A Quality of Service Scheduling Technique for Optical LANs

Panagiotis G. Sarigiannidis, *Member, IEEE,*
Sophia G. Petridou, *Member, IEEE,*
Georgios I. Papadimitriou, *Senior Member, IEEE*
Department of Informatics
Aristotle University
Thessaloniki, Greece 54124
Email: {sarpan,spetrido,gp}@csd.auth.gr

Mohammad S. Obaidat *Fellow, IEEE*
Department of Computer Science
Monmouth University
West Long Branch, NJ 07764
Email: obaidat@monmouth.edu

Andreas S. Pomportsis
Department of Informatics
Aristotle University
Thessaloniki, Greece 54124
Email: apombo@csd.auth.gr

***Abstract*—Quality of Service (QoS) support has become a key factor in designing Media Access Control (MAC) protocols. This paper introduces a novel scheduling scheme which supports priority based QoS in Wavelength Division Multiplexing (WDM) star networks. The proposed Interval-based Prioritized Orderly Scheduling Strategy (IPOSS) employs a collision-free scheduling approach and handles variable-length data packets. In practice, it is designed to handle real-time traffic, on the basis that each node may generate high- and low-priority packets with high-priority packets being scheduled prior to low-priority ones. Moreover, the proposed scheme differentiates the packets' schedule order, by prioritizing the long-length over the short-length packets. The performance of IPOSS is evaluated under Bernoulli traffic and simulation results indicate that the novel scheme achieves a significantly high throughput-delay performance for real-time traffic, without sacrificing the performance for non-real-time traffic.**

I. INTRODUCTION

The increasing number of Internet users along with the demanding network applications seem to be satisfied by the WDM technology [1], [2]. Up to now, a large number of MAC protocols has been proposed in order to handle the nodes' transmission requests in an optical WDM star network [3]–[6]. However, if we focus on a network, in which each node is equipped with a tunable transmitter and a fixed receiver (TT-FR), without any control channel, then we can observe that there is a lack of MAC protocols, supporting QoS. In this paper, a novel scheduling scheme is introduced in order to meet the QoS requirements in a TT-FR optical WDM star network. The proposed Interval-based Prioritized Orderly Scheduling Strategy (IPOSS) protocol falls in pre-transmission coordination based category, due to the fact the transmission schedule is dynamic and depends on the network traffic. A scheduling algorithm is applied to construct the service order and the transmission program (schedule). The proposed algorithm manages to schedule real-time traffic, by prioritizing the real traffic data packets, which have high-priority, over the nonreal traffic data (normal data), which have low priority. In this way, an application such as an urgent alarm message or teleconference data generates high-priority packets, while a file transfer or a text message produces low-priority ones.

The new scheduling strategy comes as an improvement of the scheduling algorithm proposed in [6]. The novelty of IPOSS is that it supports QoS, realized with high- and low-priority packets. In practice, it differentiates packets' schedule order, since high-priority packets precedes the low-priority ones in transmission procedure. At the same time, IPOSS adopts the strategy of linear search with equality examination (ls-ee), which is an improvement of linear search for K elements (K-ls), proposed in [6]. According to ls-ee strategy, long-length requests are prioritized over the short-length ones, in such a way that a sorting process is avoided for complexity reasons. Furthermore, IPOSS determines the maximum length request of each reservation phase and then modifies the service order of each request, by giving priority to the most demanding requests. The new strategy also considers the case that two or more requests demand the same transmission time. In practice, while K-ls randomly selects among the competitive requests, the novel ls-ee considers two decision vectors, namely the node time vector (NTV) and the channel time vector (CTV). These vectors indicate the earliest time availability of each channel for the node whose request is being processed. The final choice among the competitive requests is based on these vectors. This schedule strategy allows a better usage of channels' utilization, since the possibility of finding an appropriate interval to schedule short-length requests is increased. Overall, IPOSS keeps the following scheduling order: high-priority long-length requests, high-priority short-length requests, low-priority long-length requests and low-priority short-length requests. If two or more equal-priority requests demand the same amount of transmission time, IPOSS exploits the information of NTV and CTV and schedules the request with the minimum scheduling delay. In other words, the request that can be scheduled earliest is favored among equal-priority requests.

