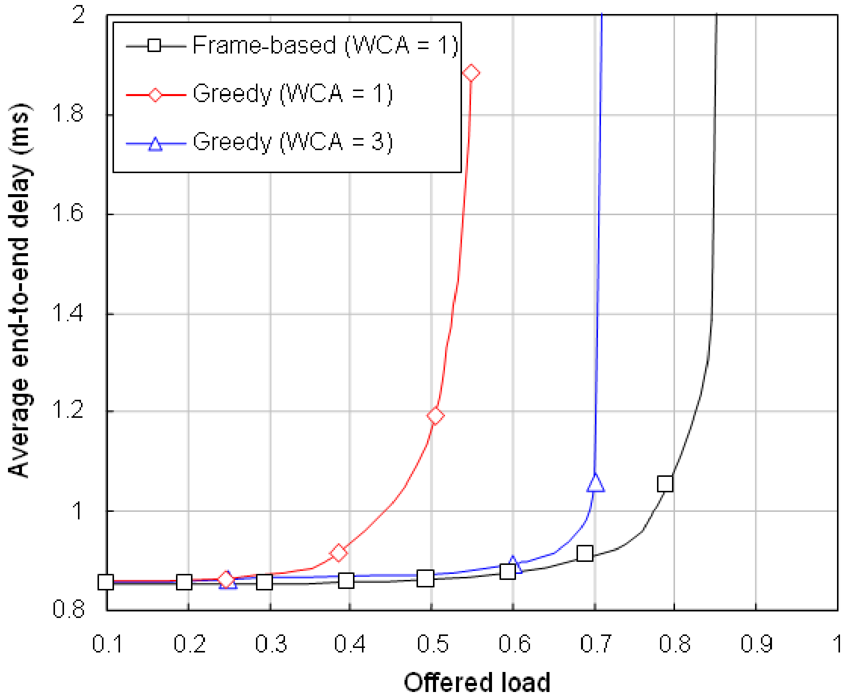

Figure 4.7: Average end-to-end delay as a function of the offered load comparing the Greedy and the Frame-based algorithm

4.4 Optimization

Two main problems can be identified from the performance evaluation section.

The first one regards the waiting time in the queuing stage of the node due to the Head of the Line (HoL) blocking [77]. The HoL blocking is typical of FIFO queues: a packet at the head of the FIFO queue that cannot be transmitted to avoid collisions or contentions on the PON may prevent a successful transmission of another packet following in the FIFO order. To avoid this undesirable situations, the optimized queueing stage is composed by several queues, one per each destination PON. In this way, the queue architecture is very similar to the VOQ (Virtual Output Queue) architecture used in Input Queued (IQ) switches [77], where, at each input port, packets are stored in separate queues on the basis of the destination port they should reach.

The second one regards the retransmission of the nodes' requests after having been received a negative allocation answer from the NC. The physical distance between nodes and NC adds important latency which lead to a drastic increase of the waiting time in the queues. To cope with this problem, the requests are pipelined to the NC, i.e., the node does not wait for the corresponding grant to arrive, but sends new requests in every frame. In such a way, if a node sends a request to the NC and receives a negative answer, it does not require to send it again during the next frame. The NC maintains information on the non-allocated requests and tries to schedule them in next frame together with the new incoming requests.

The optimized solution has been compared with the original solution using the Frame-based algorithm and a network with $P = 4$ PONs, $n = 32$ nodes, $W = 32$ wavelengths, $WCA = 1$, and $F = 100$ slots per frame. The power-of-two traffic matrix is used for this study.

Figure 4.8 and Figure 4.9 plot the throughput and the maximum end-to-end delay, respectively, as a function of the offered load comparing the original and the optimized solution. It is evident that both measures indicate that the optimized solution achieves better results reaching very good performance.

4.5 QoS provisioning

4.5.1 Problem formulation

In Chapter 3 we discuss that we are interested to support 3 different services: guaranteed, priority and best-effort. Since work in [6] proposed a method to support guaranteed service and best-effort, we concentrate our contribution in proposing a method to support the priority service. Since the priority class only requires better treatment than best-effort without precluding the guaranteed service and/or increasing the control complexity, the mechanism to provide it must be as simple as possible. Following this directive, we design a distributed scheme that need some additional functionalities at the nodes while little changes are required for both the scheduling and the MAC protocol.

