

A novel medium access control protocol for optical local networks based on data request ordering

P.G. Sarigiannidis, G.I. Papadimitriou*, A.S. Pomportsis

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

Received 12 September 2006; received in revised form 14 February 2007; accepted 14 February 2007

Available online 6 April 2007

Abstract

The basic aim of the scheduling algorithms in wavelength division multiplexing (WDM) single-hop optical networks is to construct a short schedule, in order to support quick communication between the nodes of the network. A design of medium access control (MAC) protocol without collisions is introduced for scheduling variable-length data packets based on a broadcast and select architecture. The system includes only data channels and the coordination of the transmissions is achieved via control packets, functioning before the beginning of actual transmissions. The proposed scheme adopts a prediction mechanism in order to eliminate the possible delay introduced by the scheduling computation between the control and data phases of each cycle of transmission. The two common scheduling strategies suggested try to reorder the service sequence of the nodes, by prioritizing the nodes with long data packets, compared with the nodes with short data packets. The final schedule, formed by a scheduling matrix, seems to be much shorter and leads to an increase, in terms of channel utilization. The extensive simulation results show that the novel scheduling techniques offer more free time space for schedule and allow more data to travel on the medium for the same amount of time. Also, the mean packet delay on the queues is reduced and the relation throughput-delay is much better.

© 2007 Elsevier Ltd. All rights reserved.

Keywords: Optical WDM networks; Service sequence; Scheduling

1. Introduction

The most important target of the communication systems is to provide the users of the network access to the transferred information when they need it, where they need it and in whatever format they need it [1]. In order to achieve this aim, three key technologies compete each other to cover the constantly rising needs of the users. These are the copper technology, which requires copper as the main transfer medium, the wireless technology, which does not require of course any cable connection, and the optical technology [2–5]. As the network capacity increases, electronic switches nodes seem unable to keep up. Apart from that, electronic equipment is strongly dependent on the data rate and protocol, and thus, any system upgrade

results in the addition and/or replacement of electronic switching equipment. Realizing that the maximum rate at which an end-user access the network is limited by electronic speed (to a few Gb/s), the key in designing optical communication networks in order to exploit the fiber's huge bandwidth is to introduce concurrency among multiple user transmissions into the network architectures and protocols. Consequently, copper technology seems unable to cope with the needs of modern communication era, while wireless technology goes everywhere, but involves limited communicative channels with heavy loss due to the distance between the nodes. The optical networks are the most modern means of transmission, offering huge bandwidth in order to cover the applications which really demand high-speed data transmission such as tele-conference video, video phone, images for medical purposes, three-dimensional representation in real time, multimedia distribution, network games, etc. [6]. Generally

*Corresponding author.

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

speaking the advantages of the optical fiber as a transmission medium are as follows [7]:

1. huge bandwidth (over 50 Tb/s),
2. immunity to electromagnetic interference,
3. absence of crosstalk and interferences between fibers in the same cable,
4. low signal attenuation (as low as 0.2 dB/km),
5. ease in the installment and in the maintenance, and
6. high security of signal because of no electromagnetic radiation.

The realization of all these remarkable features of the optical fibers means that the need for division of the huge capacity of the medium into multiple communication lines in an effective way is instant. The wavelength division multiplexing (WDM) divides the optical frequency spectrum into channels (or wavelengths). The transmission channels are efficiently separated from each other, so that there is no interference. The most significant benefit of the WDM technique is its compatibility with the limited speed of the electronic circuits that each node possesses. If we consider that the maximum speed at which a node has access to the network is limited by the electronic technology to some Gb/s, the WDM technology offers a magnificent way of exploiting the huge bandwidth of the optical fibers, since it allows multiple users to transmit data simultaneously to different channels at electronically feasible transfer speed. It is a fact that if we consider a network with 500 transmission channels with 2.4 Gbps transmission speed each, then the optical fiber covers a total network speed equal to 1200 Gbps. A very convenient and effective way of realizing a local WDM optical network is the architecture of broadcast and select, according to which the connection of each node with each other is achieved through an optical passive star [8–10]. A remarkable attribute of this class of networks is that they can easily provide broadcast, multicast, and unicast services. This is achieved in a straightforward way just by having all, some, or one desirable destination node of a broadcast, multicast, and unicast transmission on some wavelength tune their receivers to this wavelength. It should be noted, also, that in the general case of a

broadcast and select network the total number of nodes can be greater than the number of wavelengths. This advantage gives flexibility and simplicity to the broadcast and select networks. Finally, a variation of the star configuration allows us to connect individual nodes with each other directly as well as through a central node. This is an expensive but powerful variation that can handle enormous traffic. Thus, a likely form that the overall network might take in the future is a combination of broadcast and select LANs interconnected by a wavelength routing network. According to broadcast and select architecture, each node transmits its data to the star coupler through one of the available channels, using a source of light as a laser. Each transmitted information from all the nodes of the network is combined in the star and sent forward to all the nodes of the network through a two-way optical fiber. The acceptance of the data is achieved through optical filters, which select the desired data sent by the star. The passive star is a broadcast device, so a signal that is inserted on a given wavelength from an input fiber port will have its power equally divided among all output ports on the same wavelength. Each node has at least one transmitter and one receiver. Both the transmitter and the receiver can be either fixed or tunable. The tunable transmitter can be tuned in any channel of the network in order to transmit the flow of data. Respectively, the tuned receiver can accept data from any channel of the network. On the other side, the fixed transmitter and the fixed receiver can send or receive data only in a predetermined transmission channel.

Two basic approaches have been proposed for the architecture of a broadcast and select optical network: the single-hop approach and the multi-hop approach. Unless there are intermediate nodes between every possible source–destination pair in the network, the WDM network is called single-hop. On the contrary, if a signal has to pass through intermediate nodes then the broadcast and select is said to be multi-hop. Fig. 1(a) shows the physical and Fig. 1(b) shows the virtual topology of a single-hop broadcast and select network with three nodes and three available channels. Fig. 1(a) illustrates a full physical connection through the optical fibers of all the nodes of the network. Every node communicates with every possible

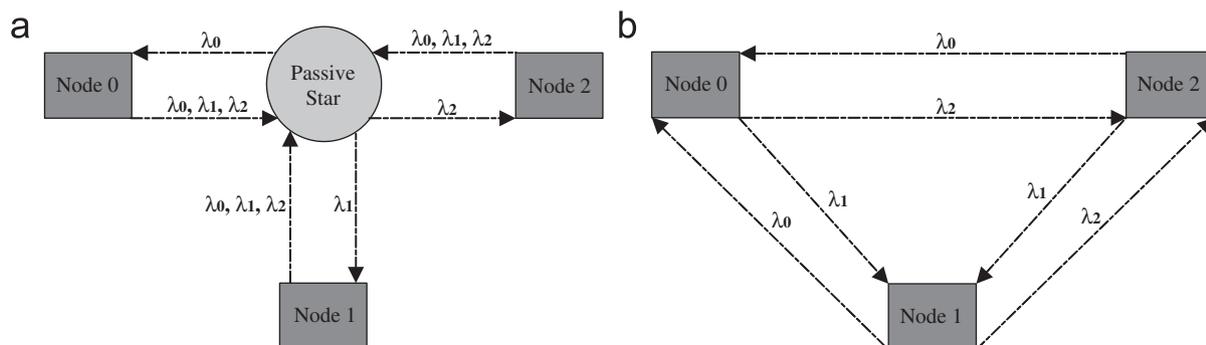


Fig. 1. An example of a single-hop network with three nodes and three channels.

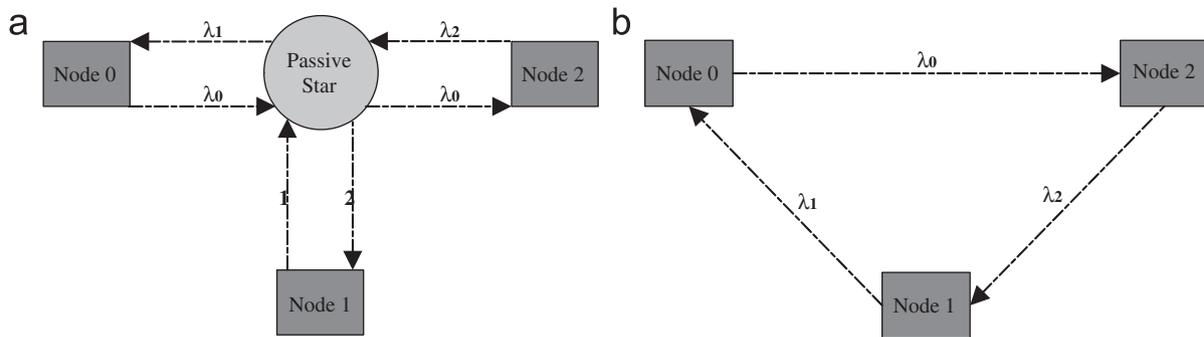


Fig. 2. An example of a multi-hop network with three nodes and three channels.

destination node in a single hop. For example, node 0 communicates with node 1 via channel λ_1 and with node 2 via channel λ_2 . In the second approach (Fig. 2), there is no full physical connection of all the nodes. However, the data packets travel through intermediate nodes till they reach their final destinations. Fig. 2(a) illustrates the physical topology of a multi-hop WDM broadcast and select network including three nodes and three channels. Fig. 2(b) shows the virtual interconnection between the nodes via different interconnected channels. It is obvious, for example, that node 0 can transmit to node 1 only via node 2, using channels λ_0 and λ_2 . It is clear that when a signal has to pass through intermediate nodes, it has to be converted into electronic form at each intermediate node, and then back to optical form, before it is routed through the network toward the intended final receipt. In contrast, in single-hop networks, the light signal remains in optical form all the way, from one end to the other; this implies high speed and data transparency. Also, single-hop systems can accommodate a large and time-varying user stations and they can properly utilize the available channel capacity [5]. In this context, single-hop broadcast and select networks are quite appropriate solutions, since tunable laser transceivers with fast tuning speeds and tuning ranges have been available [11,12] and so this paper focuses on the single-hop approach.