The remainder of this paper is organized as follows. Section II provides the network issues and presents related packet scheduling algorithms for WDM star networks. Section IV presents the proposed scheduling algorithm, while Section V discusses the simulation results. Conclusions are given in Section VI.

978-1-4244-3806-8/09/$25.00 © 2009 IEEE

TABLE I
BASIC SYMBOLS' NOTATION

Symbol	Description
n, w	Number of nodes and data channels
$U = \{u_1, \ldots, u_n\}$	Set of network's nodes
$\Lambda = \{\lambda_1, \ldots, \lambda_w\}$	Set of data channels
D	$n \times w$ demand matrix
D_h	$n \times w$ demand matrix of high-priority packets
D_l	$n \times w$ demand matrix of low-priority packets
NTV	Node time vector
CTV	Channel time vector
$maxV$	max value
t	Schedule's length in timeslots
$L = \{l_1, \ldots, l_t\}$	Set of timeslots
S	$w \times t$ scheduling matrix

II. NETWORK ISSUES

Let us consider a local area WDM single-hop network with broadcast-and-select architecture, consisting of n nodes, which are connected in a passive star coupler via a two-way optical fiber, and w data channels (wavelengths), where $n \geq w$, which are of the same capacity. According to Table I, $U = \{u_1, \ldots, u_n\}$ denotes the set of network's nodes while $\Lambda = \{\lambda_1, \ldots, \lambda_w\}$ indicates the set of data channels. Even though there is no separate control channel for coordination, the proposed protocol is still pre-transmission coordination-based, since the set of w data channels are used for both control and data packets [1]. Thus, each node may transmit data on different channels using a tunable transmitter (TT), while it receives packets in a dedicated channel, also known as home channel, using a fixed receiver (FR), as depicted in Fig. 1.

In the above TT-FR implementation, transmission is organized into frames, where each frame consists of a reservation (or control) phase and a data phase. During the reservation phase, the n nodes include in their control packets the priority information of their data packets i.e. $p_r = 1$ for high-priority packets and $p_r = 0$ for low-priority packets, while they also send their requests to the common data channels. Nodes' requests are formed as variable-length messages consisting of one or more fixed-length data packets and time is divided into timeslots, where each data packet is transmitted in time equal to a timeslot. Real-time and non-real-time requests are recorded in the $n \times w$ D_h and D_l demand matrices, respectively, where $d_h(i,j)$ ($d_l(i,j)$) element, $i = 1, \ldots, n$ and $j = 1, \ldots, w$, indicates the number of high-priority (low-priority) data packets at node u_i that are destined for channel λ_j. Based on D_h and D_l, D can be defined as $D_h + D_l$, where $d(i,j)$ element represents the total number of data packets at node u_i that are destined for channel λ_j. The proposed IPOSS operates in conjunction with a distributed scheduling algorithm and produces the w x t scheduling matrix S, where t denotes the length of the schedule in timeslots. Each $s(i,j)$ element, $i = 1, \ldots, w$ and $j = 1, \ldots, t$, represents the node that transmits on channel λ_i during the timeslot l_j. During the data phase the packets' transmission takes place according to the matrix S which was built during the reservation phase.

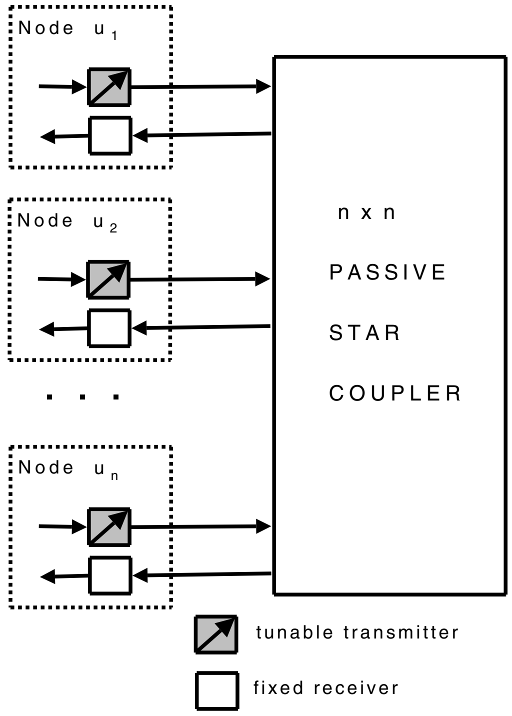

Fig. 1. Network Topology.