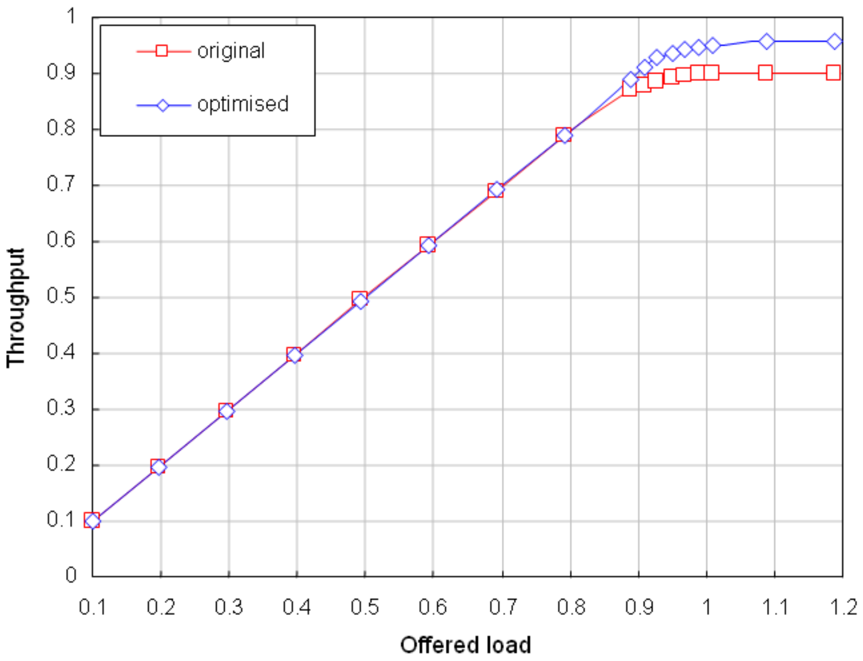

Figure 4.8: Throughput as a function of the offered load comparing the original and the optimized solution

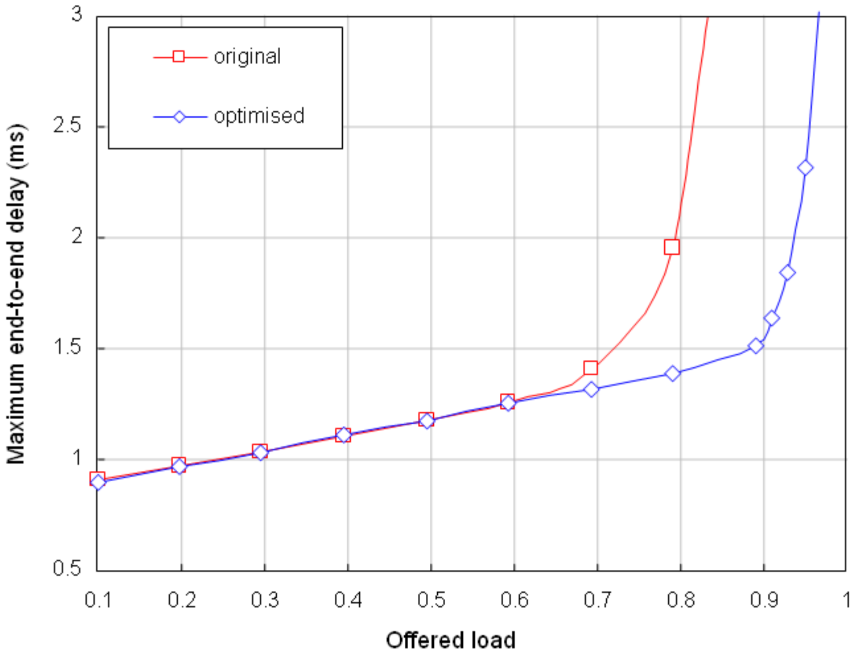

Figure 4.9: Maximum end-to-end delay as a function of the offered load comparing the original and the optimized solution

4.5.2 QoS strategy: the Limited Attempts (LA) technique

This optical packet network has no optical buffers and its switching functionality can be non-blocking for any traffic matrix with sufficient WC arrays. Therefore we assume that the scheduling algorithm along with the adopted QoS strategy are the only factors determining the maximum delay and the packet loss rate because of failing to serve the requests within the QoS constraints. While still yielding the maximum possible throughput, the scheduling algorithm and the nodes must ensure that the PLR required by the traffic class is achieved and that the maximum delay reaches acceptable levels. Hence we are not interested in controlling the delay or in controlling its variation -as opposed to other propositions [2]- as a QoS strategy. Two traffic classes are considered, namely a Best-Effort (BE) class as low priority traffic and a High-Quality (HQ) class as high priority traffic; the latter offering lower PLR and limited maximum delay.

