Available online at www.sciencedirect.com
ScienceDirect
Optics & Laser Technology 40 (2008) 39–51
Optics & Laser Technology
www.elsevier.com/locate/optlastec

LENA: An efficient channel eclectic algorithm for WDM optical networks

Panagiotis G. Sarigiannidis, Georgios I. Papadimitriou*, Andreas S. Pomportsis

Department of Informatics, Aristotle University, Box 888, 54124 Thessaloniki, Greece

Received 24 October 2006; received in revised form 2 March 2007; accepted 2 April 2007
Available online 23 May 2007

Abstract

Media access control (MAC) protocols are methods and ways of accessing the optical fibers, in order to support communication to the nodes of the network. One of the important themes on the design of high-efficient optical networks is the time duration of the schedule process. A good protocol supports effective scheduling methods, without significant time cost. In this work, the performance of a new pre-transmission coordination based protocol on a wavelength division multiplexing (WDM) broadcast and select optical network is studied. A novel scheduling algorithm is proposed, which maintains a prediction scheme and concurrently constructs the pre-transmission scheduling form, based on the demanded traffic of each available channel. Also, this work includes the presentation of the new algorithm, the comparison of the new algorithm with two prior predicted scheduling schemes, along with the presentation and the description of a series of graphs, which plot the results of our simulation. According to simulation results, it is realized that the proposed algorithm not only improves the output of the network but also reduces the average time delay of the packets in the buffer memories of each node.

Keywords: Optical WDM networks; Service sequence; Scheduling

1. Introduction

It is a fact that the current implementation of the communication networks does not have the available capacity of the growing demands of the multiple end users. Also, the ever-growing demands for communication capacity go stronger and a vital solution must be found in order to meet our tremendous needs [1–3]. The bandwidth required by each individual user has been increased dramatically. Optical technology comes to solve this problem, as optical fibers offer radically higher bandwidths than alternative transmission media [4–8]. More specifically, optical fiber technology can untie the capacity problem because of its great capabilities, as the huge bandwidth, the low signal attenuation, the immunity to electromagnetic interferences, the high security of signal, the absence of crosstalk, the low-signal distortion, the low-power requirement, the low material usage, and the high electrical resistance [9]. If we want to utilize the optical fiber in a cost-effective way, it is useful to share all of its huge capacity among several communication stations. Wavelength division multiplexing (WDM) technique offers an excellent way of exploiting the huge bandwidth of optical fibers by introducing concurrency among multiple users transmitting at 'electronically' feasible rates [10]. By allowing multiple WDM channels to carry data concurrently on a single fiber, we can exploit the corresponding challenges, in order to develop an appropriate network, with an efficient architecture. It is very important to examine the probable architecture form too. The most common forms for WDM networks are (a) point to point, (b) wavelength routing, (c) passive optical networks, and (d) broadcast and select [11,12]. WDM broadcast and select networks comprise a number of nodes and a passive star coupler in order to broadcast from all inputs to all outputs. Every node can select at a given time among the channels available to perform transmission. Transmitters

*Corresponding author.
E-mail address: gp@csd.auth.gr (G.I. Papadimitriou).

0030-3992/$ - see front matter
doi:10.1016/j.optlastec.2007.04.001

and receivers are connected via two-way fibers to the optical star coupler. Each node of the network has a transmitter, in order to send data and a receiver, in order to accept and filter data. Each transmitter and receiver can be fixed or tunable [13,14]. This paper focuses on the single hop broadcast and select star local area network with one tunable transmitter and one fixed receiver (TT-FR) per node (Fig. 1).

It is obvious that a protocol, oriented to WDM TT-FR single-hop broadcast and select network, has to provide two sets of appropriate rules for (a) coordination transmissions between all the nodes of the network, and (b) determination (or elimination) probable collisions. A very effective way to organize the data sending (transmission) and the data acceptance (receipt) by the protocol is to divide the time into two independent periods [15]. During the first period, known as reservation phase, the protocol accepts the total transmission requests of the nodes of the network and performs a scheduling process in order to define the order of the data transmission of each node to the desired transmission channel. During the second period, known as data phase, the real sending of data occurs, according to the method, which has been agreed in reservation phase. This paper presents a novel scheduling technique, which tries to improve the performance of the network. Its main goal is to increase the utilization of the network channels, which achieves improved rates in terms of network throughput. At the same time, the proposed algorithm considers each node and each channel individually and serves the requests of the nodes on each channel according to some criteria and metrics. In other words, simulation results show an improvement in network performance, by changing the order of service and examination of the nodes and the channels while the scheduling process is formed. The improvement that new algorithm brings up is presented through a detailed series of figures and the question whether the algorithm brings extra delay is answered through throughput-delay figures, while the performance of the algorithm is examined in different contexts of network workload through throughput-load figures.

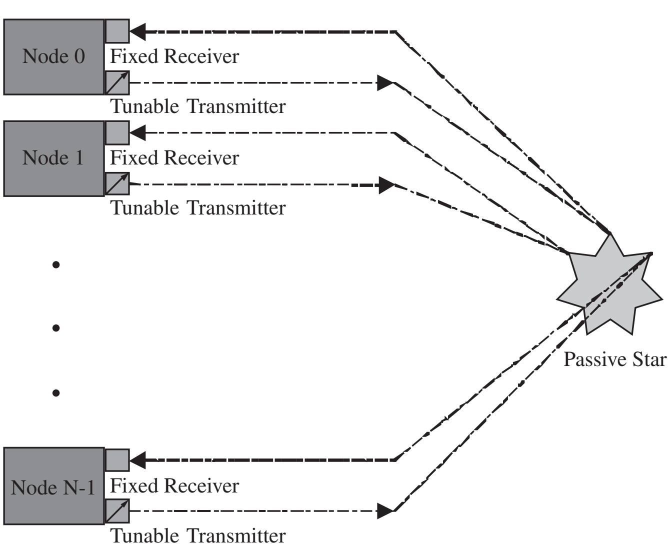

Fig. 1. The broadcast and select topology with N nodes and W channels. Each node is equipped with a fixed receiver and a tunable transmitter.

The paper is organized as follows: Section 2 describes the network assumptions, while Section 3 analyses the three prior scheduling algorithms protocols (OIS, POSA, and CS-POSA) with the previous progress and work to be improved. Section 4 presents the new suggested scheduling algorithm, Section 5 analyzes the performance measures and is followed by the figures and the detailed comparisons between the performances of the whole three algorithms in Section 6. Finally, concluding remarks are given in Section 7.

2. Network structure

The examined network consists of N nodes. Each node communicates by sending and receiving data to and from the rest of $N-1$ nodes. Generally, in a N-node [16] optical network the most effective communication structure is achieved when each one of the nodes has a tunable transmitter and a tunable receiver (TT-TR system) in combination with N channels, one per node. In this way, each node has its own unique home channel and there is not possibly any collision. Such a structure, though, is very difficult to realize in action for two reasons. On the one side, each node may be equipped with only a few channels, due to technological constraints and on the other side the realization of the channels equal to the number of the nodes does not co-exit with the financial standards of the network. So, after all the above, the solution of a TT-FR system seems to go along with the current technological and financial developments. Therefore, we consider that the network has N nodes and W channels with $N>W$. The interconnection of the nodes is managed through a passive star [17]. The passive star is a $N \times N$ device without power, so as signal that is inserted on a given wavelength from an input fiber port, has its power equally divided among all output ports, on the same wavelength. Each node is connected with the passive star via a two-way optical fiber.