III. RELATED SCHEDULING ALGORITHMS

The proposed IPOSS algorithm is based on two pre-transmission coordination-based MAC protocols namely the Online Interval-based Scheduling (OIS) [7] and the Predictive Online Scheduling Algorithm (POSA) [8]. OIS and POSA are characterized as online schemes and prevent channel collisions by employing a distributed scheduling approach. OIS protocol incorporates online scheduling on the basis that the scheduling algorithm begins the schedule construction once the first node's control packet is received. In other words, OIS calculates the schedule when the first row of the demand matrix D is known. It has low time complexity and it is very simple in practice.

More specifically, the OIS algorithm maintains two sets of intervals, one for each channel and a second for the node whose reservation is currently being scheduled. Let us suppose that the list of the available intervals for a certain channel λ_j at a time point contain the timeslots $[2, \ldots, 7]$ and $[12, \ldots, \infty)$. This implies that timeslots 1 and $[8, \ldots, 11]$ have already been assigned to one or more nodes and, thus, they are blocked for any other node for the current frame. Furthermore, for each node whose request is being processed, the algorithm maintains an additional list of intervals. This second list represents the timeslots that have not yet been assigned to this node. If we assume that the list of intervals for a certain node n_i contains the timeslots $[7, \ldots, 11]$, $[16, \ldots, 22]$ and $[30, \ldots, \infty)$, then, it holds that the node n_i has already been scheduled to transmit at timeslots $[1, \ldots, 6]$, $[12, \ldots, 15]$ and $[23, \ldots, 29]$. Given the available intervals of channel λ_j and node n_i, a possible request $d(i,j) = 4$ would be scheduled during the timeslots $[16, \ldots, 19]$.

The OIS protocol produces a schedule for transmitting the requests of demand matrix D and runs in time linear on the number of nodes n i.e. $O(nw^2k)$, where k is the upper bound of nodes' requests on each channel. The fact that OIS is a simple algorithm including simple and fast procedures without decomposing the demand matrix D, as other protocols of this category (HRP/TSA [9]), offers powerful advantages as far as its execution is concerned. The main drawback of OIS is that it requires a lot of construction time for the final schedule of each frame, since the data transmission should wait for the completion of the schedule of each frame.

POSA tries to eliminate the possible delay introduced by the scheduling computation between the control and data phases of each frame in order to reduce the waiting time of the nodes due to the control phase. POSA is actually an extension of OIS which is based on traffic prediction according to the history of recent reservation requests. POSA assumes that the reservation information of each control phase constitutes the input to a mechanism consisting of n x w predictors whose operation is mainly based on the Hidden Markov chain Model. More specifically, after an initial period i.e. learning phase, the protocol learns the traffic pattern by maintaining a reservation history and then switches into a prediction phase. During the prediction phase, POSA calculates the schedule based on the predicted reservations (of the previous control phase), instead of the actual reservations of the current control phase. As a result, POSA offers more computational time for the construction of the next schedule and, thus, the schedule is available at the beginning of the data phase.

It is important to be mentioned that during the learning phase which lasts, for example, v frames, the POSA protocol operates like OIS. Furthermore, the prediction mechanism of POSA should be accurate enough otherwise it leads the scheduling algorithm to false output. According to [8], it has been found that in the 70% of the predictions, the percentage of failure was less than 20%, regardless of the number of nodes n, channels w or the upper bound of nodes' requests k. Therefore, POSA does not affect the schedule process. Its goal is to minimize the calculation time of the scheduling process. Thus, POSA protocol does not add extra complexity to the system, since its prediction mechanism runs linearly with the increase of nodes or channels: $O(k + 1 + v)(nw)$.

IOSS protocol presented in [6] improves significantly both OIS and POSA. In practice, IOSS achieves a shorter schedule than OIS and POSA, by giving priority to the schedule of the long-length packets. A better channel utilization is obtained, since the possibility of finding an appropriate interval to schedule the short-length requests is increased. The main weakness of the aforementioned protocols is the absence of QoS support.