The novelty of the new QoS strategy presented in this work lies in the combination of two different mechanisms, distributed between the scheduling algorithm (matching + TSA) in the central node and more importantly the way switching requests are issued in the nodes. The multi-class-adapted, scheduling algorithm works in the central node giving different priorities to different types of traffic. On the other hand, nodes give switching requests further opportunities in subsequent frames when they fail to be served in the current frame being scheduled; the number of attempts depending on the type of traffic. The number of attempts is pre-determined (h for BE and k for HQ traffic) and when requests fail to be served after (h, k) times respectively they are dropped regardless of the buffer backlog. Varying the pair (h, k) is a compromise between achieving low losses and limiting the maximum delay. We call this approach the Limited Attempts (LA) technique.

Various possibilities exist for applying the scheduling algorithm to different classes of traffic. Here we present two methods for handling High and Low Priority types of traffic.

1. The first method tries to maximize the overall throughput simultaneously for both High Priority (HP) and Low Priority (LP) traffic requests. HP traffic takes precedence over LP traffic in the granting and acceptance phases of the matching process, which is run only once for both types of traffic. We refer to this technique as Throughput Maximization (TM);

2. The second method maximizes the throughput of the HP traffic class. In this case, the HP requests are scheduled first, then the LP requests afterwards using the remaining resources. Hence the matching algorithm must be run twice and this is the 'traditional' technique [100]. We refer to this technique as High-Priority Maximization (HM).

4.5.3 Performance evaluation

Although we only show results regarding the Frame-based algorithms, the same QoS strategy can be applied to the Greedy algorithms as shown in [29].

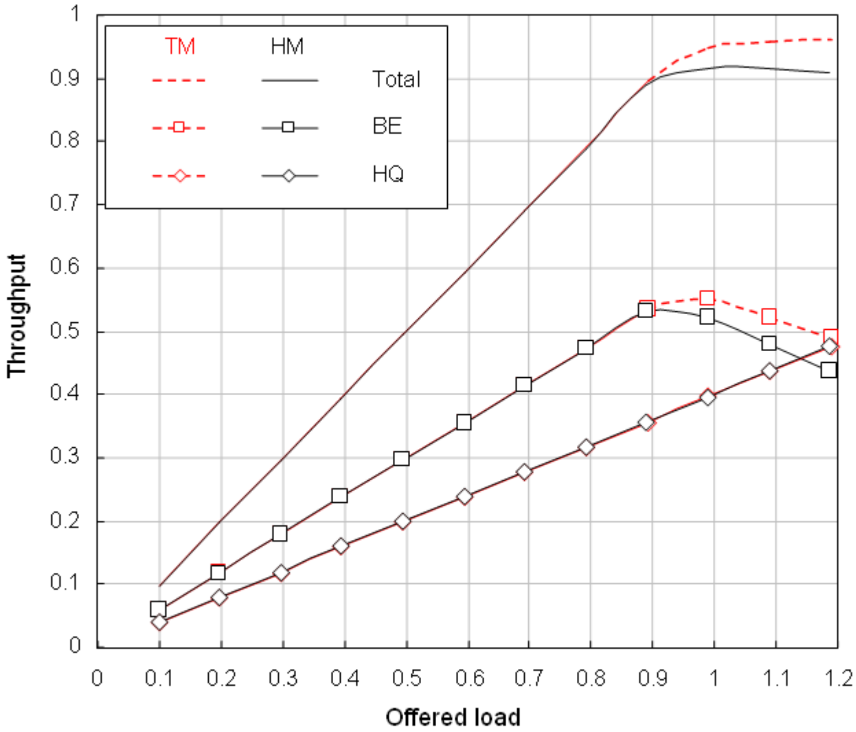

Figure 4.10: Throughput as a function of the offered load comparing TM and HM techniques and considering $(h = 3,\ k = 7)$

In next figures we consider a network with $P = 4$ PONs, $n = 32$ nodes, $W = 32$ wavelengths, and $WCA = 1$. The power-of-two traffic matrix is used for this study. Results shown look at a node pair within the PON belonging to the anti-diagonal links in power-of-two traffic matrix. As already stated, packets are lost only when requests fail to be served after $(h,\ k)$ request attempts have been given.