Indubitably, the basic aim of a medium access control (MAC) protocol is to define the total set of rules according to which the broadcasting and the selecting functions are managed and executed. In other words, the protocol coordinates the transmissions of the nodes in the network and when necessary it defines how a node should choose one among many simultaneous transmissions sent to it at a definite time. Obviously, when the transmitters are tunable, it is possible that two or more nodes will attempt to transmit to the same channel simultaneously. Such an event is called channel collision and as a consequence causes the destruction of the data and their retransmission. In case each node possesses only one tunable receiver and two or more nodes transmit to the same receiver-node at the same time in different channels, there is an acceptable (receiver) collision. A MAC protocol usually aims at the avoidance of channel collisions and the definition of procedures on which the selection of transmissions is

based. The WDM optical protocols are generally characterized as pre-transmission coordination-based protocols and as pre-allocation-based protocols. These are protocols that require the use of at least one channel for control and coordination, called pre-transmission coordination based. Examples of this category are [13–15]. When a control channel is not used, the protocols use channel pre-assignment. Some representatives of the pre-allocation-based protocols appear in [16–18]. However, it is very important an exception to the above general definitions to be mentioned, that is, a comparatively small set of pre-transmission coordination-based protocols that do not entail any separate control channel. The coordination takes place with control packets that are transmitted over the same channels used for data. Such samples are the protocols [19–24]. Comparing the two major protocol categories, we can say that pre-allocation-based protocols are generally much simpler in operation and more easily implemented but come behind in efficiency. On the other hand, pre-transmission-based schemes are often capable of utilizing the available bandwidth more efficiently and lead the network to higher performance levels than the pre-allocation-based schemes. Moreover, pre-transmission coordination-based MAC protocols offer more flexibility than pre-allocation schemes, when a large and time-varying number of nodes is considered and they constitute the most preferred choice made by a vast number of researches, including this paper.

The proposed protocol refers to a WDM single-hop optical LAN with broadcast and select topology (Fig. 3). Each node is equipped with one tunable transmitter and a fixed receiver. So, the system to be studied is a TT–FR system. Moreover, the protocol under discussion belongs to the category of pre-transmission coordination-based protocols with no control channel. There are no receiver collisions due to TT–FR system and channel collisions are prevented due to the construction of a suitable time schedule. In our system, the construction of the schedule is taken over by a schedule algorithm distributed in each node. Coordination is achieved through the update of a general global information which is updated to all the nodes every now and again. The form of this information is a traffic demand matrix $D = [d_{i,j}]$, where $d_{i,j}$ represents the number of data packets at node i that are destined for

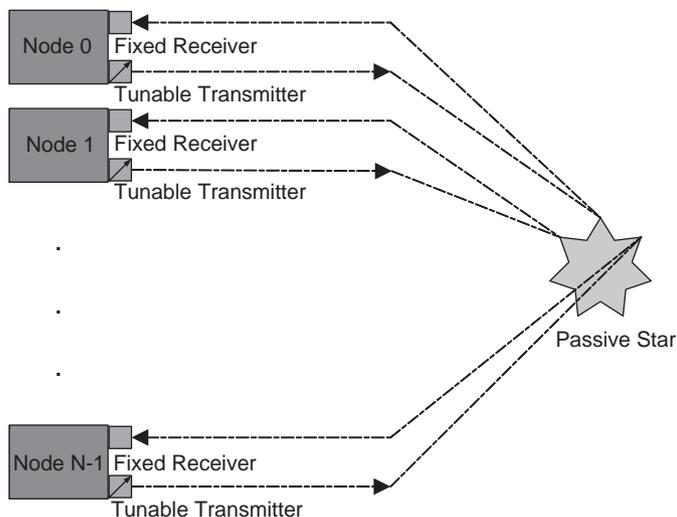


Fig. 3. The network architecture.

channel j ($0 \leq i \leq N-1, 0 \leq j \leq W-1$). Apparently, table D contains the same number of N lines as nodes, and the same number of W columns as channels. Transmissions take place within distinct transmission frames or cycles, where each frame consists of two phases, the reservation phase and the data phase. During the reservation phase, each node sends the number of packets that should be transmitted and the demand matrix D is formed. During the data phase, the transmissions of the packets are realized according to the schedule matrix constructed by the scheduling algorithm. For reasons of coordination, the time of each transmission frame is divided into timeslots. Each node generates variable-length data streams (messages), containing one or more fixed-length data packets. We assume that the basic transmission time unit is the time needed to transmit one data packet. In other words, each data packet needs exactly one timeslot to be transmitted.

A hybrid protocol (referred to as FatMAC) combining the advantages of pre-allocation- and reservation-based protocols has been proposed in [6]. Transmission in FatMAC is organized in frames, where each frame is composed of a reservation phase followed by a data phase. It uses time division multiplexing to provide access during the reservation phase. This protocol lacks in the sense that each node may reserve access to exactly one channel in the data phase. An extension of FatMAC, denoted as HRP/TSA (hybrid reservation pre-allocation/timeslot assignment), was suggested in [19]. This work aimed to improve the performance of FatMAC by allowing a node to reserve access to more than one channel during the data cycle. Thus, at the end of the reservation phase, the transmission requests of the nodes can be represented by a traffic demand matrix. In the approach using HRP/TSA, the schedule is computed immediately after the first node's reservation request is received. This pipelining allows overlap of the schedule computation and the reception of reservations and also reduces overall delay between the end of the reservation phase and the start of the data phase.

Since the reservations are broadcast to all the nodes, the nodes execute the same algorithm independently and generate identical schedules. The matrix decomposition principle of HRP/TSA leads to inefficiency in terms of schedule length.

In this paper a new access protocol is presented, named interval-based orderly scheduling strategy (IOSS), which is an improvement of a previous scheduling algorithm suggested in [19]. The new scheme proposes a new scheduling technique in order to achieve a shorter schedule than the previous protocols in this area. The main weakness of these protocols is the increased number of idle (unused) time spaces of the constructed schedule. The high level of these gaps leads to low channel utilization and keeps network efficiency low. At the same time, a prediction mechanism is adopted, proposed in [20], offering significant time profit through pipelining the calculation procedure of the schedule with the real transmission of the packets. The suggested scheme includes two common strategies: the strategy of linear search for K elements (K -ls) and the strategy of the balanced binary searching tree (BBST) with prototypes. Both strategies share a common goal. They give priority to the service of the long-length packets, in order to manage the unallocated time spaces of the schedule. It is evident that long-length requests reserve more (continuous) timeslots than short-length requests. The schedule technique of IOSS changes the service order of the data demands, by giving priority to the most demanding time requests. This schedule technique allows a better utilization of the channels, due to the fact that it increases the possibility of finding an appropriate interval to schedule the short-length demands. The rest of the paper is organized as follows: Section 2 describes network architecture, while Section 3 analyzes the three previous scheduling algorithm protocols (OIS, POSA, and CS-POSA). Section 4 presents the new suggested scheduling algorithm, whereas Section 5 analyzes performance measures and is followed by figures and detailed comparisons between the performances of the three algorithms in Section 6. Finally, concluding remarks are given in Section 7.

2. Network background

This section presents background material regarding the architecture of the network and details concerning the reservation nature of the protocol.

2.1. Network architecture

The network consists of N nodes, each connected to a passive star coupler via two-way optical fibers. The system supports W data channels of the same capacity. Even though there is no separate control channel used for coordination, the suggested protocol is still pre-transmission coordination-based, since the set of W data channels are used for both control and data packets. Generally, in

a N -node optical network, the most effective communication structure is achieved when each node has a tunable transmitter and a tunable receiver (TT–TR system) in combination with N channels, one channel per node. In this way, each node has its own unique channel and the protocol becomes collision free. Such a structure though would be very difficult to be applied because network size (N) should be scalable, transceivers are relatively expensive so that each node can be equipped with only a few, technological constraints dictate the number of WDM channels supported in a fiber should be limited to W (whose value limited in a few tens today, perhaps as high as 160 in some systems, and is expected to improve with time and technological breakthroughs), and switches with large port count (i.e., large W) are very costly [1]. So, the solution of a TT–FR system seems to go along with the current technological and financial developments. Each node receives data on a specific channel referred to as home channel. We consider that the number of nodes is larger than the number of channels ($N > W$), so N/W nodes share a home channel.

2.2. Reservation protocol

Each node generates (requests) variable-length packets (or messages), each containing one or more fixed-length data packets. Also, each node maintains W queues, one queue per channel [20]. In this way, a packet is generated at a node, is queued in the channel queue, corresponding to the destination's home channel. Each generated packet has variable length ranging from 0 (idle request) to K timeslots. So, K denotes the upper bound on a node's request of a channel. For synchronization reasons it is considered that a fixed-length data packet needs exactly one timeslot to be transmitted. Hence, for a generated message of length equal to t fixed-length, data packets need exactly t timeslots for transmission.

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 node is assigned a unique slot for broadcasting its control packet to all channels (simultaneously) by means of its transmitter. Control packets are received by all nodes in their corresponding home channel by means of their FR and make reservations for the data phase. Then, each node computes the transmission schedule for the data phase independently. Given the high transmission speeds (of the order of a few Gbps) the schedule computation time should be as small as possible.

It is clear that N/W nodes share the same home channel, something that will probably bring about a problematic situation, meaning 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 attempt to transmit to node n_3 at the same time, a collision will occur, since the data travel to the same home channel simultaneously. So,

the packets are destroyed and there is need for updating the nodes and retransmitting the signals; processes which are time-consuming and complicated. Another way of avoiding 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.

The protocol examined is collision free, i.e., it secures the transmission of the data to all the nodes, without any collision. Of course, in order this to be achieved, a synchronization mechanism is required. 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 info obtained by a shared control channel [25]. The distributed algorithm accepts the transmission time demands of each node of the network and stores them in a matrix $D = [d_{ij}]$, 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 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. Thus, the algorithm can be able to decide how transmissions and receptions for the next phase should be made.

3. A review of OIS, POSA, and CS-POSA scheduling algorithms

In this paper we design and evaluate a channel access protocol that improves a set of previous pre-transmission coordination-based protocols. We adopt online interval scheduling (OIS) as the basic scheduling technique. The choice of OIS is justified by the fact that OIS presents very low computational complexity (the lowest time complexity of TT–FR family protocols) and it is a very simple online algorithm. Although OIS has many advantages, e.g., very low time complexity (linear of N), collision free, and simple hardware implementation, it lacks in efficiency. Also, the suggested protocol uses the same prediction mechanism as the predictive online scheduling algorithm (POSA). POSA introduces a prediction system, which reduces the computation time of the scheduling process by means of predicting the requests of each node for the following frame. However POSA functions in the same way as OIS and therefore produces a schedule with low channel utilization and low schedule efficiency. Check or sort predictive online scheduling algorithm (CS-POSA) is an improvement of OIS and POSA and offers a better utilization of the available channel. It shifts the service order of the requests of the nodes on the basis of the total requests of each node. However, this improvement is restricted by the network traffic and keeps network

throughput higher but still quite near to the levels of POSA. Moreover, CS-POSA adds some extra queue delay. In the rest of this section the three protocols, OIS, POSA, and CS-POSA, are analyzed.