IV. THE PROPOSED ALGORITHM

The core idea of the proposed IPOSS scheduling algorithm is to improve the previous IOSS scheme and at the same time to provide QoS support. IOSS algorithm reorganizes the service order of the requests per node and per channel based on the value of each entry of the demand matrix. More specifically, it forms the schedule starting from the most demanding request and continuing with the subsequent demanding request. Thus, IOSS uses the K-ls strategy in order to put the requests in order, without the usage of a sorting process. The IOSS main disadvantage is that it randomly selects among equal-length requests, since this type of selection leads to downgraded network performance. On the contrary, IPOSS provides a new approach regarding the quite possible case that a number of requests may be of equal length. IPOSS adopts a new improved strategy, called ls-ee, in order to handle these requests. According to ls-ee strategy, two vectors, namely the NTV and CTV vectors, one for each node and one for each channel are maintained. The $NTV(i)$ element indicates the earliest available time at which the node u_i will be available for transmission, while the $CTV(j)$ element denotes the earliest available time at which the channel λ_j will be available for transmission. IPOSS keeps NTV and CTV informed for each transmission frame. If two or more requests are of equal length, then IPOSS computes the maximum value between NTV and CTV, namely $MaxV$, for each request and it finally chooses the request with the minimum $MaxV$ value.

More specifically, IPOSS algorithm consists of two main phases. During the first phase ls-ee strategy is applied and a reordered demand matrix is returned. This action is operated twice, one for the low-priority requests and a second for the high-priority ones. Next the schedule formation takes place. IPOSS constructs the transmission schedule according to the reordered demand matrices.

First of all, ls-ee strategy finds the request with the maximum length for each priority level. Let us suppose that K_h and K_l denotes the maximum length request for the high- and low-priority elements of the demand matrices D_h and D_l, respectively. Then, ls-ee looks for the K_h or K_l different numbers just once (excepting zero). In other words, IPOSS performs $K_h + K_l$ linear searches to form the two reordered lists σ_h and σ_l. The search order is the following: initially the value of K_h (or K_l), which is the maximum entry of D_h (or D_l) is in search. The search returns all the values of K_h (or K_l) found in matrix D_h (or D_l) and more precisely, returns the position where the elements were found (the number of node and the number of channel of the referring request). So, if while searching for the value K_h (or K_l) the algorithm finds that the value is on the D_h (or D_l) matrix z times, where $z < nw$, then it returns the positions found, i.e., $D_h(i_1, j_1), D_h(i_2, j_2), \ldots D_h(i_z, j_z)$ (or $D_l(i_1, j_1), D_l(i_2, j_2), \ldots D_l(i_z, j_z)$), where $i_1, i_2, \ldots, i_z \in U$ and $j_1, j_2, \ldots, j_z \in \Lambda$. These values are stored in the σ_h (or σ_l). If two or more requests are of equal length then NTV and CTV vectors are computed and the request with the minimum $MaxV$ is selected. Continuously, the selected request is stored in the reordered list σ_h (or σ_l) and is removed from the demand matrix D_h (or D_l).

Afterwards, the algorithm searches for the next biggest value, which is equal to $K_h - 1$ (or $K_l - 1$), in the same way. Next, the algorithm completes the process of linear searching for the requests with length equal to value 1. There is no point in searching for zero value, since it would not have practical interest on the final schedule matrix, because the specific node, on the specific channel does not have any packets to transmit. IPOSS algorithm completes the formation of the σ_h and σ_l lists in $O(nwK_h + nwK_l)$ time. It is obvious that the complexity of IPOSS algorithm, using the ls-ee strategy, is less than the complexity of OIS algorithm which is in $O(nw^2k)$ time, where k is the maximum value of $D_h + D_l$ matrix.