In Figure 4.10–4.13 we assume a bullish scenario for high quality traffic; 40% of the traffic is HQ and 60% is BE. This balance is changed in Figure 4.14 by varying the HQ traffic percentage from non-existent to 100%, always at 100% load. Figure 4.10 compares the TM and the HM techniques considering $(h = 3,\ k = 7)$, while Figure 4.11, Figure 4.12 and Figure 4.13 show curves for different values of $(h,\ k)$ using the TM technique. Figure 4.14 shows the throughput attained by each traffic type and overall traffic when the percentage of HQ traffic changes.

Although very similar, Figure 4.10 shows that at very high loads and at congestion levels, the overall throughput and BE throughput are higher using the TM method than the HM. HQ traffic is served as requested with either of the two techniques at the expense of BE traffic when the overall demand cannot be satisfied and therefore the HQ throughput is maintained by the multi-class matching technique. As we are interested in maximizing the overall throughput, and for the sake of brevity are only considering this result and not other issues such as delay, we have chosen the TM technique for the rest of the QoS strategy evaluation. Hence this choice is not intended to demonstrate a superiority of one technique over the other, as more detailed studies

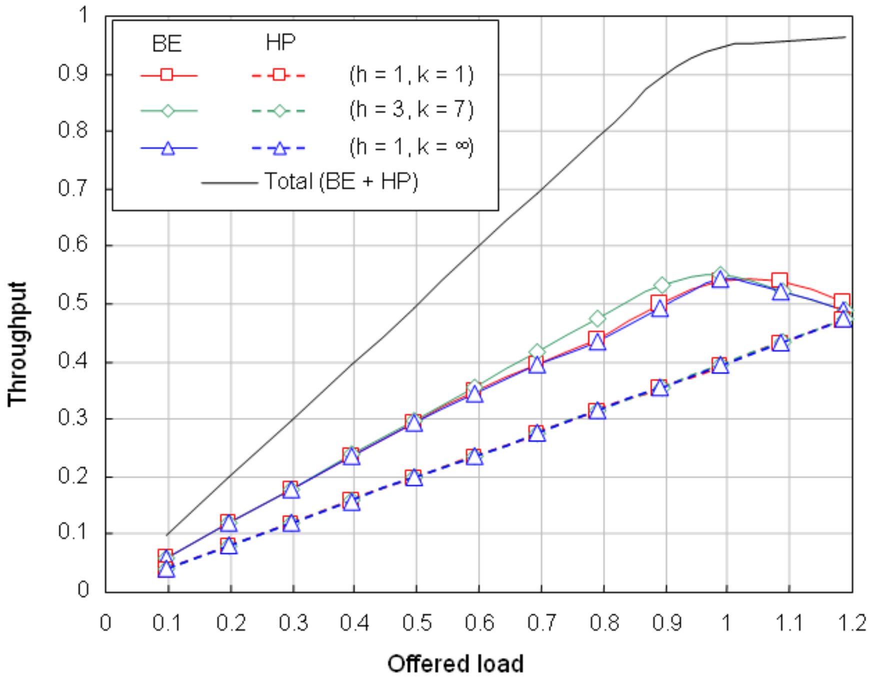

Figure 4.11: Throughput as a function of the offered load comparing different values of (h, k) and using the TM technique

would be necessary and most likely it would rather depend on particular scenarios and strategies.

Figure 4.11 shows the class-relative and total network throughput as a function of the offered load. The different tested (h, k) values have little effect on the network throughput. For low values of offered load, both BE and HQ traffic grow proportionally to their level, and above 50% load differences are small. At congestion levels, i.e., for total loads higher than the network capacity, the amount of admitted BE traffic decreases to ensure the transmission of HQ traffic, which means that this QoS strategy enforces priority to HQ traffic. Losses are small and the graph resolution does not show them until very high loads are reached.