The OIS produces a schedule for transmitting the demand matrix and runs in $O(NW^2K)$, where N represents the number of the nodes in the network, W represents the number of the transmission channels, and K represents the maximum possible value in the demand matrix. The fact that OIS is a simple algorithm and includes simple and fast procedures without decomposition of the demand matrix

that node n_1 requests three timeslots in channel w_1 , then transmission cannot be programmed in the space $[3, 5]$ where the channel is available, because the node has been programmed to transmit elsewhere. So, transmission will be programmed in the space $[11, 13]$ and the lists will be renewed accordingly. In other words, the new lists for channel w_1 will be $[3, 8]$, $[14, 18]$, and $[22, \infty]$, while the lists for node n_1 will be $[14, 19]$ and $[25, \infty]$. The algorithm and additional information can be found in [19]. A pseudo-code is given in order the operation of the OIS algorithm to be explained more clearly:

OIS algorithm

Begin frame

Initialize W intervals (one for each channel)

For each node n_i ($0 \leq i \leq N - 1$)

Initialize the interval of node n_i

For each channel w_j ($0 \leq j \leq W - 1$)

Find a suitable time space for the request of node n_i on channel w_j , starting from the beginning of frame (timeslot 0), so that there is no other node scheduled for this space, for this channel and there is no schedule for node n_i on other channel.

Update the interval of channel w_j

Update the interval of node n_i

End frame

like the previous HRP/TSA [19], offers powerful advantages as far as its execution time is concerned. As an online algorithm, the OIS starts to construct the schedule matrix as well, as soon as it receives a set of requests from a node (without the whole demand matrix being available). Then, the requests of each node are examined in sequence before proceeding to the next row and the schedule is progressively built. So, within each node, the same distributed algorithm runs concurrently and the same status information (demand matrix) is updated. This distributed algorithm leads all the nodes to end up making the same decisions and the protocol does not allow collisions. In order each node to be able to execute the scheduling algorithm of OIS, it needs to maintain a list of time intervals available on every data channel. Furthermore, for each node whose request is processed nodes maintain an additional list of intervals that have not been assigned yet to the specific node for transmission. For example, the list for the channel w_1 can have the intervals $[3, 8]$, $[11, 18]$, and $[22, \infty]$. That means that the intervals (which denote timeslots) $[1, 2]$, $[9, 10]$, and $[19, 21]$ have been assigned to nodes and no more transmissions should be assigned to the specific time spaces in order to avoid collisions. The other list refers to the node scheduled each time. Assuming the list of node n_1 to consist of the intervals $[11, 19]$ and $[25, \infty]$, it means that node n_1 has been scheduled to transmit in the timeslots $[1, 10]$ and $[20, 24]$. If we suppose

The main problem of OIS is that it requires a lot of construction time for the final schedule matrix of each frame. As it has already been mentioned in Section 2, each frame consists of two phases, the reservation phase and the data phase. According to OIS, the transmission of the data should wait for the completion of the schedule of each frame. In order to reduce the waiting (for transmission) time of the nodes, due to the delay of the schedule construction (by OIS), the POSA tries to eliminate the possible delay introduced by the scheduling computation between the reservation and the data phases of each frame. POSA is an extension of OIS and is based on the traffic prediction in accordance with the history of the recent traffic requests. The main aim of the algorithm is the elimination of the delay which is caused by the calculation of the scheduling matrix between the phases of reservation and transmission. For this purpose, POSA assumes that the reservation information of each reservation phase is inserted in a prediction mechanism, whose function is mainly based on the model of the hidden chain Markov. After an initial training period, which lasts for V cycles, the protocol works like OIS. However, during the same period, the prediction mechanism is informed about the reservation requests, filling a history queue for each possible transmission to each node, in each channel. As soon as V cycles are completed, the algorithm moves to a prediction phase. During this phase, the schedule computed for the

Table 1
A brief comparison between OIS, POSA, and CS-POSA

	OIS	POSA	CS-POSA
Scheduling algorithm complexity	$O(W^2KN)$	$O(W^2KN)$	$O(W^2KN) + O(\log N/W)$
Prediction mechanism complexity	–	$O(K + 1 + V)$	$O(K + 1 + V)$
Channel utilization	Low	Low	Medium
Network throughput	Medium	High	High
Schedule efficiency	Low	Low	Medium
Packet delay	Low	Low	Medium

N denotes the number of nodes, W denotes the number of channels, K denotes the maximum length of packet, and V denotes the size of the history queue where the real requests of the nodes are stored (in the prediction mechanism). The complexity is calculated using p processors.

data phase of a certain frame, instead of taking into account the reservation requests of the same frame's reservation phase (as in OIS), relies on the predicted reservations (output of the predictor), provided for the scheduling algorithm in the previous frame. The current reservation phase is not used to provide the requests for the current frame but to train the prediction mechanism, which affects the predictions of the following frames. It is obvious that the prediction mechanism should be accurate, so that it does not lead the scheduling algorithm to false conclusions. According to [20] the prediction system does not lack accuracy to a great extent, since it has been found that in 70% of the predictions, the percentage of failure was less than 20%, regardless of the number of the nodes, channels, or the maximum value of K of each entry in the demand matrix. It is also clear that POSA algorithm does not add extra complexity to the system since it works linearly with the increase of the nodes or the channels. Its complexity is equal to $O(K + 1 + V)(NW)$. As a conclusion, it can be said that the prediction mechanism of POSA provides safe predictions for the demand matrix of the following transmission frame, minimizing (if not eliminating) the calculation time of the scheduling process, which takes place in the reservation phase.

The CS-POSA [21,22] protocol is an improved edition of POSA. Its aim is to extend POSA, maintaining the pipelining of the schedule computation and the full operation of the predictor. On one hand, it retains the same algorithm that constructs the schedule matrix as OIS, but on the other hand, it adopts the prediction mechanism as regards the requests of each node, as it presented above. The extension of CS-POSA is based on shifting the schedule computation of the nodes, in other words, on guiding the order of checking and programming of the nodes. Shifting depends on the workload of each node, which means that CS-POSA comprehends better not only the general traffic of the network, but also the specific workload contained in each node. CS-POSA introduces the operation of a new function before the construction of the schedule matrix. More specifically, after it has received the prediction requests from the prediction mechanism, it proceeds to the following two steps: first, it adds the total requests of each node for each channel and enters them in a vector. Then, it sorts the vector in a declining order so that

the requests of the node with the biggest total of requests have priority over the one with the smallest total of requests. After this procedure has been completed, the same algorithm as OIS is applied, starting from the node with the most requests in all channels and finishing with the node with the fewest requests. The crucial difference between the CS-POSA and OIS/POSA lies in the fact that the OIS and POSA algorithms do not examine the total of the requests of the nodes, i.e., they do not analyze extensively the load requested by each node of the network but process one by one the network nodes regardless of their distributed load. Table 1 presents a summary with the main characteristics of the above protocols.

4. Targets of the IOSS protocol

IOSS protocol is an improved edition of OIS algorithm. It attempts to produce a shorter schedule, an action leading to an efficient schedule. At the same time, POSA prediction mechanism is adopted in order to maximize the throughput of the network. More specifically, IOSS protocol sets OIS as the basis schedule technique owing to the fact that it is very simple and allows fast scheduling with neither slow nor complex functions and actions. It is very important to mention that the complexity of OIS is lowest in comparison with the other scheduling algorithms. On the other hand, the prediction mechanism is optional. That means that IOSS could also function without a prediction mechanism, if, for example, we are not able to predict the traffic prototype of the network (in cases of the requests of the nodes appear at a totally random order) or if the prediction mechanism for a variety of reasons, cannot predict a great number of correct requests (i.e., if it has low performance and accuracy). Whether the proposed IOSS algorithm functions with a prediction mechanism or not, we pose a set of distinct targets the proposed algorithm should fulfill. In case that the system does not include a prediction mechanism, the following targets are set:

- the algorithm should not surpass the OIS algorithm in complexity;
- the algorithm should be scalable, i.e., its complexity should be in linear to the values of N and W ;

- the algorithm should perform better than OIS, i.e., the produced schedule matrix should allow a better use of the channels regardless of the values of N , W and K .

As far as the first target is concerned, we can say that it is accomplished by IOSS, since its complexity does not surpass the time $O(NW^2K)$ of OIS. Two strategies have been developed, the K -ls strategy and the BBST strategy, in order to achieve this goal. The algorithm using either the one or the other strategy does not surpass the complexity time of OIS. As far the second target is concerned, it has also been handled with success since both strategies, developed in Section 4, produce performance times up to $O(KNW)$. Last but not least, the third target, the main performance of the algorithm, is accomplished through extensive results of simulation. What is better analyzed in Section 4 is that the new IOSS algorithm offers a better use of the channels (thus, a better network throughput) regardless of the number of the nodes, the number of the channels, and the parameter K , which expresses the size of the network load.

In case the system includes a prediction mechanism, two extra targets are set:

- the prediction mechanism should be accurate and at the same time capable of adjusting to the changes of the traffic in the network;
- the prediction mechanism should work within lower complexity than OIS and IOSS, and should be progressive.

Concerning the first target, a percentage of failure is expected and predictable as well. Undoubtedly, a protocol predicting the traffic of the network should be effective, thus accurate. According to the measurements that have been carried out by POSA, the prediction mechanism is considered to be accurate and capable of coping with the changes related to the traffic and the load. It must also be said that multiple values of K have been used during the experiments and the incorrect predictions proved that they cannot decrease tragically, in terms of channel utilization. As far as the second is concerned, we must mention that the complexity of the prediction mechanism is equal to $O(K + 1 + V)(NW)$, where V is a value denoting the size of the history queue where the real requests of the nodes are stored. The prediction mechanism has $N \times W$ independent mechanisms. Hence, it is possible to use more than one processor at the same time, let us say E processors. If $E = NW/p$, then E is a conjunction of N and W divided by a value p and then we could say that the complete complexity of the prediction mechanism is

$$O\left(\frac{(K + 1 + V)(NW)}{E}\right).$$

We should also highlight the fact that with the suitable number of processors, which at the same time calculate the

scheduling algorithm, we could assume that the prediction mechanism is applied in a stable asymptotic time.