For example, let us suppose that during a transmission frame three requests, namely the $D_l(2,3), D_l(3,1)$ and $D_l(3,3)$ have the same value of 5 data packets. In that case, IOSS would randomly select one among the aforementioned requests. On the other hand, IPOSS applies the ls-ee strategy and computes the NTV and CTV vectors of each request. The $D_l(2,3)$ request is applied form node u_2 to channel λ_3, the $D_l(3,1)$ request comes from node u_3 and destines for channel λ_1, while $D_l(3,3)$ requests comes from node u_3 and demands transmission on channel λ_3. Let us suppose that $NTV(1) = 4, NTV(2) = 3$ and $NTV(3) = 5$, whereas $CTV(1) = 4, CTV(2) = 3$ and $CTV(3) = 6$. It holds that IPOSS will select $D_l(3,1)$ for transmission, since this request has the minimum $MaxV$:

$$maxV(d_l(2,3)) = max(NTV(2), CTV(3)) = 6$$

$$maxV(d_l(3,1)) = max(NTV(3), CTV(1)) = 5$$

$$maxV(d_l(3,3)) = max(NTV(3), CTV(3)) = 6$$

Supporting QoS means that the protocol has to deal with real-time traffic, co-existing with non-real-time traffic. IOSS does not provide support for real-time traffic. The proposed IPOSS comes to cover this issue, by applying a handling method in order to satisfy the demanding needs of real-time traffic. In this manner data packets are divided into high-priority packets, carrying real-time traffic, and low-priority ones, carrying non-real-time traffic. While IOSS does not differentiates high- and low-priority packets, IPOSS schedules high-priority packets prior to low-priority ones. In other words, high-priority packets are scheduled first, independently of their length. Once the high-priority packets' schedule is completed, IPOSS begins to schedule the low-priority ones (at the same schedule matrix). It is clear that ls-ee strategy is executed twice; firstly, it is executed regarding the high-priority packets schedule and, secondly, it is executed regarding the low-priority packets schedule. Consequently, high-priority packets are scheduled prior to low-priority ones. If two or more equal-priority requests demand the same amount of transmission time, IPOSS selects the request with minimum $maxV$. In this way, a more effective schedule is produced and QoS is supported, giving priority to real-traffic packets.

A. A Numerical Example

This section provides a numerical example which illustrates the IPOSS scheme and the way that it applies the ls-ee strategy and supports QoS. It is assumed that the WDM star network consists of $n = 3$ nodes i.e. $U = (u_1, u_2, u_3)$ and $w = 2$ data channels i.e. $\Lambda = (\lambda_1, \lambda_2)$. Given these parameters, the following 3×2 demand matrix D could represent the aggregate network traffic:

$$D = \begin{pmatrix} 2 & 1 \\ 3 & 5 \\ 3 & 4 \end{pmatrix},$$

while the following 3×2 demand matrices D_h and D_l could indicate the real-time and non-real-time traffic respectively. Obviously, it holds that $D = D_h + D_l$:

$$D = D_h + D_l = \begin{pmatrix} 0 & 0 \\ 1 & 2 \\ 0 & 3 \end{pmatrix} + \begin{pmatrix} 2 & 1 \\ 2 & 3 \\ 3 & 1 \end{pmatrix}$$

Example 1: In the above demand matrices the fact that $D(2,2) = 5$ means that node u_2 requests five (5) packets on channel λ_2. Two (2) out of these five (5) packets are of high-priority, since $D_h(2,2) = 2$, while the rest three (3) packets are of low-priority, since $D_l(2,2) = 3$. □

As it has already been mentioned, IPOSS begins to construct the scheduling matrix handing high-priority packets i.e. the matrix D_h. The ls-ee strategy is applied and the following scheduling matrix is formed:

	Timeslots					
	l_1	l_2	l_3	l_4	l_5	l_6
λ_1	$\mathbf{u_2}$					
λ_2	$\mathbf{u_3}$	$\mathbf{u_3}$	$\mathbf{u_3}$	$\mathbf{u_2}$	$\mathbf{u_2}$	