Each node comprises a pair of a tunable transmitter (or an array of W fixed transmitters) and a fixed receiver. It is capable to transmit in all the available channels and is able to accept data in a specific only home channel. Of course, if a node n_1 wishes to transmit to another node n_2 tunes its laser to n_2's home channel. On the other side, if node n_1 is ready to receive data from the node n_2, a certain action is not required from the node n_1, knowing that it receives data exclusively in its home channel. Conclusively, N/W nodes share the same home channel, fact that can create a problematic situation, as a collision. More specifically, if two or more nodes try to transmit within the same wavelength simultaneously then a (channel) collision appears. For instance, if node n_1 and node n_2 try to transmit to the node n_3 at the same time, there is going to be a collision, since, the data travel simultaneously to the same home channel. So the packets are destroyed and there is a need for updating the nodes and retransmitting the signals, processes that are time consuming and complicated. Another way to avoid collision is the determination of a node selection policy. One data packet is selected, while the remaining ones have to be retransmitted. The transmission of the ''rejected'' data packets wastes bandwidth and the protocol performance is considerably degraded.

Transmission is organized into frames, where each frame consists of a reservation phase and a data phase. During the reservation phase of each frame taken to be N slots long, each source node is assigned a unique slot for broadcasting its control packet to all channels by means of its tunable transmitter (hence access is TDM based). Control packets are received by all nodes on their corresponding home channel by means of their fixed receiver and are assumed to make reservations for the data phase. For example, we consider a network with four nodes and two channels. During the reservation phase, each node sends their control packets (each control packet includes two important parameters: the number and the destination of the packets) to all channels. Moreover, we assume that the first node (n_1) and the second node (n_2) share the same home channel (w_1) and the third node (n_3) and the fourth node (n_4) share the other home channel (w_2). First of all, node n_1 starts to broadcast its control packet, using firstly the channel w_1 and secondly the channel w_2. At this point all the nodes know the control packet, transmitted by node n_1. After this, node n_2, node n_3, and node n_4 operate at the same way, using both the channels. The reservation phase finishes and each node knows the total requests for all the channels. So, each node calculates the (same) data, in order to produce the (same) schedule.

The protocol we examine here is a collision free, i.e., it secures the transmission of the data to all the nodes, without any collision. Of course, in order to achieve it, it requires a synchronization mechanism. For this purpose, the protocol maintains a distributed algorithm to all nodes. Each node has to maintain some global status information and to update it every now and again, according to information obtained by a shared control channel (or a shared data channel) [18]. The distributed algorithm accepts the transmission time demands of each node of the network and stores them in a matrix $D=[d_{i,j}]$ called traffic demand matrix. The matrix has N rows and W columns, as N is the number of the nodes and W is the number of the channels. Hence, the cell stored in the i (i belongs to N) row and j (j belongs to W) column contains the amount of time (usually in timeslots), which i node requests to transmit on the j channel, as j channel is the home channel of the destination node. As each frame starts, all nodes run the same distributed scheduling algorithm, based on the same information. So, the algorithm can be able to decide how transmissions and receptions should be made for the next phase. A typical form of a demand matrix D is presented:

$$D = \begin{matrix} 1 & 3 & 2 \\ 2 & 4 & 2 \\ 3 & 1 & 3 \end{matrix}$$

It is clear that in the specific example the network contains three nodes (three rows) and three channels (three columns). The first node requests one timeslot for the first channel ($d_{0,0}$), three timeslots for the second channel ($d_{0,1}$), and two timeslots for the third channel ($d_{0,2}$). With the same manner, the second node requests two timeslots for the first channel ($d_{1,0}$), four timeslots for the second channel ($d_{1,1}$), and two timeslots for the third channel ($d_{1,2}$). Lastly, the third node needs three timeslots for the first channel ($d_{2,0}$), one timeslot for the second channel ($d_{2,1}$), and three timeslots for the third channel ($d_{2,2}$). In the continuance of this work we will use the specific demand matrix to compare a set of scheduling algorithms. In order to make it easier for the synchronization and the transmission, we suppose that the time is divided in transmission frames. Each frame is composed of a reservation phase and a transmission frame. Also, each frame consists of a number of timeslots during which the reservation and the packet transmission take place. A list of scheduling algorithms for optical WDM networks can be found in Refs. [19–26].

3. Prediction-based online scheduling algorithms

There are many interesting scheduling algorithms that could be used to solve the schedule problem. Online interval-based scheduling (OIS) [19] is a simple and practical online scheduling algorithm. The algorithm contains only simple operations in order to construct the schedule very quickly. This scheme incorporates online scheduling on the basis of available time intervals on channels and for each examined node that requests reservation. As we said before, each node runs at the same time the same distributed scheduling algorithm. The entire demand matrix is not necessary; hence, the construction of the scheduling matrix begins immediately after the first set of requests (by the first node) is known. In order the algorithm to be able to function properly, each node maintains a list of time intervals that are available on every data channel. More specifically, the algorithm maintains two sets of intervals, one for each of the channels and another for the node whose reservation is currently being scheduled. The interval list per channel or current node represents the unallocated time on that channel or node. If we assume that node n_1 requests t_1 timeslots to transmit packets using channel w_1, the OIS algorithm searches for an available set of timeslots after time t_1 in order to reserve for this node. When OIS finds a suitable set of timeslots beginning at time t, it reserves channel w_1 from time t to $t+(t_1-1)$. Of course, the algorithm is not allowed to assign more nodes at the same interval, in order to keep the schedule collision free. The same idea is implemented for the requests of the rest nodes. It is obvious that OIS cannot reserve another channel for node n_1 that overlaps with the above reservations. If the algorithm concludes that the scheduled transmission does not result in any collisions, it includes it in the scheduling matrix that is being constructed. As a result, at any given timeslot, the request table (scheduling matrix) of OIS comprises the nodes that are scheduled to transmit and the wavelengths, which they will transmit in. The mathematical and implementation details are available for OIS [19]. We examined the main idea of OIS, whose operation becomes clear in Section 4 in which we consider an example.

The basic problem with OIS is the large amount of time, of the schedule computation period of each frame. As we mentioned before, each frame is composed by a reservation phase and a data phase. According to OIS, the transmission data have to wait for the algorithm to finalize the schedule for each frame. In order to decrease the delay that a ready node experiences while waiting for OIS to compute the schedule, predictive online scheduling algorithm (POSA) [20] attempts to eliminate the duration of the schedule computation process by predicting the nodes' requests for the next frame. In this direction, POSA makes use of a hidden Markov chain and after an initial learning period of several frames, POSA attempts to predict the requests of the nodes for the subsequent frame based on their requests for the previous frame. Because the algorithm does not wait for the nodes to send their requests in order to compute the schedule but starts working based on the predictions, a significant amount of time is saved. The predictor uses two different algorithms, i.e., the learning algorithm and the prediction algorithm. During each frame of data, the predictor first runs the learning algorithm and then the prediction algorithm. The first algorithm is responsible for informing and updating the data of the history queue, while the second one is responsible for predicting the demand matrix as accurately as possible. However, POSA brings about some performance improvement, if the average duration of the control and data phases is at least equal to the time needed for predicting reservations and computing the corresponding schedule. Finally, we must pinpoint that POSA uses the same algorithm as OIS to construct the scheduling matrix.