Figure 4.12 shows PLR as a function of offered load. For any value of h, HQ traffic has the highest and the lowest losses for $k = 1$ and $k = \infty$ respectively. The losses for the latter case do not show up in the graph meaning that they are lower than 10^{-8} and therefore practically non-existent (they are not measurable within the simulation time). For the other simulated case, i.e. $k = 7$, losses are very small, always lower than 10^{-5}. Note that the results for the HQ traffic may change for different values of h. On the other hand, BE losses for $h = 3$ are remarkably lower than for $h = 1$, almost two orders of magnitude at high loads. These results show the good performance of this QoS strategy. However, we do not only need a reasonable average delay, but also a bounded maximum delay within reasonable values.

Figure 4.13 shows the maximum end-to-end delay (worst case) as a function of offered load. Obviously, it is not possible to simulate all possibilities, and therefore

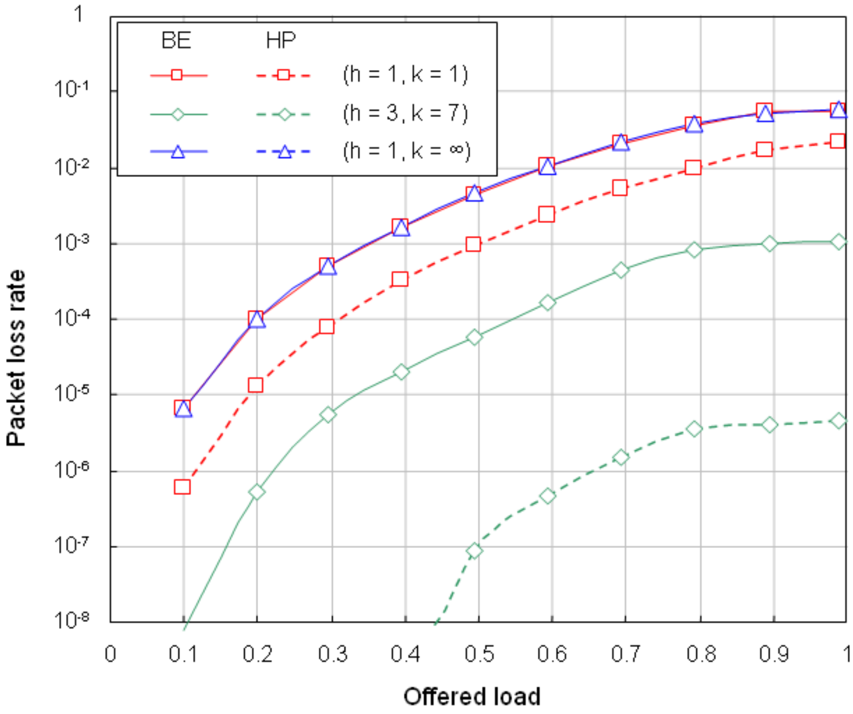

Figure 4.12: Packet loss rate as a function of the offered load comparing different values of $(h,\, k)$ and using the TM technique