4.1. Function of the IOSS protocol

The proposed IOSS protocol reorganizes the service order of the requests per node and per channel based on the value of each entry of the demand matrix. The scheduling algorithm calculates the schedule matrix starting from the request of the demand matrix with the biggest length (value) and continuing with the subsequent biggest length (value) of the requests. In other words, we can say that IOSS grades the demand matrix in descending order and then feeds the OIS algorithm with the reorganized demands. The sorting of the demand matrix, which grades $N \times W$ elements, is a time-consuming procedure and requires high complexity. Therefore, two different architectural alternatives are presented, so that the proposed algorithm is as fast and simple as possible.

The basic aim of the algorithm is to supply OIS with a reordering sequence, called σ_{IOSS} , which includes the elements of the demand matrix, sorted in descending order. Obviously, the input of the proposed IOSS algorithm is the (predicted at times) demand matrix and the output of the algorithm is the NW ordered elements, denoted by σ_{IOSS} . At this point we should note that the input to IOSS is the demand matrix predicted by POSA, and thus all its elements are directly known. In other words, once POSA has completed its prediction, IOSS receives the predicted demand matrix $P = [p_{i,j}]$ and each element $p_{i,j}$, where $i \in N$ and $j \in W$, from the first $P_{0,0}$ to the last $P_{N-1,W-1}$ is made at the same time known with no delay at all. So, IOSS starts to reorder the elements of the P matrix, something which constitutes the main procedure of the algorithm.

It has already been mentioned that the elements of P table could be reorganized with the use of a sorting algorithm, but something like that would result in wasting an extremely important amount of time. As a result of this limitation, a technique is required in order to set the entries of the demand matrix in descending order. Two different techniques for the support of a suitable procedure are suggested, so that the desired order is achieved. The first alternative strategy uses the linear search as a basic function. The second strategy sets a balanced searching IOSS with prototypes as its basic method. In Sections 4.2–4.5, there is an analysis of all the alternative strategies which have a main characteristic, although they use different types of data structure: their final output is the same, i.e., the reformed demand matrix in descending order.

4.2. Strategy of linear search for K elements

The sorting of NW elements is analyzed in the simplest way within the problem of the NW maximum elements search. In other words, the problem is reformed during the process of searching for the biggest appearing for each time

element in the demand matrix P of two dimensions. However, the solution to the specific problem does not mean that we should search for NW elements, except for K . This occurs because the values in the demand matrix P range from 0 to K and are completely positive values. Consequently, we know beforehand that the maximum value in the demand matrix is equal to K , and the minimum value is equal to 0. It is relatively easy to put the NW elements of the P in order in a descending array, on condition that we also seek for the K different numbers (without 0) just once. In other words, the IOSS algorithm performs K linear searches to form the σ_{IOSS} . The search order is the following: initially the value of K , which is the maximum entry in the matrix, is searched. The search returns all the values of K found in table P and more precisely, returns the position where the elements were found (the node and the channel). So, if while searching for the value K the algorithm finds that the value is on the P table z times, where $z < NW$, then it returns the positions found, i.e., $P_{i_1,j_1}, P_{i_2,j_2}, \dots, P_{i_z,j_z}$, where $i_1, i_2, \dots, i_z \in N$ and $j_1, j_2, \dots, j_z \in W$. These values are stored in the σ_{IOSS} , randomly. Afterward, the algorithm searches for the next biggest value, which is equal to $K - 1$, in the same way. Next, the algorithm completes the process of linear searching for the value 1. There is no point in searching for value 0, since it would not have practical application on the final schedule matrix, because the specific node, in the specific channel does not have any packets to transmit. IOSS algorithm completes the formation of the σ_{TPEE} in $O(KNW)$ time. It is obvious that the complexity of IOSS algorithm, using the strategy of linear search for K elements, is less than the complexity of OIS algorithm which is in $O(KNW^2)$ time.

4.3. Numerical example of the strategy of linear search for K elements

Let us consider a network that consists of four nodes and two channels and that the maximum entry of the demand matrix can be equal to 5. So, $N = 4, W = 2$, and $K = 5$. We also assume a random transmission frame f , in which the demand matrix, predicted by eight different predictors, denoted by P^f , is as follows:

$$P^f = \begin{array}{c|cc} & w_0 & w_1 \\ \hline n_0 & 3 & 2 \\ \hline n_1 & 4 & 1 \\ \hline n_2 & 2 & 5 \\ \hline n_3 & 5 & 5 \end{array}$$

It is obvious from the demand matrix P^f that node n_0 requires three timeslots in channel w_0 and two timeslots in channel w_1 . Respectively, node n_1 requires four timeslots in channel w_0 and one timeslot in channel w_4 . Node n_2 requires two timeslots in channel w_2 and five timeslots in

channel w_1 . Finally, node n_3 requires five timeslots both in channel w_0 and in channel w_1 . In total, the nodes requested 27 transmission timeslots for the transmission frame f .

According to IOSS algorithm and following the strategy of linear search for K elements, the algorithm begins K searches in total, one for each possible value that can exist in the demand matrix P^f . The first search refers to the value of 5, since it is the maximum value likely to appear (once or more than once) in P^f matrix. The search of the element in the two-dimensional P^f table is successful and returns three values equal to 5. These values are found in the positions $P_{2,1}^f, P_{3,0}^f$, and $P_{3,1}^f$. The positions are stored randomly in σ_{TPEE} , which at this moment contains only three elements:

$$\sigma_{\text{IOSS}} = \{P_{3,1}^f, P_{3,0}^f, P_{2,1}^f\}.$$

Then the algorithm searches for entries that have the value 4. Only one entry, $P_{1,0}^f$, is found in matrix P with this value and is added to σ_{IOSS} . Shortly afterward, the search for value 3 returns the position $P_{0,0}^f$, and the search for value 2 returns the positions $P_{0,1}^f$ and $P_{2,0}^f$. Finally, the search is completed with value 1 which is stored in the entry $P_{1,1}^f$. The final order is

$$\sigma_{\text{IOSS}} = \{P_{3,1}^f, P_{3,0}^f, P_{2,1}^f, P_{1,0}^f, P_{0,0}^f, P_{0,1}^f, P_{2,0}^f, P_{1,1}^f\}.$$

This order feeds the basic scheduling algorithm OIS to construct the final schedule matrix.

4.4. Strategy of the BBST tree with prototypes

A searching binary tree is an alternative way of achieving the reorganization of the elements of the demand matrix in a descending order. The most important feature is that the calculations on the trees are simple and of low complexity. It is known that each calculation (data input, delete, and search) that the IOSS algorithm has, is of $O(\log n)$, where n is the total of the elements (keys) of the tree. Using trees with prototypes means that some values of the demand matrix can appear more than once. So, the structure of the tree, expressing the collection of the values of the demand matrix in descending order, has also the ability to store multiple entries (keys).

So, the algorithm initially accepts the output of the prediction mechanism, i.e., the P demand matrix. The binary tree, denoted by T_p , is formed by the algorithm as the system accepts the requests of each node and each channel POSA predicted. For every value IOSS accepts, it enters the equivalent key on the tree T_p . Since the demand matrix consists of NW elements, we reach the conclusion the tree will also have NW keys. Additionally, because of the fact that the tree is balanced, its height is equal to $\log NW$. Thus, every action of the algorithm on the T_p tree has maximum $O(\log NW)$ complexity. As mentioned above, the algorithm searches for K different keys (which can be found once or more than once on the tree), K times. Each of these searches has $O(\log NW)$ complexity. As a result, the

whole process has $O(K \log NW)$ complexity. Besides that, note that the tree must be constructed by the algorithm in real time. Nevertheless, the initial construction of the tree (IOSS algorithm) occurs along with the continuous acceptance of the values of the demand matrix that POSA predicts. This piece of evidence in combination with the fact that it is not necessary to delete keys on the tree during the execution of the algorithm contribute to the acceptance of the view that the construction of the tree can be regarded as an online process and the degree of complexity is not taken into consideration for simplicity reasons. Conclusively, it can be claimed that the use of a tree structure of data functions as effectively as the strategy of linear search for K elements with $O(K \log NW)$ complexity.

4.5. Numerical example of the BBST with prototypes strategy

Let us consider the same network that was mentioned in the numerical example of the strategy of K searches with four nodes, two channels, and upper limit of requests equal to 5. A random transmission frame f and the same demand matrix P^f are analyzed below:

$$P^f = \begin{array}{c|cc} & w_0 & w_1 \\ \hline n_0 & 3 & 2 \\ n_1 & 4 & 1 \\ n_2 & 2 & 5 \\ n_3 & 5 & 5 \end{array}$$

This demand matrix consists of $4 \times 2 = 8$ elements produced as an output of POSA prediction mechanism. According to the strategy of the BBST, the IOSS algorithm accepts one by one the requests POSA predicts and places its values on T_p tree. It is known that the height of a binary tree with n elements is equal to $\log n$. Also, note that during

the input of the keys, i.e., the values of the P^f , the tree might become unbalanced. In this case, some replacements must take place so that its balance can be ensured (like the case of AVL trees or the red–black trees). The main point we examine regarding the IOSS algorithm and the specific strategy is whether every action on the tree during its formation is $O(\log n)$, where n is the total of the keys of the tree. In the specific numerical example, the total of the keys stored on T_p tree is NW or 8. The final appearance of the tree after the input of the eight keys is shown in Fig. 4.