It is clear that the three (3) packets of node u_3 destined for channel λ_2 are favored and followed by the two (2) packets of node u_2 which are also destined for λ_2 channel. Once the IPOSS handles the high-priority packets of D_h matrix, it proceeds to the scheduling of low-priority packets stored in D_l matrix. The matrix D_l has many equal-length requests for whose handing IPOSS employs the ls-ee strategy. More specifically, IPOSS begins with $d_l(2,2)$ and $d_l(3,1)$ requests which are of equal size, i.e. 3, and it examines their NTV and CTV vectors. For $d_l(2,2)$, it holds that, $NTV(2) = 6$, since node u_2 becomes available for transmission after l_6 timeslot, while $CTV(2) = 6$, because the channel λ_2 also becomes available after l_6 timeslot. Thus, $maxV(d_l(2,2)) = 6$. Following the same logic, $maxV(d_l(3,1)) = max(NTV(3), CTV(1)) = 4$. Given that the request $d_l(3,1)$ has the minimum $maxV$, it holds that it is scheduled prior to $d_l(2,2)$. At this stage, the scheduling matrix is updated as follows:

	Timeslots							
	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8
λ_1	$\mathbf{u_2}$			u_3	u_3	u_3		
λ_2	$\mathbf{u_3}$	$\mathbf{u_3}$	$\mathbf{u_3}$	$\mathbf{u_2}$	$\mathbf{u_2}$	u_2	u_2	u_2

Finally, the requests $d_l(1,1), d_l(2,1), d_l(3,2)$ and $d_l(1,2)$ are processed following the same logic and the final scheduling matrix are formed as follows:

	Timeslots									
	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_9	l_{10}
λ_1	$\mathbf{u_2}$	u_1	u_1	u_3	u_3	u_3			u_2	u_2
λ_2	$\mathbf{u_3}$	$\mathbf{u_3}$	$\mathbf{u_3}$	$\mathbf{u_2}$	$\mathbf{u_2}$	u_2	u_2	u_2	u_1	u_3

From the above matrix, where the high-priority packets are depicted in bold face, it is obvious that IPOSS achieves significantly low delay for these packets and, thus, it successfully supports QoS requirements.

V. EXPERIMENTATION

To evaluate the proposed algorithm we carried out experiments, where we compared IPOSS with IOSS and POSA. The performance of the compared algorithms is evaluated in terms of network throughput and mean packet delay. Network throughput represents the average number of bits transmitted per frame on each channel, while mean packet delay denotes the mean time in timeslots that packets wait at the queues till the beginning of their transmission. It is crucial to keep the mean delay of high-priority packets low, in order to avoid long delays.

The experiments carried out are based on the following assumptions:

1) Traffic pattern follows the Bernoulli distribution.
2) The line is defined at 3 Gbps per channel and the tuning time is considered to be negligible.
3) The outcome results from 10000 transmission frames.

Fig. 2 and 3 depict the mean delay of aggregate, real-time and non-real-time traffic as a function of network throughput and network load, respectively, for $n = 30$ nodes and $w = 10$ channels. Given that IOSS and POSA do not handle real-time traffic, it is natural to have only one curve to present their mean delay. On the other hand, the mean delay of high- and low-priority packets differ significantly under IPOSS scheme. It is important to mention that IPOSS clearly outperforms POSA in terms of mean delay for both high- and low-priority packets. Given that IOSS has already been superior to POSA, it was expected that IPOSS will be marginally improved in comparison to IOSS in terms of aggregate traffic, since they are of the same logic. However, IPOSS's better performance is obvious under real-time traffic, where it achieves significantly lower delay compared to IOSS. This is due to the fact that high-priority packets have the privilege of being scheduled prior to low-priority ones. As shown in these figures, the mean delay of high-priority packets in case of IPOSS is up to 95.4% and 90.6% lower than the corresponding mean delay of POSA and IOSS, respectively. This significant improvement is not made in the cost of a high delay of low-priority packets, since as depicted in Fig. 2 and 3 the non-real-time traffic curve of IPOSS is very close to that for the aggregate traffic.

In Fig. 4 the network throughput is presented as a function of the network load for $n = 30$ nodes and $w = 10$ channels. The throughput improvement that IPOSS provides over POSA scheme indicates that the use of the proposed algorithm leads to a significant reduction of the schedule's length. This is due to the fact that IPOSS apart from prioritizing the real-time traffic, it also provides to the long-length requests the privilege of being scheduled prior to the short-length ones and, thus, it allocates more free timeslots for the rest of requests. However, the improvement of IPOSS over IOSS is marginal, since the two schemes employ the same logic in handling the long- and short-length packets which contribute to throughput performance.