Check and sort-predictive online scheduling algorithm (CS-POSA) [21] is an extension of POSA. Its aim is to extend POSA, while maintaining the pipelining of the schedule computation and the full operation of the predictor. The extension of CS-POSA is based on shifting of the schedule computation of the nodes or in other words, on guiding the order of checking and programming of the nodes. POSA ignores the variety of the traffic among the nodes building the transmission scheduling matrix starting from the predicted requests of the first node, then the second one and so on until the last one. This is due to the fact that POSA uses OIS to construct the scheduling matrix examining one after the other the requests of the first to the last node. CS-POSA, on the contrary, does not always blindly follow the same service order, i.e., from the first to the last. It examines the cumulative workload, i.e., the sum of the requests of each node to all destinations and based on it, it processes them in a declining order. Shifting is based on the workload of each node, which means that the CS-POSA comprehends better not only the general traffic of the network but also the specific workload in each node. Before CS-POSA constructs the schedule matrix, it takes the two following steps. In the first step, CS-POSA adds each row of the traffic matrix D in a new vector S that will register the total amount of requests by each node. So, vector S consists of the total amount of the requests of the whole nodes for the whole transmission channels. In the second step, CS-POSA grades vector S in a declining order. In case those two nodes are found with the same total number of requests, then the selection is random.

4. Load eclectic-navigated algorithm (LENA)

In this section we present a novel scheduling technique called LENA. The new scheme changes the service order of each node based on the load of each channel demand in declining order. Simulation results show that if a change in the sequence of the service of the channels is made from the channel with the more requests to the channel with the fewer requests then a simultaneous improvement is observed both in the network throughput and the mean time delay in the waiting queues. All these are achieved with the algorithm LENA, which satisfies the criterion of scalability, since it keeps complexity at low levels.

The target of this work is to reduce the time required per transmission frame by decreasing the schedule length. In other words, the aim of the new technique is to reduce the time that occupies the idle timeslots. As idle timeslot, it is defined the timeslot that includes an inactive channel. It is obvious that the more idle timeslots in a schedule frame are, the more time the packets remain in the waiting queues and their transmission is delayed. On the contrary, an algorithm that forms a short schedule without (as far as it is possible) idle timeslots will result into quick transmissions without big time delays. Moreover, the number of the idle timeslots is inversely proportional with the performance of the network. The fewer idle timeslots are, the more packets are transmitted in the time unit. The typical metric of the efficiency of the schedule is the channel utilization that is the percentage of the demanded slots over the total slots. The fewer idle timeslots exist, the higher channel utilization is and therefore the higher the network throughput is. LENA algorithm must obey to three criteria:

1st Criterion: Simplicity of the operations. In order LENA algorithm to obey to this rule, the OIS scheduling algorithm is adopted as the basic schedule algorithm. So, LENA adopts from OIS [19] the simplicity of its algorithm and the absence of any complex procedure or manufacture. We must mention that OIS produces a transmission schedule in $O(KNW^2)$, where N is the number of nodes, W is the number of channels, and K denotes the upper bound on a node's request on a channel. It is clear that the time complexity of the OIS algorithm is linear with N.

2nd Criterion: Prediction-based system. LENA, similarly to POSA [20], works with the help of the prediction scheme. In order to reduce the time required per frame, LENA, as POSA, pipelines the schedule computation phase of the frame with the reservation and the data phases. So, LENA keeps an $N \times W$ demand predicted matrix at the end of the reservation phase. Having as a basis this predicted demand matrix, and not the actual demand matrix, the schedule matrix is finally constructed by the LENA algorithm.

3rd Criterion: LENA draws the third rule of the algorithm from CS-POSA [21]. As we discussed in Section 3, CS-POSA shifts the service order of the nodes, based on the total sum of load of each node. LENA adopts this feature with the same manner. So, before the construction of the final schedule matrix changes the service order of the nodes, based on the sum of the requests of each node.

In this paper, we further expand the previous work by presenting LENA, which is analyzed into two novel operations:

Novelty 1

After the finishing of the service sequence of the nodes, LENA prioritizes the requests of the channel with the greatest total of requests. The algorithm begins the manufacture of scheduling matrix with the channel that contains the most requests. That means that the algorithm does not choose the channels from the first one to the last one, but selects each time the channel with the most time requests for the specific node. If there are multiple channels with the same number of demands, then the algorithm chooses the one with the smallest channel number, if the number of the current frame is odd, and the one with the largest channel number, if the number of the current frame is even. In that way, the distributed algorithm chooses the same channel selection in each node, at the same time, providing fairness in the selections.

Novelty 2

LENA imports also a new improvement, which has to do with the selection of two or more nodes with the same sum requests. If a set of nodes has the same number of total requests (for all channels), then LENA selects the node that has the biggest number of request (the max request).

4.1. A simple numerical example

It is useful if a specific workload example is studied. Let us consider that the following demand matrix D has been predicted for the frame f by nine individual predictors:

$$D = \begin{matrix} 1 & 3 & 2 \\ 2 & 4 & 2 \\ 3 & 1 & 3 \end{matrix}$$

According to matrix D, node n_0 demands one transmission timeslot for the channel w_0 (the element $d_{0,0}$), node n_1 demands two transmission timeslots for the channel w_0 (the element $d_{1,0}$), and node n_2 demands three transmission timeslots for the channel w_0 (the element $d_{2,0}$). With the same manner, node n_0 demands three timeslots on w_1 and two timeslots on w_2. Node n_1 demands four timeslots on w_1 and two timeslots on w_2. Finally, node n_2 demands one timeslots on w_1 and three timeslots on w_2. It is clear that a set of 21 timeslots have been produced for all nodes for the current frame f.

To present the different scheduling approaches of each algorithm, three schedule matrixes are shown below one for each scheduling algorithm. First of all, let us check the scheduling method of OIS/POSA algorithm. The OIS/POSA algorithm starts the constructing of the final schedule by examining the demands of the first node, i.e., node n_0 and finishes with the examination of the least node, i.e., node n_2. Let $\sigma_{OIS/POSA}$ denotes the cell's examination sequence of OIS/POSA scheduling algorithm. So, OIS/POSA algorithm keeps the $\sigma_{OIS/POSA}$ sequence of the matrix D in order to form the final schedule:

$$\sigma_{OIS/POSA} = \{D_{0,0}, D_{0,1}, D_{0,2}, D_{1,0}, D_{1,1}, D_{1,2}, D_{2,0}, D_{2,1}, D_{2,2}\}$$

The final schedule matrix is shown in Fig. 2. From the final schedule matrix of OIS/POSA we can export the fact that the algorithm needs 13 timeslots to complete the transmission. At the same time, a part of 18 subtimeslots are unused, because the channel remains idle and waits for the next transmission demand. The percentage of the waste subtimeslots that is constructed by OIS/POSA algorithm for the specific demand matrix D is equal to $18/(3 \times 13)$ or 46%, so the algorithm utilization is equal to 54%.

Secondly, the scheduling method of CS-POSA is examined. CS-POSA operates in two steps. Firstly, it creates a sum vector S, which stores the cumulative load of each node and then it constructs the final schedule, by shifting the service order of the nodes based on the sorted vector S'. So, the vector S, as concerned the demand matrix D, has the same length as the number of the nodes and is

timeslots

	0	1	2	3	4	5	6	7	8	9	10	11	12
w_0	n_0	n_1	n_1	n_2	n_2	n_2	idle	idle	idle	idle	idle	idle	idle
w_1	n_2	n_0	n_0	n_0	n_1	n_1	n_1	n_1	idle	idle	idle	idle	idle
w_2	idle	idle	idle	idle	n_0	n_0	idle	idle	n_1	n_1	n_2	n_2	n_2

Fig. 2. The final schedule matrix constructed by OIS/POSA for demand matrix D.

shown below:

$$S = \begin{matrix} 6 \\ 8 \\ 7 \end{matrix}$$

The vector S is sorted and reformed to sorted vector S':

$$S' = \begin{matrix} 8 \\ 7 \\ 6 \end{matrix}$$

Vector S informs about the sum load of each node. More specifically, node n_0 has six timeslots totally, node n_1 has eight timeslots totally, and node n_2 has seven timeslots totally. CS-POSA changes the process order of the nodes, according to vector S' in declining order. Hence, node n_1 is processed first, followed by node n_2. Then node n_0 is served. In this manner the $\sigma_{\text{CS-POSA}}$ sequence of the matrix D for CS-POSA is equal to

$$\sigma_{\text{CS-POSA}} = \{D_{1,0}, D_{1,1}, D_{1,2}, D_{2,0}, D_{2,1}, D_{2,2}, D_{0,0}, D_{0,1}, D_{0,2}\}$$

Fig. 3 shows the final schedule matrix, constructed by CS-POSA. It is clear that CS-POSA constructs a shorter schedule than OIS/POSA with length of 11 timeslots. A shorter schedule offers a better transmission output due to the fact that it transmits the same data packets in lesser time. Of course, 12 subtimeslots remain idle and the schedule utilization is 64%.