After the completion of the tree, the algorithm executes $K = 5$ searches, one for every different value, from 1 to 5. Initially, the algorithm searches for key 5 and finds it three times. At the same time, it enters the positions of these keys randomly (or alternatively a pair of two integer numbers, the number of the node and the number of the channel that the transmission request found is about). Also, these keys are deleted and the algorithm looks for key 4. Following the same reasoning, it looks for all the keys until it finds key 1 and the research comes to an end. The order is the same as the one in the example of the strategy of K linear researches, i.e.

$$\sigma_{\text{IOSS}} = \{P_{3,1}^f, P_{3,0}^f, P_{2,1}^f, P_{1,0}^f, P_{0,0}^f, P_{0,1}^f, P_{2,0}^f, P_{1,1}^f\}.$$

Alternatively, if we use the simplest symbolism for the pair of the numbers of the node and the channel, then the order would be a little different, i.e.

$$\sigma_{\text{IOSS}} = \{(3, 1), (3, 0), (2, 1), (1, 0), (0, 0), (0, 1), (2, 0), (1, 1)\}.$$

4.6. Comparative example with a random demand matrix

In order to compare and present the behavior of the different scheduling algorithms, it is useful to study further the way they function, based on a numerical example with a random demand matrix. The algorithms compared are POSA, CS-POSA, and the proposed IOSS. Let us consider

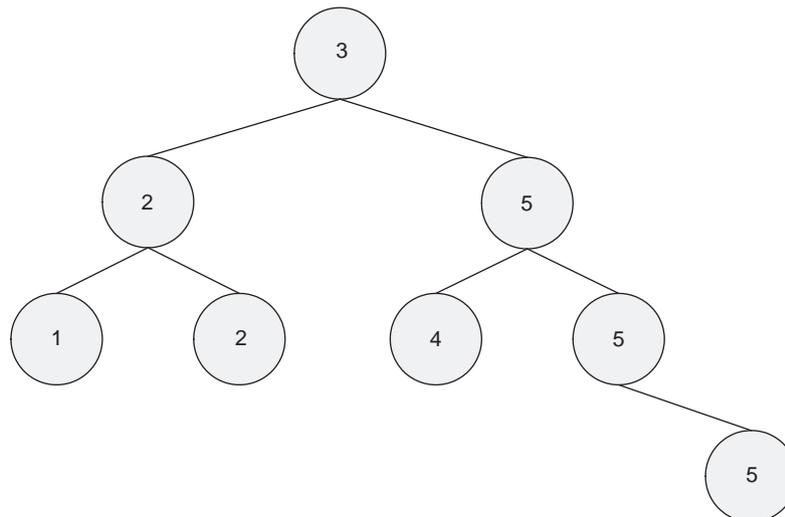


Fig. 4. The constructed tree by the balanced binary searching tree strategy.

P^f demand matrix of Section 4.3:

$$P^f = \begin{array}{c|cc} & w_0 & w_1 \\ \hline n_0 & 3 & 2 \\ \hline n_1 & 4 & 1 \\ \hline n_2 & 2 & 5 \\ \hline n_3 & 5 & 5 \end{array}$$

P^f matrix will be regarded as input to all the algorithms under examination, so that the way the schedule matrix is formed to be written down step by step. As it has been mentioned, the POSA algorithm works in the same way as OIS, regarding the building of the final schedule matrix. Counting on these data, POSA starts examining the requests of the first node n_0 , continues with the requests of the second node n_1 , then the requests of node n_2 , and finally the requests of the last node n_3 . It is important to mention that POSA algorithm does not change the service order of the nodes and always starts the formation of the schedule matrix from the first node and finishes with the last node (regardless the additional load of each node). In this way, the service order of network nodes for POSA

timeslots	0	1	2	3	4
Channel 0	No	No	No		
Channel 1				No	No

Fig. 5. The schedule constructed by POSA, after the service of the node's n_0 requests.

Timeslots	0	1	2	3	4	5	6
Channel 0	No	No	No	N1	N1	N1	N1
Channel 1	N1			No	No		

Fig. 6. The schedule constructed by POSA, after the service of the node's n_1 requests.

Timeslots	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Channel 0	No	No	No	N1	N1	N1	N1	N2	N2	N3	N3	N3	N3	N3	idle	idle	idle	idle	idle
Channel 1	N1	idle	idle	No	No	idle	idle	idle	idle	N2	N2	N2	N2	N2	N3	N3	N3	N3	N3

Fig. 7. The final schedule constructed by POSA.

algorithm would be

$$\sigma_{\text{POSA}} = \{P_{0,0}^f, P_{0,1}^f, P_{1,0}^f, P_{1,1}^f, P_{2,0}^f, P_{2,1}^f, P_{3,0}^f, P_{3,1}^f\},$$

Node n_0 requests three timeslots in channel w_0 and two timeslots in channel 1. POSA algorithm places the requests of node n_0 in the first phase of the construction of the time schedule matrix. The matrix is shown in Fig. 5. Then, POSA examines the requests of node n_1 . Node n_1 requires four timeslots in channel w_0 and one timeslot in channel w_1 . POSA places the requests of node n_1 in the schedule matrix in the way OIS suggests, so the second phase is completed (Fig. 6). Now, it is the turn of node n_2 , which requires two timeslots in channel w_0 and five timeslots in channel w_1 . The last node for the transmission frame f is node n_3 , which requests five timeslots for both channel w_0 and channel w_1 . After the completion of the programming of node n_3 , the final schedule matrix for frame f is shown in Fig. 7. It is easy to observe that POSA algorithm has constructed a schedule with the characteristics below:

- its duration is equal to 19 timeslots;
- the number of the idle sub-timeslots is equal to 11;
- the percentage of the utilization of the channels is equal to demanded timeslots/(number of channels * duration of the schedule) = 27/(2 * 19) = 71%.

Furthermore, it is useful to examine step by step the way the algorithm CS-POSA works for the same demand matrix P^f . The specific algorithm forms a vector, called Sum, which displays the following properties:

- it has N positions equal to the number of the nodes of the network;
- each N position of the vector is the total of each line of P^f demand matrix.

So, the vector Sum^f referring to the P^f demand matrix for the transmission frame f is as follows:

$$\text{Sum}^f = \begin{array}{c|c} & \text{Sum} \\ \hline n_0 & 5 \\ \hline n_1 & 5 \\ \hline n_2 & 7 \\ \hline n_3 & 10 \end{array}$$

According to the algorithm CS-POSA, the service order of the nodes changes based on the vector Sum^f , in a descending order. The newly sorted vector is shown

below and according to it, the new service order of the nodes would be the following:

$$\text{Sum}^{f'} = \begin{matrix} n_3 & 10 \\ n_2 & 7 \\ n_1 & 5 \\ n_0 & 5 \end{matrix},$$

$$\sigma_{\text{CS-POSA}} = \{P_{3,0}^f, P_{3,1}^f, P_{2,0}^f, P_{2,1}^f, P_{1,0}^f, P_{1,1}^f, P_{0,0}^f, P_{0,1}^f\}.$$

It must be noted that node n_0 and node n_1 contain the same total of requests and the choice is made at random. The first step of the algorithm CS-POSA is completed resulting in the determination of the service order $\sigma_{\text{CS-POSA}}$. The next step is to form the final time schedule matrix. CS-POSA receives the requests of the node n_3 which includes the biggest total of requests, and forms the schedule matrix (Fig. 8). In turn, CS-POSA examines the requests of node n_2 , which contains the biggest total of requests, and forms the table in Fig. 9. Then, the requests of node n_1 are examined and finally, the ones of node n_0 . The final schedule matrix is shown in Fig. 10. From the matrix that CS-POSA formed, we can conclude that

- its duration is equal to 16 timeslots;
- the number of the idle sub-time cells are equal to 5;

Timeslots	0	1	2	3	4	5	6	7	8	9
Channel 0	N ₃									
Channel 1						N ₃				

Fig. 8. The schedule constructed by CS-POSA, after the service of the node's n_3 requests.

Timeslots	0	1	2	3	4	5	6	7	8	9
Channel 0	N ₃	N ₂	N ₂							
Channel 1	N ₂	N ₃								

Fig. 9. The schedule constructed by CS-POSA, after the service of the node's n_2 requests.

Timeslots	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Channel 0	N ₃	N ₂	N ₂	N ₁	N ₁	N ₁	N ₁	N ₀	N ₀	N ₀	idle	idle				
Channel 1	N ₂	N ₃	idle	N ₁	idle	idle	N ₀	N ₀								

Fig. 10. The final schedule constructed by CS-POSA.

- the percentage of the utilization of the channels is equal to demanded timeslots/(number of channels * schedule length) = 27/(2 * 16) = 84%.

For the same demand matrix, the IOSS algorithm changes the order of the sequence of the transmission requests:

$$\sigma_{\text{IOSS}} = \{P_{3,1}^f, P_{3,0}^f, P_{2,1}^f, P_{1,0}^f, P_{0,0}^f, P_{0,1}^f, P_{2,0}^f, P_{1,1}^f\}.$$

Firstly, the IOSS serves the requests $P_{3,1}$ and $P_{3,0}$. The first phase of the schedule matrix, constructed by IOSS is depicted in Fig. 11. Next, IOSS schedules the demands $P_{2,1}$ and $P_{1,0}$. Fig. 12 shows the second phase of the construction. The final schedule is formed based on the σ_{IOSS} and it is plotted in Fig. 13. The matrix that IOSS formed help us to conclude that

- its duration is equal to 14 timeslots;
- the schedule includes only one idle sub-time cell;
- the percentage of the utilization of the channels is equal to demanded timeslots/(number of channels * schedule length) = 27/(2 * 14) = 96%.

Timeslots	0	1	2	3	4	5	6	7	8	9
Channel 0	N ₃									
Channel 1						N ₃				

Fig. 11. The schedule constructed by IOSS, after the service of the node's n_3 requests.

Timeslots	0	1	2	3	4	5	6	7	8	9
Channel 0	N ₃	N ₁	N ₁	N ₁	N ₁					
Channel 1	N ₂	N ₃								

Fig. 12. The schedule constructed by IOSS, after the service of the requests $p_{3,0}$, $p_{3,1}$, $p_{2,1}$, and $p_{1,0}$.

Timeslots	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Channel 0	N ₃	N ₁	N ₁	N ₁	N ₁	N ₀	N ₀	N ₀	N ₂	N ₂				
Channel 1	N ₂	N ₃	N ₁	idle	N ₀	N ₀								

Fig. 13. The final schedule constructed by IOSS.

4.7. Phases of the protocol IOSS

As mentioned before, the protocol IOSS works with a prediction mechanism. That means that within each transmission frame, the prediction mechanism predicts the requests of the following transmission frame. However, in order to achieve an accurate prediction, the mechanism must be trained for a specific time period and should understand the traffic of the network as much as possible. For this reason, just like the POSA protocol, IOSS takes place in two different phases, the training phase and the prediction phase. In the first phase, IOSS does not activate the prediction mechanism but functions just like OIS. It receives the requests of the nodes and then constructs the schedule based on the actual requests of the current frame. In the following phase, IOSS activates the prediction mechanism and starts to predict the requests of the nodes for the next transmission frame. In other words, for a certain transmission frame, IOSS makes the transmissions based on the requests that the prediction mechanism predicted in the previous transmission frame. At the same time, it also receives the actual requests of the nodes and places them in the history queues, so that the protocol is informed of any changes about the network traffic.