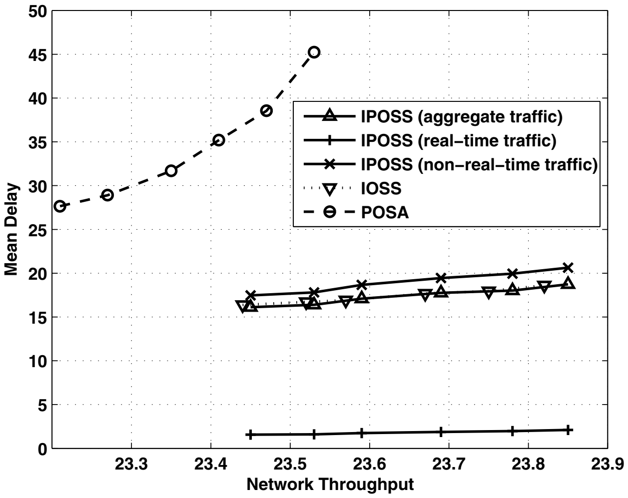

Fig. 2. Mean packet delay of aggregate, real-time and non-real-time traffic as a function of network throughput for $w = 10$ channels.

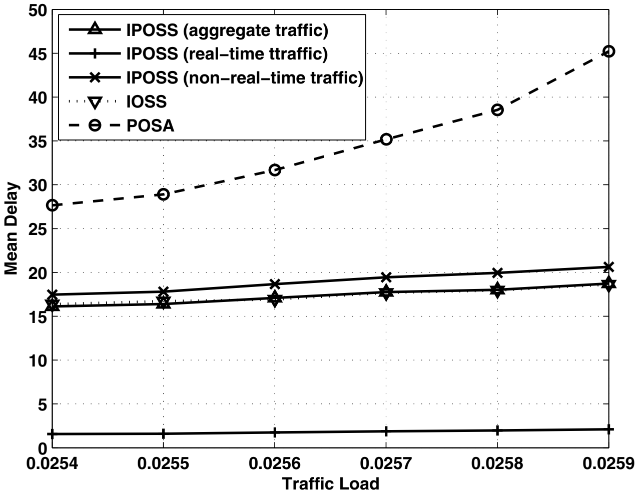

Fig. 3. Mean packet delay of aggregate, real-time and non-real-time traffic as a function of network load for $w = 10$ channels.

Finally, in Fig. 5 we studied the mean delay of aggregate, real-time and non-real-time traffic under different number of channels i.e. $w = 5, 6, \ldots, 10$, for $n = 30$ nodes and load equal to 0.026. In line with the aforementioned results, it is apparent that the proposed scheme is significantly superior under real-time traffic. In practice, as the number of channels

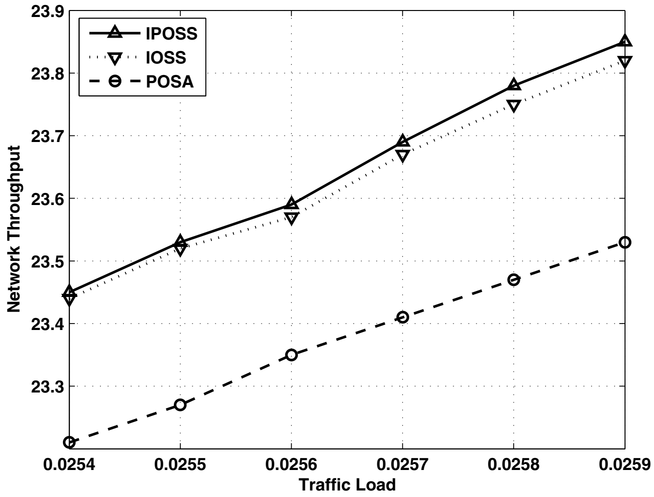

Fig. 4. Network throughput as a function of network load for $w = 10$ channels.