LENA algorithm applies a different schedule technique. It keeps the altering of the nodes's service order, introduced by CS-POSA, and also takes under consideration the individual demand of each channel. For each node, LENA chooses the channel demand in a declining order, based on the time length of each load. Hence, the algorithm checks and serves first the channel with the biggest time demand and last checks and serves the channel with the least. With this way, LENA gains time space in the final schedule, by altering the channel demands and prioritizing the requests with more time demand. In other words, the demands with the most time space are examined first and allocate more free space (or more free timeslots) for the rest short time demands. LENA constructs the vector S like CS-POSA and then sorts the channel demands of each of the nodes. LENA starts the construction of the scheduling matrix, by examining the requests of node n_1, which has the most requests totally. Node n_1 asks for two timeslots for w_0 and w_2 and four timeslots for w_1. LENA sorts these three demands ($d_{1,0}$, $d_{1,1}$, $d_{1,2}$) and changes the process sequence of node n_1. In this manner, LENA chooses the demand on w_1 as the first choice to put in the schedule matrix, then chooses the demand on w_0 as the second choice (we consider that the number of the current frame is odd and the algorithm chooses the channel with the smallest channel number), and finalizes the requests of node n_1 with the third choice w_2. Fig. 4 presents the order of the choosing for node n_1. In the same manner, LENA continues with the set of demands, which belong to node n_2 (Fig. 5). Hence, the channel w_0 is selected (three timeslots) first, channel w_2 follows (three timeslots), and LENA finishes with node n_2 by selecting the demands of w_1 (one timeslot). The selection of the channel's demands for node n_2 is shown in Fig. 6. Finally, LENA finishes the construction of the scheduling matrix, by servicing the requests of node n_0. Again, channel w_1 (three timeslots) is selected first, channel w_2 (two timeslots) follows, and the last selection is the channel w_0 (one timeslot).

Hence, the σ_{LENA} sequence of the matrix D is

$$\sigma_{\text{LENA}} = \{D_{1,1}, D_{1,0}, D_{1,2}, D_{2,0}, D_{2,2}, D_{2,1}, D_{0,1}, D_{0,2}, D_{0,0}\}$$

timeslots

	0	1	2	3	4	5	6	7	8	9	10
w_0	n_1	n_1	n_2	n_2	n_2	n_0	idle	idle	idle	idle	idle
w_1	n_2	idle	n_1	n_1	n_1	n_1	n_0	n_0	n_0	idle	idle
w_2	n_0	n_0	idle	idle	idle	idle	n_1	n_1	n_2	n_2	n_2

Fig. 3. The final schedule matrix constructed by CS-POSA for demand matrix D.

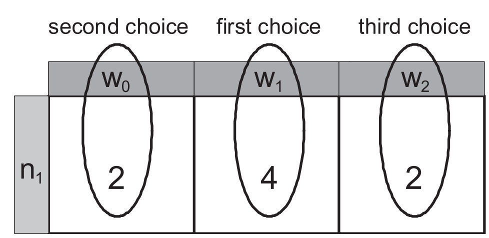

Fig. 4. The order of channel's choosing of LENA algorithm for node n_1.

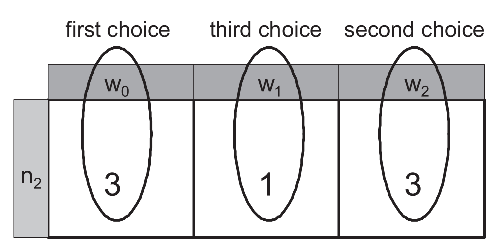

Fig. 5. The order of channel's choosing of LENA algorithm for node n_2.

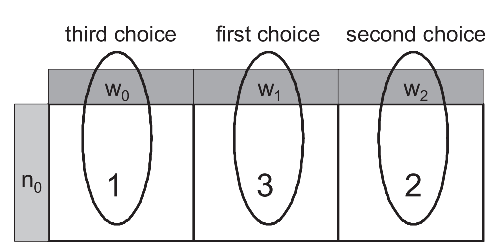

Fig. 6. The order of channel's choosing of LENA algorithm for node n_0.

	timeslots 0	1	2	3	4	5	6	7	8	9
w_0	n_2	n_2	n_2	n_0	n_1	n_1	idle	idle	idle	idle
w_1	n_1	n_1	n_1	n_1	idle	idle	n_2	n_0	n_0	n_0
w_2	n_0	n_0	idle	n_2	n_2	n_2	n_1	n_1	idle	idle

Fig. 7. The final schedule matrix constructed by LENA for demand matrix *D*.

The final scheduling matrix is shown in Fig. 7. LENA constructed a schedule scheme that composes 10 only timeslots, the shortest from each of the previous schemes. Apart from this, LENA leaves only nine subtimeslots idle, so the total percentage of utilization is 70%.

4.2. Complexity analysis of LENA

It is very important for the design of the algorithm, especially in optical networks, to run in well-bounded asymptotic time. The algorithm is assigned to optical network and so it has to be very quick with simple only operations. Furthermore, the algorithm has to be scalable and capable to operate normally regardless of the network changes. In other words, the algorithm has to be independent of the number of the nodes. The running time of LENA is dependent only on the sorting of the channel's load for each node. This is because LENA uses the same scheduling algorithm as OIS and it adopts the same exactly prediction mechanism of POSA. So, the extra complexity of LENA occurs from the sorting of the service order of the channel's loads. Therefore, the extra complexity of one node is $O(W\log W)$ due to the sorting of W elements. Additionally, in that number we have to calculate the number of nodes, which is N. For the entire process of LENA the complexity is equal to $O(N\log NW\log W)$, i.e., lower than the complexity of the basic scheduling algorithm of OIS, which is $O(NW^2K)$. Also, if there are P processors running the algorithm, the overall complexity of the algorithm becomes $O((N\log NW\log W)/P)$, where P is a function of NW, $P=(NW)/q$, where q is constant and is dependent on the number of the processors. The above analysis suggests that LENA is scalable with linear growth of N. Furthermore, the entire algorithm runs in constant asymptotic time, as N increases and the only restrictive factor is the $q\log W$ number, which is negligible, due to the short number of channels (the number of channels has an upper limit).

5. Performance measures

In the following section, there is a presentation of the performance of three prediction-based algorithms. POSA, CS-POSA, and LENA are executed, in order to compare their efficiency, in terms of channel utilization, network throughput, and mean packet waiting time (in queues) under uniform traffic. In the simulation model the following approximations are introduced:

a. Tuning time is ignored (for simplicity reasons).
b. Traffic pattern is uniform, i.e., data requests are destined to every node with equal probability.
c. For each frame, nodes may generate data requests from 0 to K with equal probability.
d. The line is defined at 2.4 Gbps per channel.