5. Performance measures

The performance of three prediction-based algorithms is presented. POSA, CS-POSA, and IOSS are executed, in order to compare their efficiency, in terms of channel utilization, network throughput, and mean packet waiting time (in queues). The varied simulation parameters 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 from 0 to K and in order the goal of scalability to be achieved, the value of K does remain constant in the following experiments but 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 as follows:

- L symbolizes schedule length, which denotes the number of slots in the data phase as determined by the schedule algorithm.
- R symbolizes total slots requested by all nodes, which denotes the total number of timeslots that were requested by all the nodes of the network.
- U symbolizes schedule or channel utilization, which denotes the number of slots actually utilized for packet transmission in a schedule matrix. Scheduling utilization is defined as

$$U = \frac{\text{total slots}}{\text{requested slots} * \text{channels}} \quad \text{or} \quad U = \frac{R}{LW}.$$

- S stands for the transmission rate of each channel and has been set in 2.4 Gbps.
- Γ symbolizes throughput, which denotes the average number of bits transmitted per transmission frame per channel. Since the three algorithms examined do not involve computation time delay due to pipelining throughput the relation becomes

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

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

6. Simulation results

This section presents the simulation results. Three algorithms, that utilize the prediction method, POSA, CS-POSA, and IOSS 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. The suggested IOSS protocol is superior to POSA and CS-POSA because it works much better than them in terms of channel utilization and network throughput. Secondly, the remarkable performance of IOSS is proved by the fact that the mean waiting time in the queues before the transmission can be reduced. According to the results of the simulation, it is assumed that N is the number of nodes, W is the number of 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. In order to study the behavior of each protocol, we make the following assumptions about the simulation model:

1. Traffic pattern is uniform, i.e., data requests are destined to every other node with equal probability.
2. For each frame, nodes may generate data requests from 0 to K with equal probability.
3. The line is defined at 2.4 Gbps per channel and the tuning time is ignored for simplicity reasons.
4. The outcomes resulted from 10 000 frames, out of which the first 1000 were learning frames and PRO functioned as OIS.
5. The same traffic pattern is used for all protocols.

6.1. Channel utilization results

Figs. 14 and 15 show the results of the performance of the three algorithms for $W \in \{5, 10\}$, $N \in \{10, 20, 30, 40\}$,

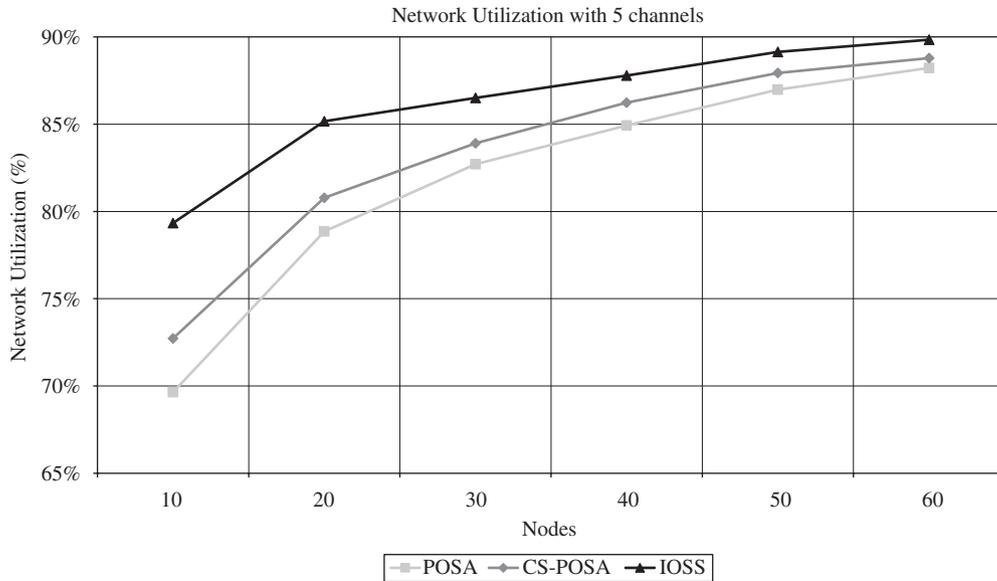


Fig. 14. Channel utilization with five channels.

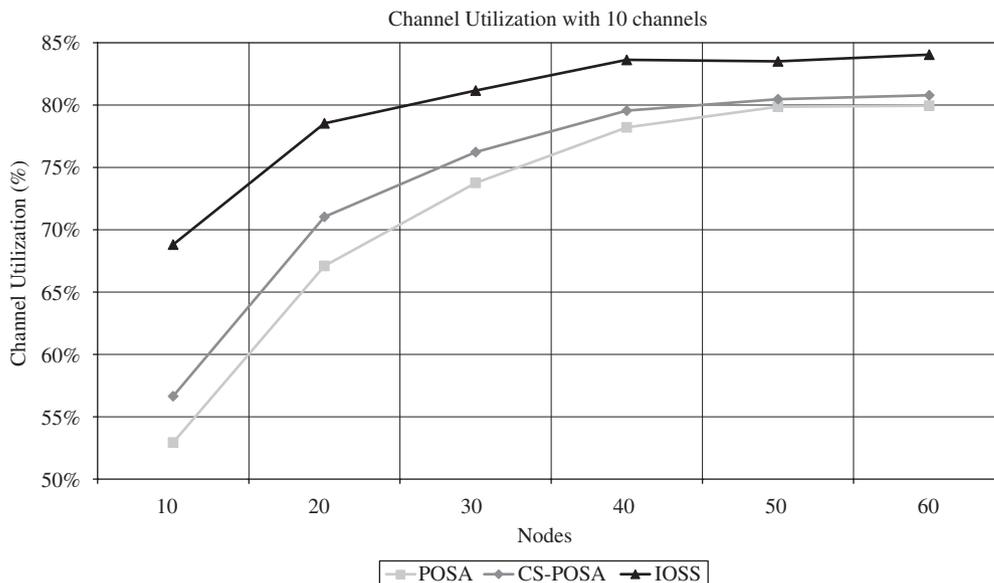


Fig. 15. Channel utilization with 10 channels.

50, 60}, $K = \text{FLOOR}(NW/5)$, in terms of channel utilization. For five channels, it can be observed that IOSS algorithm is superior to POSA and CS-POSA, its maximum difference over POSA amounts to 9.68% for 10 nodes and its minimum difference amounts to 1.62% for 60 nodes. Also for five channels, the maximum difference over CS-POSA reaches 6.66% for 10 nodes and the minimum difference for 60 nodes reaches 1.06%. In Fig. 15 the channel utilization is presented for 10 channels. Once more, IOSS turns out to be more improved than POSA and CS-POSA. The maximum difference between IOSS and POSA comes up to 15.88% for 10 nodes and the minimum comes up to 3.64% for 50 nodes. IOSS excels CS-POSA, ranging from 12.16% for 10 nodes

to 3.04% for 50 nodes. Overall, IOSS presents the best performance, as regards channel utilization, because it gives priority to long-length transmission requests and this action leads to more unallocated free intervals for the following short-length requests. The order strategy of IOSS allows a better exploitation of the available channels, by increasing the possibility of the short-length requests to capture the intermediate transmission timeslots in the schedule matrix.

6.2. Network throughput results

The achieved throughput is shown in Fig. 16 for 5 channels and in Fig. 17 for 10 channels, while $N \in$

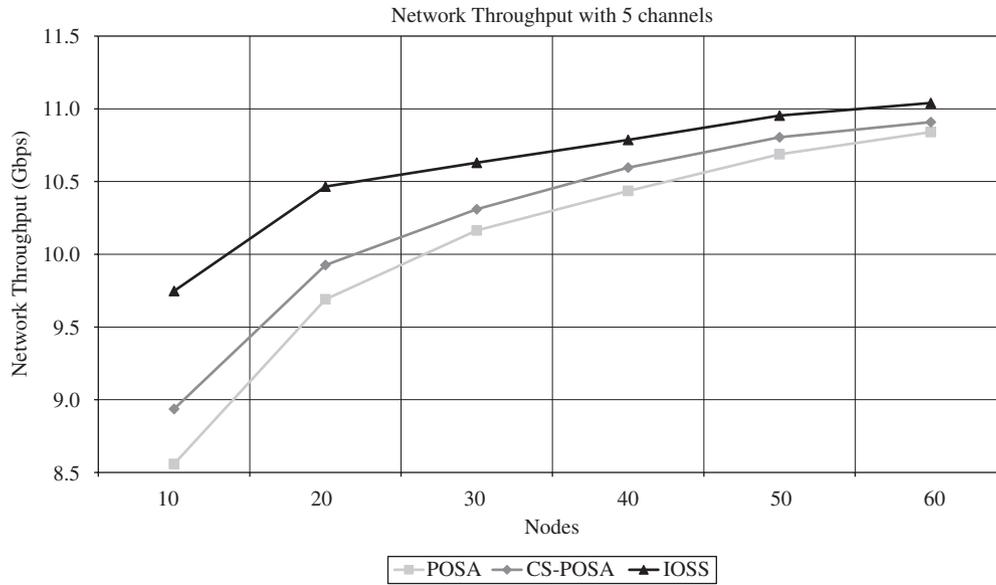


Fig. 16. Network throughput with five channels.