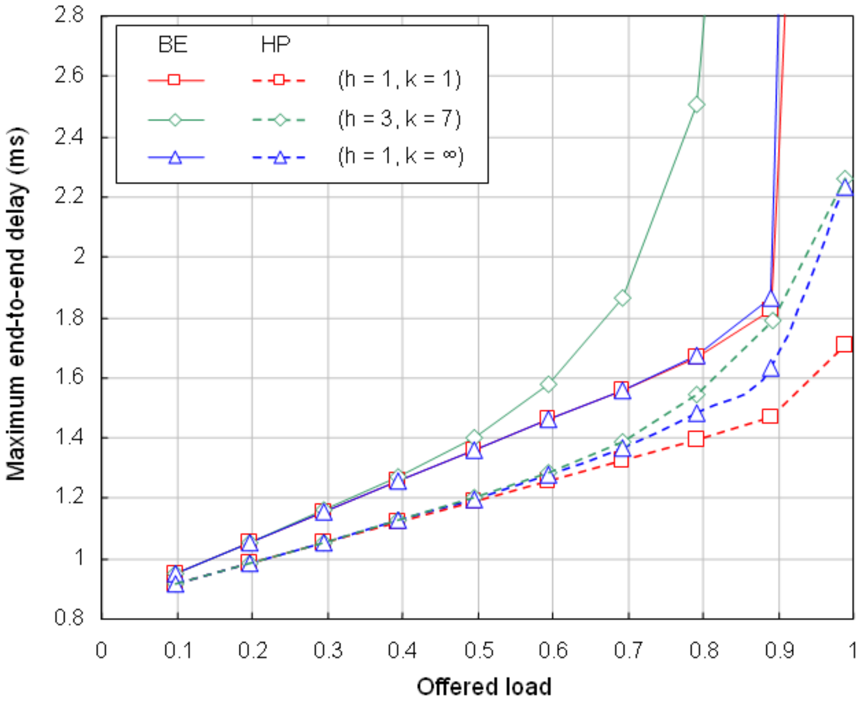

Figure 4.13: Maximum end-to-end delay as a function of the offered load comparing different values of $(h,\, k)$ and using the TM technique

the obtained maximum delay must be understood as an approximation. Nonetheless, the high number of simulation runs brings confidence that the delays shown are good approximations and can be reasonably expected. Buffer lengths are assumed to be infinite and therefore packets are not dropped because of buffer overflow, but only when packets have used up all their opportunities to send switching requests. Although Figure 4.13 shows almost vertical curves, the delays are huge but still finite, which for the sake of clarity are not shown. Even though the (h, k) values are finite, and it would seem every packet eventually should be either successfully switched or dropped, in some cases the backlogged packets are unable to send the switching requests because of high traffic load, remaining in the infinite queue, and hence the possibility of huge delays. Undoubtedly, in a real case where the buffers are finite, once a maximum delay is reached packets that have not been able to send switching requests would be discarded and losses in Figure 4.12 for BE traffic would be higher at very high loads. If we were to use other variants of the frame-based matching algorithm these delays would be shorter so the PLR with finite buffers would come close to these results.

Loosely speaking the higher the (h, k) values the higher the delays, but particular values for each traffic type affect the delay of the other class. Although this paper does not attempt to find the optimal values of (h, k), we see that h has a stronger effect on the delays than k. This is due to the higher priority of HQ traffic true for whatever value is given to (h, k) and thus it is the BE traffic that suffers the most. Delay and PLR are mutually dependent and each one can only improve at the expense of the other. For example, with $(h = 1, k = 1)$ at 100% traffic load BE and HQ traffic experience a PLR of 6×10^{-2} and 2×10^{-2} respectively, and the maximum delay for HQ traffic is 1.7 ms (1700 slots). However, whilst with $(h = 3, k = 7)$ BE and HQ traffic experience lower PLR of 10^{-3} and 4.5×10^{-5} respectively, the maximum delay for HQ traffic has now increased to 2.26 ms (2260 slots).

Finally, Figure 4.14 shows the throughput as a function of the relative percentage of HQ traffic, always at 100% total offered load (HQ+BE). The dotted line represents the relative bandwidth not used by HQ traffic left for BE traffic. The achieved throughput for HQ traffic perfectly matches the relative load percentage increase, until it reaches a value of 95% at 100% relative percentage of HQ traffic. This result shows the robustness of the QoS strategy in terms of throughput, ensuring the service of the HQ traffic with low losses, up to very high loads.

4.5.4 Comparison with other QoS techniques

Other similar QoS methods are present in the literature. For instance Absolute Priority (AP), and more effective Random Early Discard (RED) mechanism [47] are used to provide QoS applying selective packets dropping techniques. Nonetheless, our proposal is more appropriate for the considered environment. If a node sends a request to the NC and receives a negative answer, it does not require to send it again during the next available frame. Indeed, the NC maintains information on the non-allocated requests and tries to schedule them in each frame together with the new requests. The NC also maintains information on the number of attempts