The three algorithms are executed in a real-time *C* environment for 10,000 frames. The first 1000 are learning frames, during which all three algorithms operate as OIS, without any prediction action. The simulation parameters varied are N, number of nodes; W, number of channels; and K, maximum value over all entries in the demand traffic. Each entry in the matrix is a random number between 0 and K. The values range between 0 and K and in order the goal of scalability to be achieved, the value of K is not constant in the following experiments but each time it is equal to

$$K=\left\lfloor\frac{NW}{5}\right\rfloor$$

In the analysis of the three algorithms, common measures and measurements have been used and are presented below:

– Schedule length is symbolized by L and denotes the number of slots in the data phase as determined by the schedule algorithm.
– Total slots requested by all nodes are symbolized by R and denotes the total number of timeslots that were requested by all the nodes of the network.
– Schedule or channel utilization is symbolized by U and denotes the number of slots actually utilized for packet transmission in a schedule matrix. Scheduling utilization is defined as

$$U=\frac{\text{Totalslots requested}}{\text{Schedule length}\times\text{channels}}\quad\text{or}\quad U=\frac{R}{LW}$$

– The transmission rate of each channel is symbolized by S and has been set in 2.4 Gbps.
– Throughput is symbolized by Γ and denotes the average number of bits transmitted per transmission frame per channel. Since the three algorithms, which are examined, do not waste computation delay due to pipelining throughput, so the relation becomes

$$\Gamma=\frac{R}{LW}S\quad\text{or}\quad\Gamma=US$$

– Delay is symbolized by D and denotes the mean time delay of the transmitted data in timeslots. It equals the number of timeslots that pass from the moment that a packet with data is produced in the queues until the moment that the transmission starts. If, for example, a packet with data has been produced at the time moment t_1 and in the schedule matrix it has been set to be transmitted at the time moment t_2, where $t_2 - t_1 = t$ timeslots, then $D = t$.

6. Simulation results

This section presents the simulation results. Three algorithms, which utilize the prediction method, i.e., POSA, CS-POSA, and LENA, have been studied and compared in the context of channel utilization, network throughput, throughput delay as the number of nodes varies, throughput delay as the load of the network varies and throughput load, under uniform traffic. The objectives of the simulation are twofold. First, the superior of the suggested, LENA protocol, is presented by showing that LENA works much better than POSA and CS-POSA, in terms of channels utilization and network throughput. Second, the superior performance of LENA is demonstrated by showing that the mean waiting time in the queues before the transmission is reduced. In the results of the simulation, it is assumed that N is the number of nodes, W is the number of the channels, and K is the maximum value over all entries in the traffic matrix. In other words, K is the maximum value of demand timeslots for each pair of node channel. The speed of the line has been defined at 2.4 Gbps. Also, it should be mentioned that the tuning latency time is considered to be equal to zero timeslots for simplicity reasons.

6.1. Channel utilization results

In the following experiments, we compare the three algorithms in two network models. The first model comprised five transmission channels and the second one comprised 10 transmission channels. The numbers of nodes are {10, 20, 30, 40, 50, and 60}. In Fig. 8 the channel utilization of the three algorithms is shown for the six values of nodes and five channels. It is observed that LENA algorithm is superior to POSA and CS-POSA and it has maximum difference over POSA equal to 5.42% for 10 nodes and minimum difference equal to 1.08% for 60 nodes. Also, the maximum difference over CS-POSA reaches to 3.8% for 10 nodes and the minimum difference is found for 60 nodes and reached to 0.85%. In Fig. 9 the channel utilization is presented for 10 channels. Again, LENA stands better than POSA and CS-POSA for the values of nodes. The maximum difference between LENA and POSA comes up to 9.55% for 10 nodes and the minimum comes up to 2% for 60 nodes. LENA overcomes CS-POSA up to 7.68% for 10 nodes at maximum and up to 1.15% for 60 nodes. Overall, LENA has the best performance, regarding the channel utilization, because it operates a sorting to the values of the channels for each node, before the schedule construction. This action offers a better exploitation of the available channels, by reducing the appearing of idle timeslots.

6.2. Network throughput results

Fig. 10 shows the network throughput of the three compared algorithms for five channels. Again, the examined numbers of nodes are {10, 20, 30, 40, 50, and 60}. The throughput improvement with LENA algorithm proves that its scheduling length is shorter than POSA and CS-POSA. It also explains that LENA allows data to be transmitted much better at the same time than POSA and CS-POSA. In particular, LENA improves the network throughput at least to 133 Mbps and at most 666 Mbps, as compared to POSA and at least to 104 Mbps and at most 470 Mbps, as compared to CS-POSA. In Fig. 11 the throughput comparison between the three algorithms is

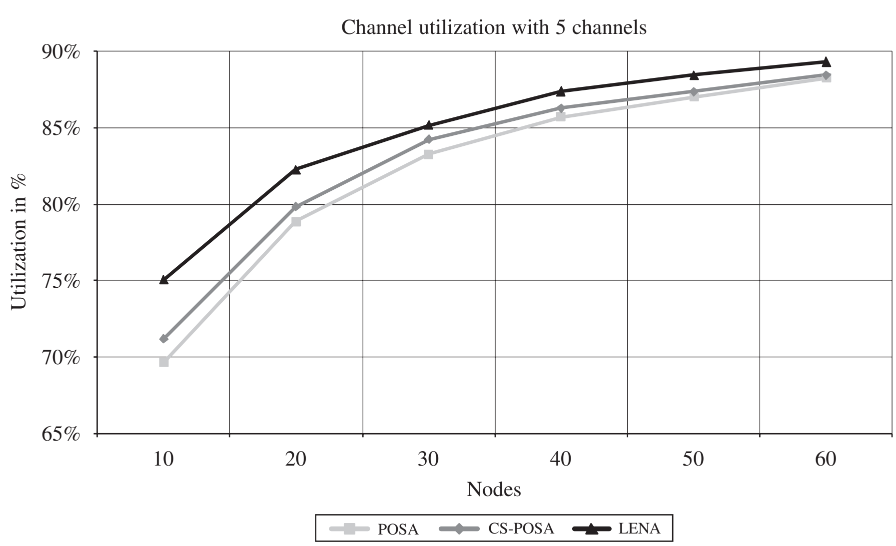

Fig. 8. Channel utilization for five channels.

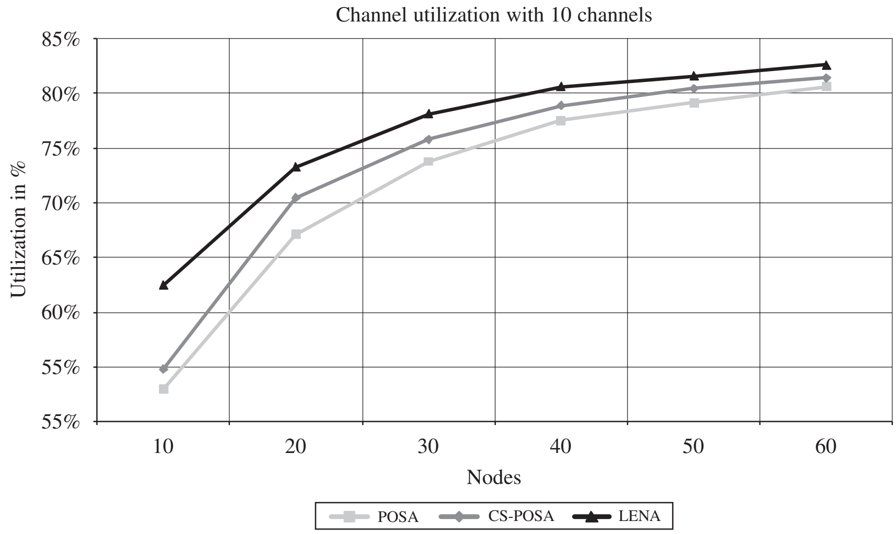

Fig. 9. Channel utilization for 10 channels.