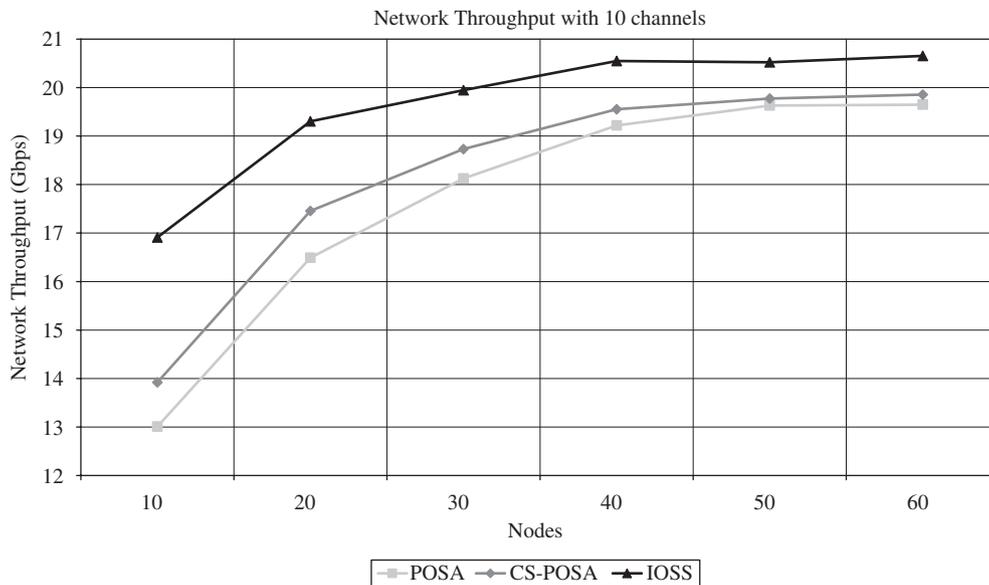


Fig. 17. Network throughput with 10 channels.

{10, 20, 30, 40, 50, 60} and $K = \text{FLOOR}(NW/5)$. The theoretical maximum throughput is $5 \times 2.4 = 12$ Gbps for 5 channels and $10 \times 2.4 = 24$ Gbps for 10 channels. IOSS achieves higher throughput than POSA and CS-POSA for both 5 and 10 channels. The throughput improvement of IOSS algorithm proves that its scheduling length is shorter than POSA and CS-POSA. It also explains that IOSS allows data to be transmitted at the same time faster than POSA and CS-POSA. In particular, IOSS improves the network throughput at least to 200 Mbps (60 nodes) and at the most by 1189 Mbps (10 nodes), as compared to POSA. As compared to CS-POSA for five channels it enhances network throughput at least by 130 Mbps (60 nodes) and at

the most by 812 Mbps (10 nodes). For 10 channels, IOSS increases, in comparison with POSA, its network throughput, which reaches 3903 Mbps at the most for 10 nodes and 895 Mbps at least for 50 nodes. For the same number of channels, IOSS achieves maximum difference equal to 2988 Mbps for 10 nodes and minimum difference equal to 748 Mbps for 50 nodes, as compared to POSA.

6.3. Network throughput versus mean packet delay as nodes vary

Fig. 18 depicts the network throughput versus mean packet delay for 5 channels and Fig. 19 illustrates the

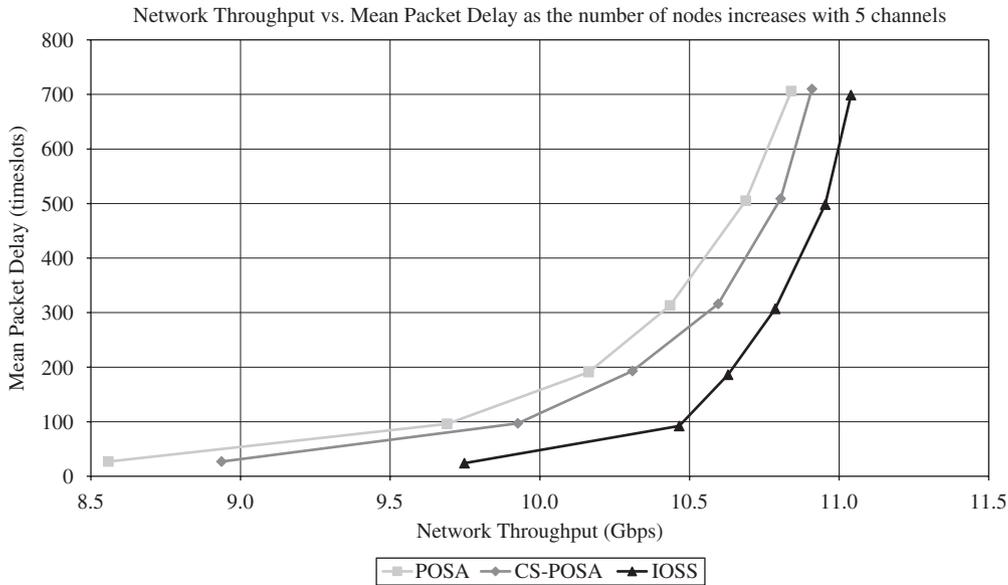


Fig. 18. Network throughput versus mean packet delay as the number of nodes varies with five channels.

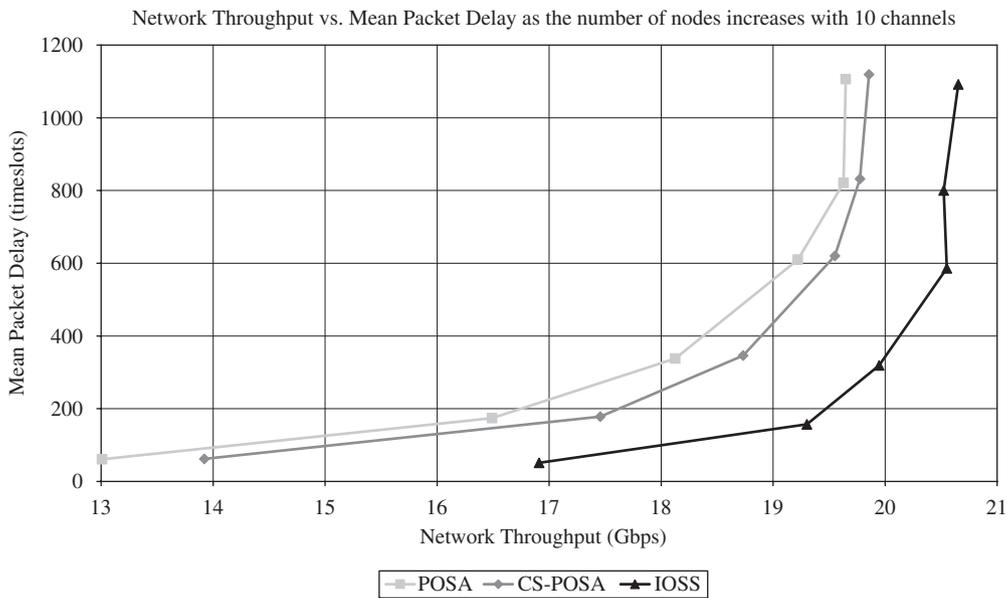


Fig. 19. Network throughput versus mean packet delay as the number of nodes varies with 10 channels.

network throughput versus mean packet delay for 10 channels, while $N \in \{10, 20, 30, 40, 50, 60\}$ and $K = \text{FLOOR}(NW/5)$. It is obvious that IOSS excels in network throughput for any combination of nodes and channels. At the same time, IOSS involves less delay than POSA and CS-POSA. For instance, when $N = 30$ and $W = 5$, the network throughput produced by IOSS is 10.6 Gbps with 186 timeslots as mean packet delay, while POSA produces 10.1 Gbps with 191 timeslots and CS-POSA offers 10.3 Gbps with 193 timeslots. Another example is set; when supposing $N = 20$ and $W = 10$, IOSS outputs 19.3 Gbps and the packets wait 157 timeslots in the waiting queues, while POSA offers 16.4 Gbps with 174 delay slots

and CS-POSA produces 17.4 Gbps with 178 slots. It is clear from both figures that IOSS offers better throughput with less delay, because it serves first the highly demanding requests for each node. In this way, long time packets wait less and the mean waiting time is a little reduced.

6.4. Network throughput versus mean waiting time as load varies

Once again, the relation between throughput and delay is analyzed on two specific network models. The first one includes 30 nodes and 5 channels, while 10 different values of K (10, 20, 30, 40, 50, 60, 70, 80, 90, and 100) are attached

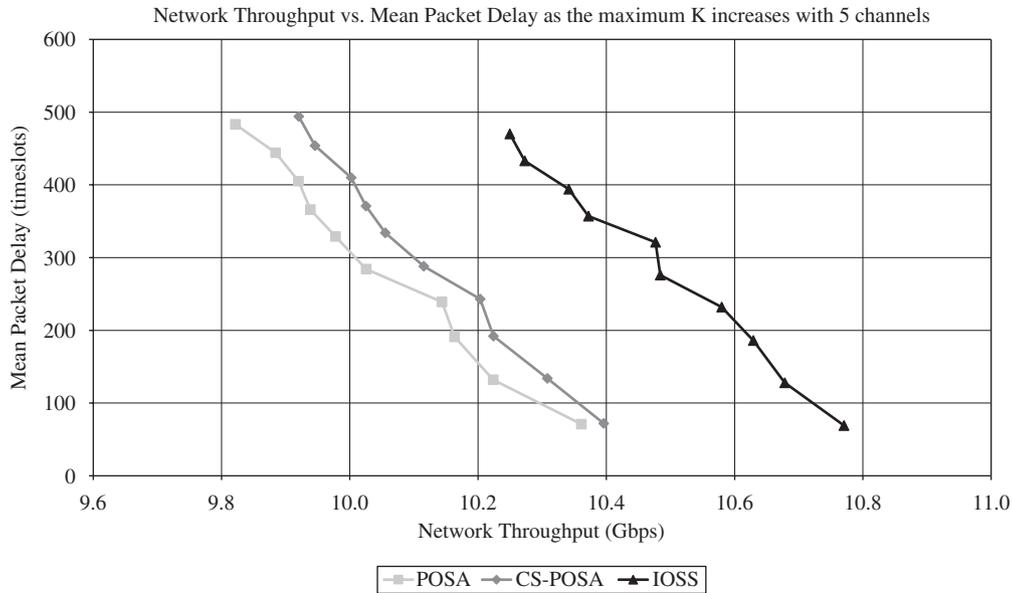


Fig. 20. Network throughput versus mean packet delay as the maximum entry of the demand matrix (K) varies with five channels.