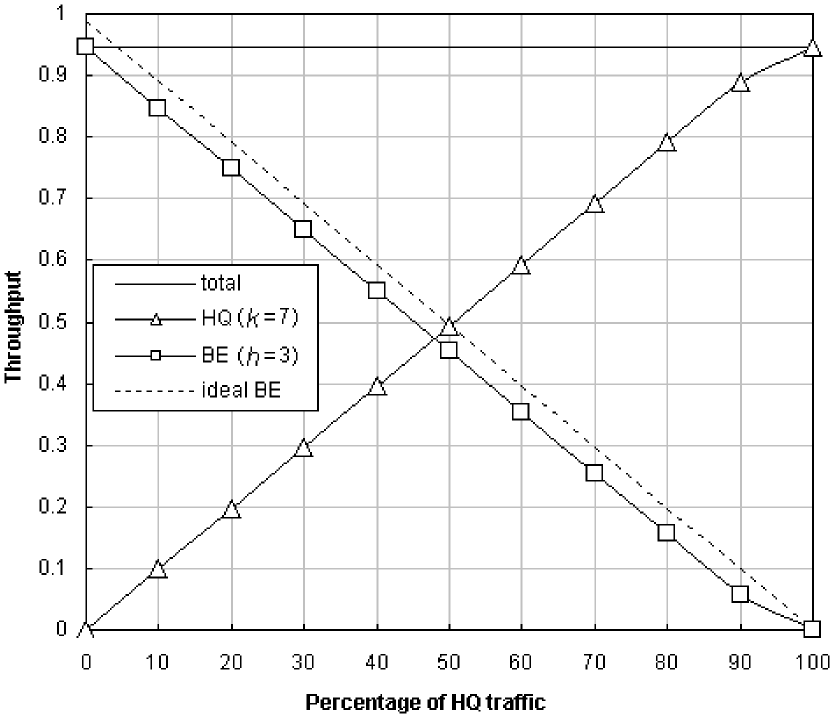

Figure 4.14: Throughput as a function of HQ traffic relative load percentage at 100% total load using the TM technique

used per each request and removes those that already spent all opportunities. At the same time, the nodes drop the corresponding packets that receives too many negative answers. The same coordinate strategy is not possible to apply to neither AP nor RED mechanism and therefore the nodes must resend new request for any negative answer.

In Figure 4.15 and Figure 4.16, we compare our mechanism (LA in the figure) with the AP and RED mechanisms in terms of network throughput and maximum end-to-end delay, respectively. The network uses the Frame-based algorithm and consider 40% of HP traffic while the rest is BE traffic. We consider a network with $P = 4$ PONs, $n = 32$ nodes, $W = 32$ wavelengths, and $WCA = 1$. The traffic matrix is the power-of-two matrix while the traffic model is self-similar.

The figures prove the advantageous of our proposal. While the AP presents the worse results in both comparisons, RED and LA show similar PLR but LA outperforms the RED in terms of delay.

4.6 Summary

The performance of the multi-PON architecture has been evaluated considering several simulation scenarios. The performance results have been obtained using a real scale simulator including self-similar traffic model and different traffic patterns between interconnected PONs.

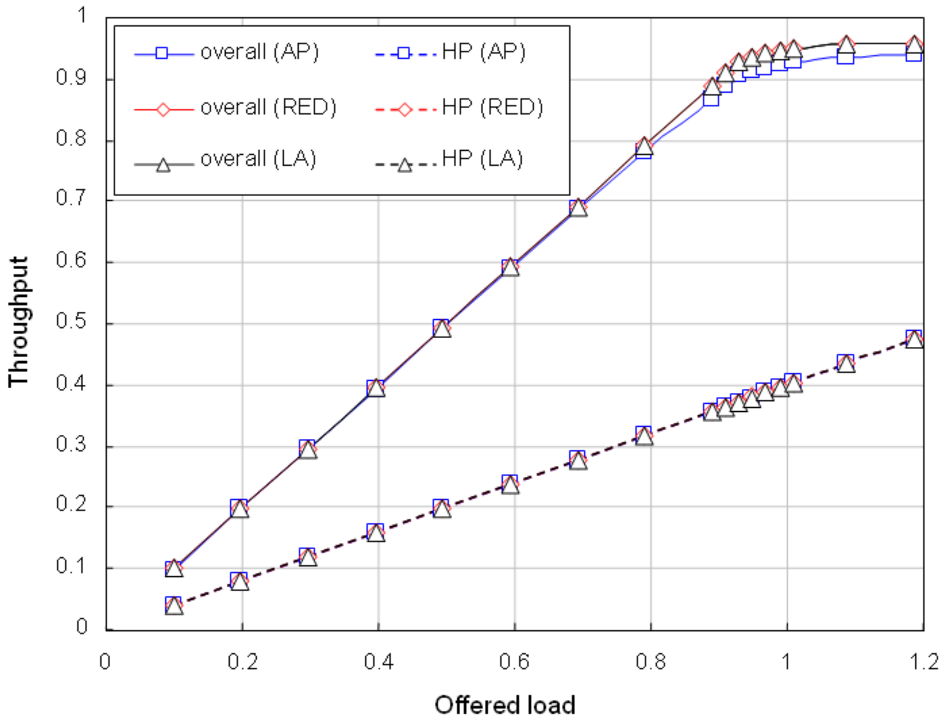

Figure 4.15: Throughput as a function of the offered load comparing the AP, RED and LA