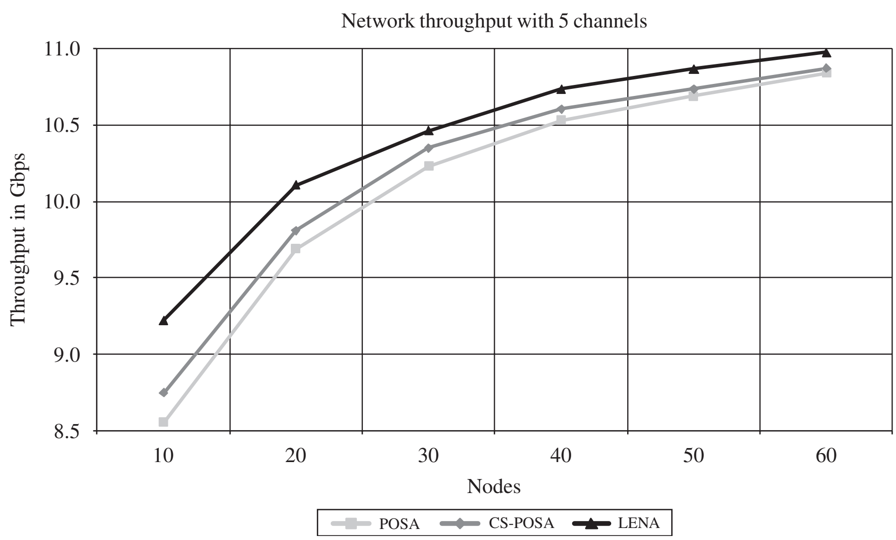

Fig. 10. Network throughput for five channels.

plotted with 10 channels. Again, LENA keeps the superiority for each value of node. Here, the maximum difference reaches to 2.3 Gbps, as compared with POSA and to 1.9 Gbps, as compared with CS-POSA. The minimum differences are 277 Mbps and 484 Mbps for POSA and CS-POSA, respectively.

6.3. *Network throughput versus mean waiting time as nodes vary*

Fig. 12 depicts the network throughput versus mean waiting time with five channels and Fig. 13 illustrates the network throughput versus mean waiting time with 10 channels. The results show the relation between network throughput and delay as the nodes vary. The set of tested numbers of nodes are {10, 20, 30, 40, 50, and 60}. It is obvious that LENA excels in network throughput for any combination of nodes and channels. At the same time, LENA keeps delay lower, not much but lower, than POSA and CS-POSA. For example, when $N = 30$ and $W = 5$, the produced network throughput by LENA is 10.5 Gbps with 189 timeslots as mean waiting time in the queues, while POSA produces 10.2 Gbps with 204 timeslots and CS-POSA offers 10.3 Gpbs with 206 timeslots. One more example is given, when $N = 20$ and $W = 10$. LENA outputs 18 Gbps and the packets wait 166 timeslots, while POSA offers 16.4 Gbps with 174 delay slots and CS-POSA produces 17.3 Gbps with 178 slots. It is clear from both figures that LENA offers better throughput with lower delay, because it serves firstly the high-demanding requests for each node. In this way, the time long packets wait less and the mean wait time is reduced a little.

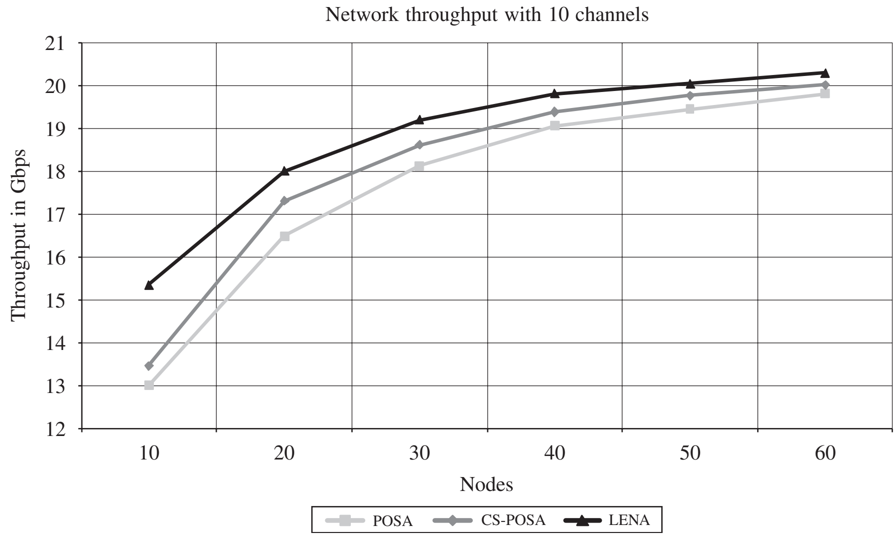

Fig. 11. Network throughput for 10 channels.

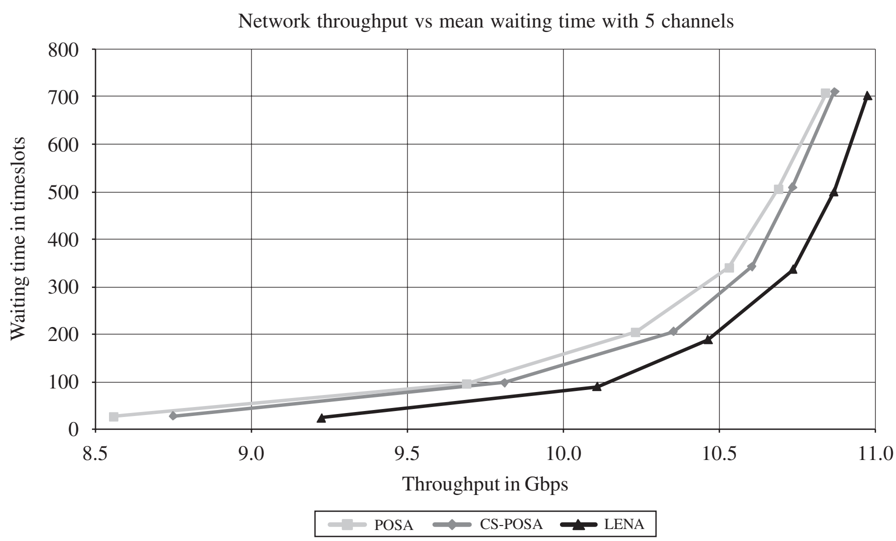

Fig. 12. Network throughput versus mean time delay as nodes vary for five channels.