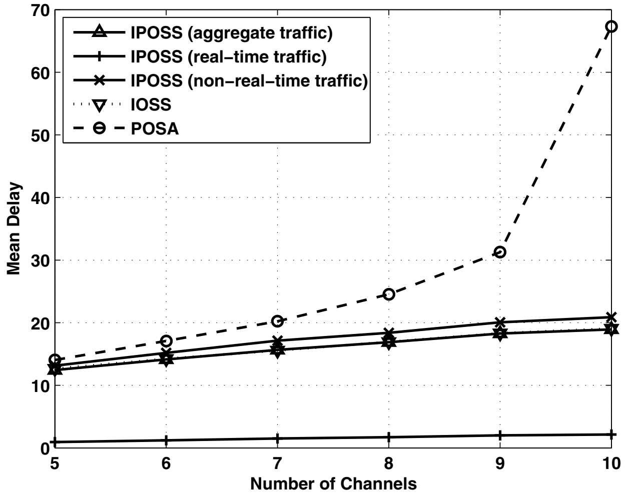

Fig. 5. Mean packet delay of aggregate, real-time and non-real-time traffic as a function of the number on channels.

increases, IPOSS achieves from 93.3% to 96.8% lower delay in comparison to POSA and from 92.6% down to 88.9% lower delay in comparison to IOSS.

Overall, we can claim that the proposed IPOSS scheme is clearly superior to POSA and IOSS in handing real-time traffic, since it creates a shorter schedule which advances the network's throughput and significantly reduces the mean delay of high-priority packets. This is due to the fact that IPOSS exploits the priority information of data packets and thus modifies their schedule by prioritizing the real-time traffic.

VI. CONCLUSIONS

A novel scheduling scheme which supports prioritized scheduling is introduced. The proposed IPOSS handles real-time traffic by providing to high-priority packets the privilege of being scheduled prior to low-priority ones. At the same time, IPOSS takes into account the length of data packets and it manages to produce efficient schedule by prioritizing long-length packets over short-length ones. Furthermore, IPOSS introduces a new scheduling strategy (ls-ee), which considers two decision vectors (NTV and CTV), indicating the earliest time availability of each channel for the node whose request is being processed. Based on ls-ee strategy the final choice among the requests that demand the same amount of transmission frame is based on these vectors, allowing a better usage of channels' utilization, since the possibility of finding an appropriate interval to schedule short-length requests is increased. As a result, the proposed scheme obtains significant throughput-delay improvements for real-time traffic, without, however, sacrificing the performance for non-real-time traffic.

REFERENCES

[1] G. Papadimitriou, P. Tsimoulas, M. Obaidat, and A. Pomportsis, *Multi-wavelength Optical LANs*. NY: Wiley, 2003.

[2] B. Mukherjee, *Optical WDM Networks*. Springer, 2006.

[3] J.-H. Wen, K.-T. Huang, C.-Y. Yang, and T.-C. Tsai, "Timeslot-sharing algorithm with a dynamic grouping for wdm broadcast-and-select star networks," *Journal of Systems and Software*, vol. 79, no. 8, pp. 1110–1119, 2006.

[4] X. Huang, M. Ma, Y. Ma, and Z. Lin, "An efficient qos prediction scheme in single-hop passive star coupled wdm optical networks wireless and optical communications networks," in *Proc. of the IFIP International Conference on*, 2007, pp. 1–5.

[5] P. Baziana and I. Pountourakis, "Performance optimization with propagation delay analysis in wdm networks," *Computer Communications*, vol. 30, no. 18, pp. 3572–3585, 2007.

[6] P. Sarigiannidis, G. Papadimitriou, and A. Pomportsis, "A novel medium access control protocol for optical local networks based on data request ordering," *Journal of Optics & Laser Technology*, vol. 40, no. 1, pp. 175–193, 2008.

[7] K. Sivalingam, J. Wang, J. Wu, and M. Mishra, "An interval-based scheduling algorithm for optical wdm star networks," *J. Photonic Network Commun.*, vol. 4, no. 1, pp. 73–87, 2002.

[8] E. Johnson, M. Mishra, and K. Sivalingam, "Scheduling in optical wdm networks using hidden markov chain based traffic prediction," *J. Photonic Network Commun.*, vol. 3, no. 3, pp. 271–286, 2001.

[9] K. Sivalingam and J. Wang, "Media access protocols for wdm networks with on-line scheduling," *IEEE/OSA Journal of Lightwave Technology*, vol. 14, no. 6, p. 1996, 1996.