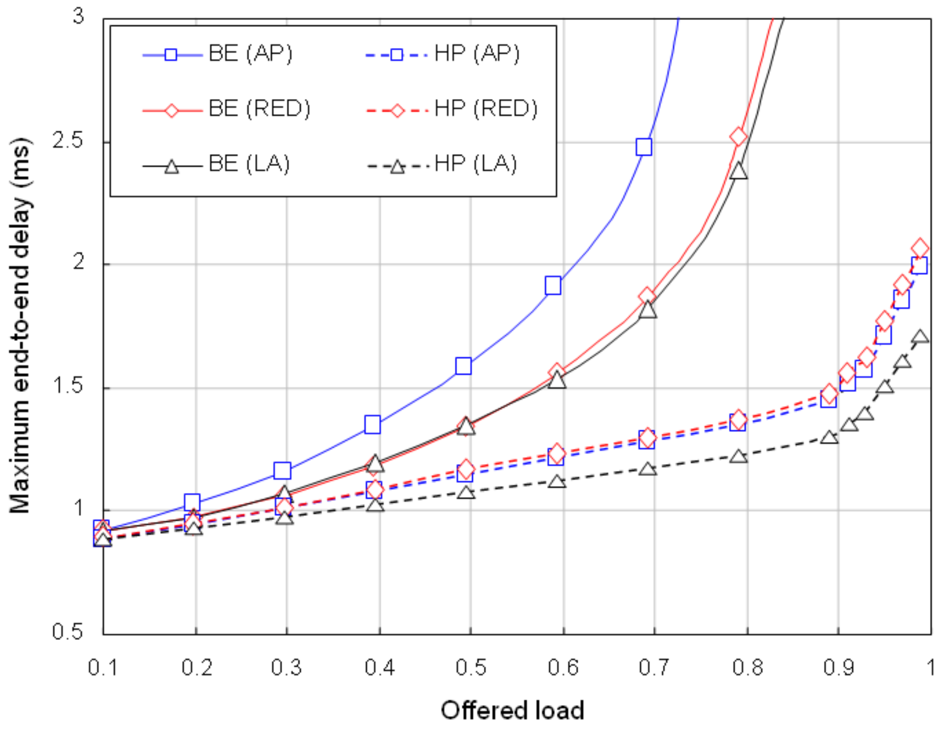

Figure 4.16: Maximum end-to-end delay as a function of the offered load comparing the AP, RED and LA

Two main weaknesses have been identified and overtaken by optimized mechanisms. The validity of the proposals have been demonstrated, indeed both throughput and maximum end-to-end delay measures achieve better results than the original proposal.

A QoS strategy for two classes of traffic has been also suggested. This strategy achieves a clear differentiation between the 2 traffic classes, each complying with their requirements in a very robust way, i.e., with no effect on the network throughput. We have also tested two different techniques for handling the two classes of traffic when using a frame-based matching algorithm. One technique runs the algorithm on a per traffic class basis, i.e. sequentially running the matching algorithm for each traffic class. The other technique runs the matching algorithm once, giving precedence to the high priority traffic in the granting and acceptance phases. Results in terms of overall throughput, tested for a particular traffic balance between two types of traffic, are better when running a single instance of the algorithm. However, neither technique affected the performance of the high priority traffic. The single instance technique was used to compare performance results for different numbers of opportunities in different frames given to each type of traffic.

The proposed novel QoS strategy shows very good performance for the two considered classes, namely Best-Effort and High-Quality traffic. The High-Quality class results show low Packet Loss Probability and bounded maximum delay, whilst acceptable levels are also achieved for the Best-Effort class. These results, obtained for a particular traffic balance, have also been shown, in terms of throughput, to be valid for other traffic balances where the percentage of High Quality traffic was varied from nil to 100%.