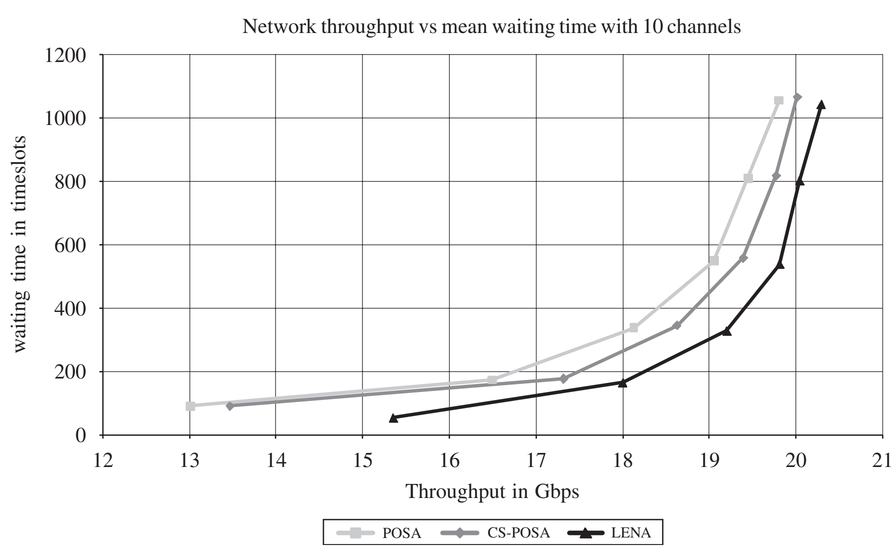

Fig. 13. Network throughput versus mean time delay as nodes vary for 10 channels.

6.4. Network throughput versus mean waiting time as K varies

Once again, the relation of throughput with delay is analyzed on two specific network models. The first one includes 30 nodes and five channels, while 10 different values of K (10, 20, 30, 40, 50, 60, 70, 80, 90, and 100) are inserted to the three algorithms. The second one includes 30 nodes and 10 channels and the same set of values for K. We can see from both these two figures (Fig. 14 for 5 channels and Fig. 15 for 10 channels) that the throughput with all algorithms drops down, as K increases. This phenomenon is not often met in the category of the networks examined. Nevertheless, it appears in algorithms examined, OIS [19], POSA [20], or CS-POSA [21], owing to the architecture of the protocols. When the workload is increased, it means that the sizes of packets that arrive at the nodes in order to be transmitted are actually increased. This is denoted with the increase of the maximum value of K. When K is increased, it is difficult for the scheduling algorithm to find open space in the constructed scheduling matrix. If there was an open space of nine slots in the constructed scheduling matrix and the packet arrived was of 10 timeslots size duration of transmission, then the algorithm cannot break it in pieces. It then places it at the end of the matrix where there is available space for a packet of 10 timeslots. This leads to a decrease of the channel utilization as the unused timeslots are increased and the throughput is decreased. Nevertheless, LENA keeps again higher throughput and lower delay for any value of load. For example, when $K = 80$ and $W = 10$, LENA produces 1.07 Gpbs more than POSA and

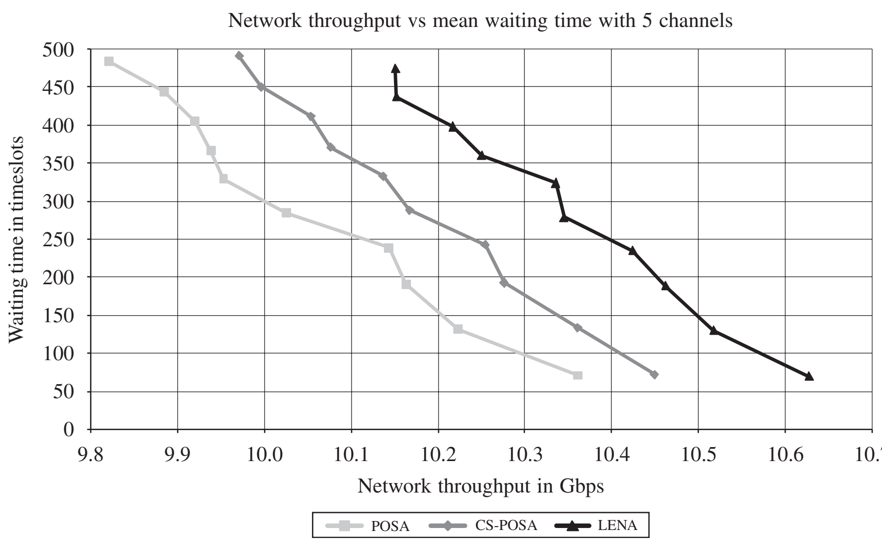

Fig. 14. Network throughput versus mean time delay as K varies for five channels.

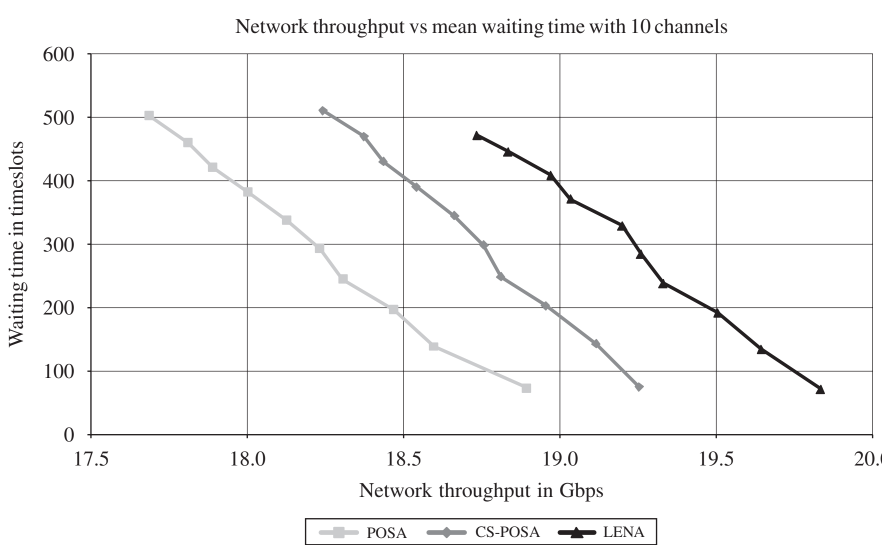

Fig. 15. Network throughput versus mean time delay as K varies for 10 channels.

0.46 Gpbs more than CS-POSA, while LENA reduces the waiting slots by 12 timeslots compared to POSA and 21 compared to CS-POSA.

6.5. Network throughput versus traffic load

Finally, the relation between throughput and traffic load, which is expressed with the metric of K, is plotted in Fig. 16 with five channels and in Fig. 17 with 10 channels as K varied from 10 to 100 (10, 20, 30, 40, 50, 60, 70, 80, 90, and 100). The traffic load increases as maximum K increases. K denotes the maximum value of demand timeslots for each pair of node-channel and so expresses the traffic load of each node. Heavy traffic means that nodes have high transmission demands, while low traffic means that nodes have low transmission demands. The traffic parameter, which expresses this relation, is the maximum value of K. Overall, LENA has the best output throughput for any K and for any W. The maximum difference with POSA seems to appear when $K = 60$ and $W = 10$ and it is 1.072 Gbps and the maximum difference with CS-POSA seems to be 580 Mbps when $K = 10$ and $W = 10$.

7. Conclusions

There has been a presentation of the design of a collision-free scheduling algorithm. The new scheduling technique refers to optical WDM networks in local area with a broadcast and select architecture. The proposed protocol innovates by altering the process order of the channels for each node, based on the load that each channel carries. The scheduling algorithm of OIS is adopted and the prediction mechanism of POSA is used.

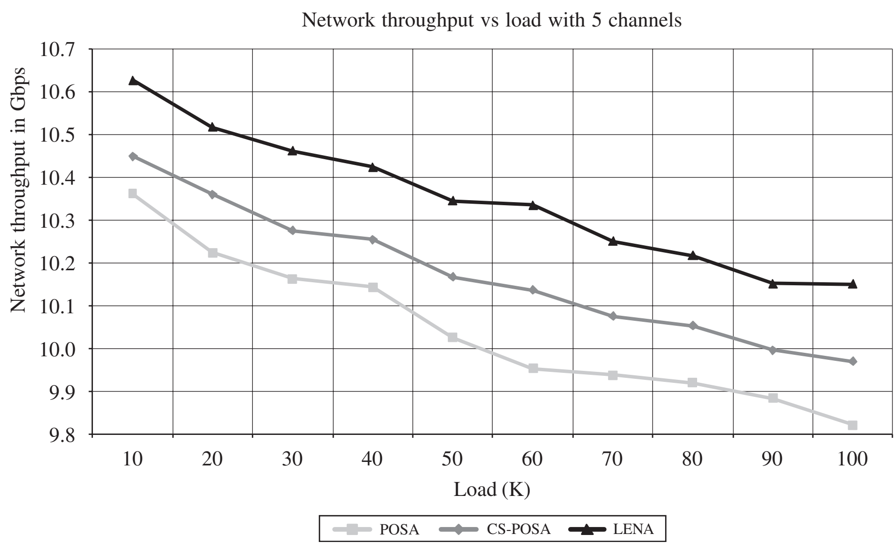

Fig. 16. Network throughput versus load for five channels.

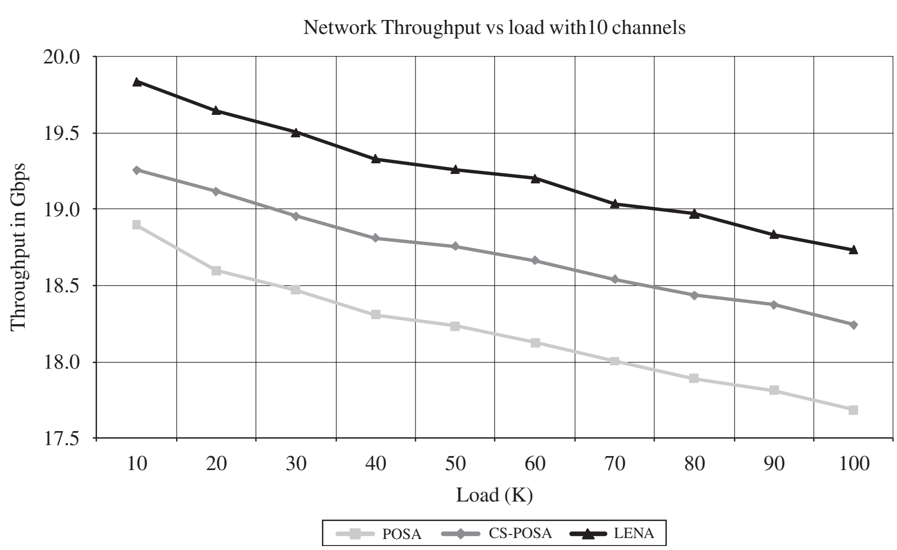

Fig. 17. Network throughput versus load for 10 channels.

The time complexity of the suggested algorithm increases in linear with the number of the nodes, resulting a scalable protocol. The algorithm manages to improve the channel utilization and the network throughput and at the same time it seems to decrease a little the time that the packets spend in the waiting queues. The performance improvement of LENA was shown by extensive simulation results under uniform traffic and a line speed of 2.4 Gpbs per channel.

References

[1] Papadimitriou GI, Papazoglou Ch, Pompotrsis AS. Optical switching: switch fabrics, techniques, and architectures. IEEE/OSA J Lightwave Technol 2003;21(2):384–405.

[2] Yao S, Mukherjee B, Dixit S. Advances in photonic packet switching: an overview. IEEE Commun Mag 2000:84–94.

[3] Green PE. Fiber optic networks. Eaglewood Cliffs, NJ: Prentice Hall; 1993.

[4] Acampora AS, Karol MJ. An overview of lightwave packet networks. IEEE Network Mag 1989;3:29–41.

[5] Mukherjee B. WDM optical networks: progress and challenges. IEEE J Selected Areas Commun 2000;18(10):1810–24.

[6] Murthy CSR, Gurusamy M. WDM optical networks: concepts, design, and algorithms. Prentice Hall PTR; 2002.

[7] Tsukada M, Keating AJ. Broadcast and select switching system based on optical time-division multiplexing (OTDM) technology. IEICE Trans Commun 1999;E82-B(2):335–43.

[8] Chen MS, Dono NR, Ramaswami R. A media-access protocol for packet-switched wavelength-division metropolitan area networks. IEEE J Selected Areas Commun 1990;8(6):1048–57.

[9] Brackett CA. Dense wavelength division multiplexing network: principles and applications. IEEE J Selected Areas Commun 1990;8: 948–64.

[10] Green P. Progress in optical networking. IEEE Commun Mag 2001; 39(1):54–61.

[11] Papadimitriou GI, Tsimoulas PA, Obaidat MS, Pomportsis AS. Multiwavelength optical LANs. New York: Wiley; 2003.

[12] Habbab IMI, Kavehrad M, Sundberg CW. Protocols for very high-speed optical fibre local area networks using a passive star topology. IEEE/OSA J Lightwave Technol 1987;LT-5(12):1782–94.

[13] Papadimitriou GI, Miliou AN, Pomportsis AS. OCON: an optically controlled optical network. Comput Commun 1998;22:811–24.

[14] Papadimitriou GI, Miliou AN, Pomportsis AS. Optical logic circuits: a new approach to the control of fibre optic LANs. In: Proceedings of IEEE 23rd annual conference on local computer networks (LCN'98), Boston, MA; October 1998. p. 326–35.

[15] Stern TE, Bala K. Multiwavelength optical networks. Reading, MA: Addison-Wesley; 1999.

[16] Mukherjee B. Optical WDM networks. Berlin: Springer; 2006.

[17] Wang B, Hou J, Han C. Design and analysis of a channel access protocol for WDM-based single-hop optical networks. Photonic Network Commun 2003;6(3):289–308.

[18] Diao J, Chu PL. Packet rescheduling in WDM star networks with real-time service differentiation. IEEE J Lightwave Technol 2001; 19(12):1818–28.

[19] Sivalingam KM, Wang J, Wu J, Mishra M. An interval-based scheduling algorithm for optical WDM star networks. Photonic Network Commun 2002;4(1):73–87.

[20] Johnson E, Mishra M, Sivalingam KM. Scheduling in optical WDM networks using hidden Markov chain based traffic prediction. Photonic Network Commun 2001;3(3):271–86.

[21] Sarigiannidis PG, Papadimitriou GI, Pomportsis AS. CS-POSA A high performance scheduling algorithm for WDM star networks. Photonic Network Commun 2006;11(3):209–25.

[22] Sarigiannidis PG, Papadimitriou GI, Pomportsis AS. A high throughput scheduling technique, with idle timeslot elimination mechanism. IEEE/OSA J Lightwave Technol 2006;24(12):4811–27.

[23] Sarigiannidis PG, Papadimitriou GI, Pomportsis AS. A high performance scheduling priority scheme for WDM star networks. IEEE Commun Lett 2007;11(1):76–8.

[24] Ma M. A cost-based scheduling algorithm for differentiated service on WDM optical networks. IEEE Commun Lett 2003;7(9):460–2.

[25] Sue C-C, Kuo S-Y. Design and analysis of accelerative preallocation protocol for WDM star-coupled networks. IEEE/OSA J Lightwave Technol 2002;20(3):338–49.

[26] Xu J, Qiao C, Li J, Xu G. Efficient burst scheduling algorithms in optical burst-switched networks using geometric techniques. IEEE/OSA J Selected Areas Commun 2004;22(9):1796–811.