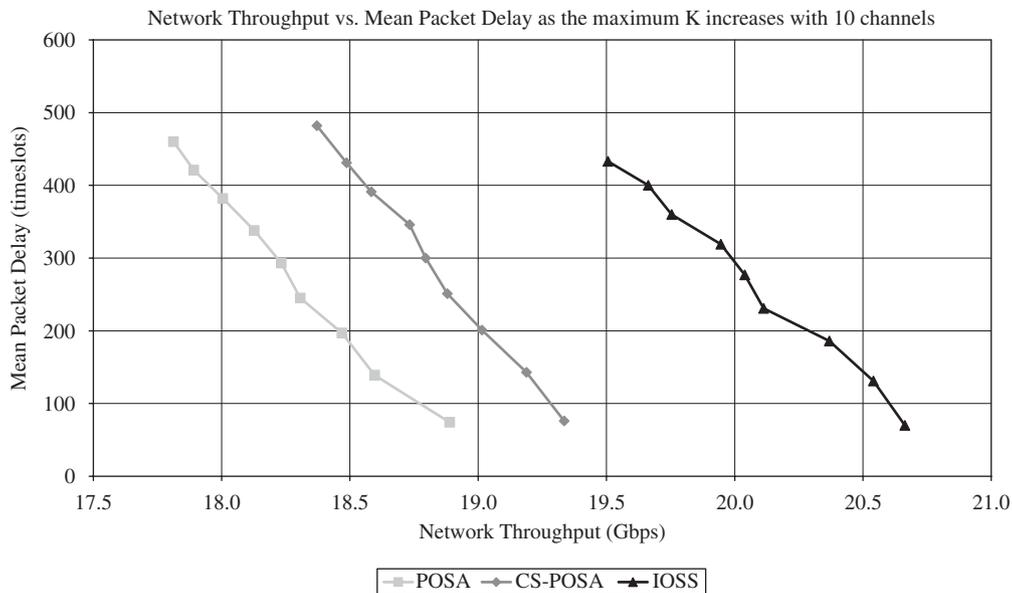


Fig. 21. Network throughput versus mean packet delay as the maximum entry of the demand matrix (K) varies with 10 channels.

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 figures (Fig. 20 for 5 channels and Fig. 21 for 10 channels) that the throughput (for any algorithm) decreases, as K increases. This phenomenon is not often observed in the category of the networks examined. Nevertheless, it appears in all algorithms examined, OIS [19], POSA [20], or CS-POSA [21], due to the architecture of the protocols. When the workload increases, it signifies that the sizes of packets that arrive at the nodes are actually increasing. This is indicated by the increasing of the maximum value of K . When K increases, it is difficult for the scheduling algorithm to find open space in the

constructed scheduling matrix. If there was an open space of 9 slots in the constructed scheduling matrix and the packet which arrived was of 10 timeslots size-duration of transmission, then the algorithm could not be able to break it in pieces. It could then place it at the end of the matrix, where there would be available space for a packet of 10 timeslots. This leads to a decrease of channel utilization as the timeslots not used increase and the throughput decreases. Nevertheless, IOSS still maintains higher throughput and less delay for any value of load. For example, when $K = 20$ and $W = 10$, IOSS produces 1.95 Gbps more than POSA and 1.35 Gbps more than CS-POSA, while IOSS reduces the waiting slots by 8, as compared to POSA,

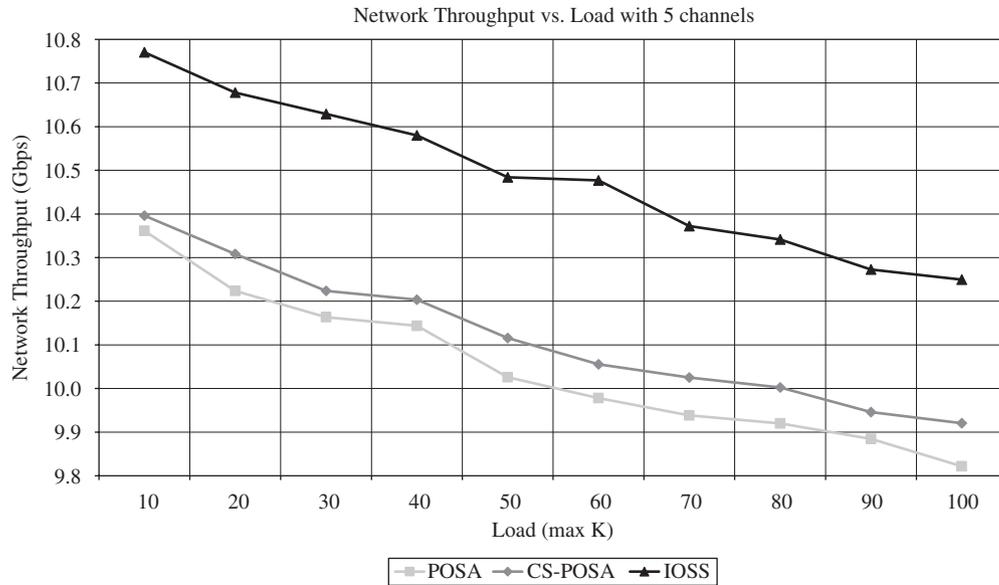


Fig. 22. Network throughput versus load with five channels.

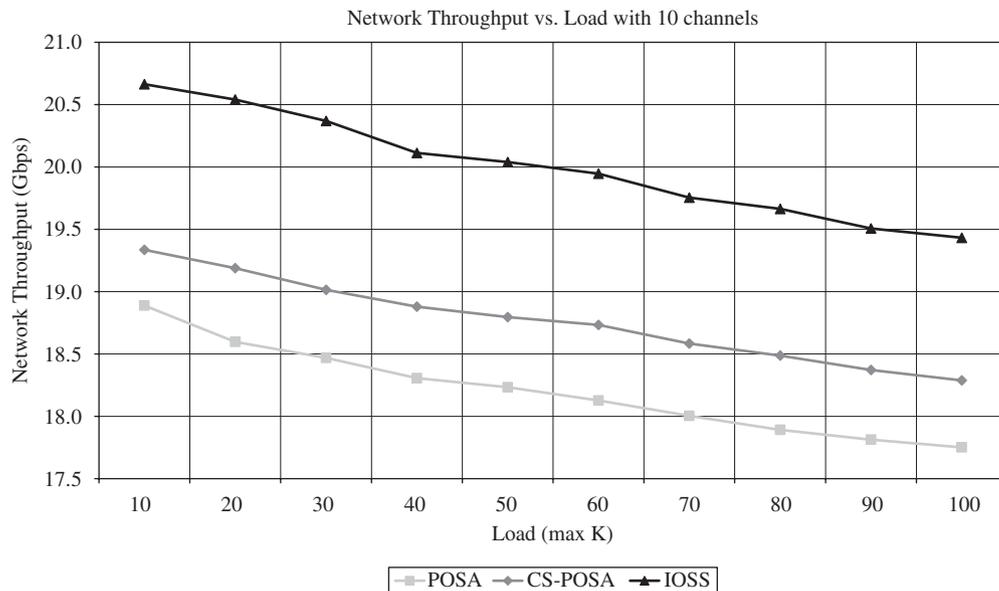


Fig. 23. Network throughput versus load with 10 channels.

and by 12, as compared to CS-POSA. An additional example is given, when $K = 80$ and $W = 5$, IOSS offers 421 Mbps more than POSA and 339 Mbps more than CS-POSA. At the same time, IOSS reduces the waiting time of the packets by 13, as compared to POSA, and by 24, as compared to CS-POSA.

6.5. Network throughput versus traffic load

Finally, the relation between throughput and load is shown in Fig. 22 with 5 channels and in Fig. 23 with 10 channels as K varied from 10 to 100 (10, 20, 30, 40, 50, 60, 70, 80, 90, and 100). Overall, IOSS provides the best

output throughput for any K and for any W . The maximum difference from POSA, when $W = 5$, seems to appear when $K = 60$ and it is 500 Mbps and the maximum difference from CS-POSA seems to be 421 Mbps when $K = 60$. The maximum difference from POSA, when $W = 10$ and $K = 20$, amounts to 1.95 Gbps, and 1.53 Gbps when $K = 30$.

7. Conclusions

This paper considered a single-hop WDM optical LAN with broadcast and select architecture, providing access to transmission nodes without collisions. Each node is

Table 2
A brief comparison between OIS, POSA, CS-POSA, and IOSS

	OIS	POSA	CS-POSA	IOSS
Scheduling algorithm complexity	$O(W^2KN)$	$O(W^2KN)$	$O(W^2KN) + O(\log N/W)$	$O(W^2KN) + O(K)$
Prediction mechanism complexity	–	$O(K + 1 + V)$	$O(K + 1 + V)$	$O(K + 1 + V)$
Channel utilization	Low	Low	Medium	High
Network throughput	Medium	High	High	High
Schedule efficiency	Low	Low	Medium	High
Packet delay	Low	Low	Medium	Low

N denotes the number of nodes, W denotes the number of channels, K denotes the maximum length of packet, and V denotes the size of the history queue where the real requests of the nodes are stored (in the prediction mechanism). The complexity is calculated using p processors.

equipped with a tunable transmitter and a fixed receiver and a passive star connects all nodes together. The suggested scheduling algorithm is pre-transmission coordination-based and uses control packets, instead of extra control channels, for the arbitration of the nodes. Two common scheduling strategies were proposed in order to change the service sequence of the transmission requests in descending order, without a sorting. The novel strategies add no extra complexity and run at better times than the scheduling basis algorithm of OIS. Also, the prediction mechanism of POSA is adopted in order to reduce the amount of time spent in computing the schedule by predicting traffic requests. The simulation results show that if the hard-traffic requests are given priority the network effectiveness gains in terms of channel utilization and the schedule is much shorter. Finally, the packets wait at the waiting queues less than POSA and CS-POSA and the relation between the network throughput and the mean packet delay is improved (Table 2).

References

- [1] Bernstein G, Rajagopalan B, Saha D. Optical network control: architecture, protocols, and standards. Reading, MA: Addison-Wesley; 2003.
- [2] Green P. Progress in optical networking. IEEE Commun Mag 2001; 39(1):54–61.
- [3] Papadimitriou GI, Papazoglou C, Pomportsis AS. Optical switching: switch fabrics, techniques, and architectures. IEEE/OSA J Lightwave Technol 2003;21(2):384–405.
- [4] Yao S, Mukherjee B, Dixit S. Advances in photonic packet switching: an overview. IEEE Commun Mag 2000;84–94.
- [5] Mukherjee B. WDM optical networks: progress and challenges. IEEE J Sel Areas Commun 2000;18(10):1810–24.
- [6] Murthy CSR, Gurusamy M. WDM optical networks: concepts, design, and algorithms. Englewood Cliffs, NJ: Prentice-Hall; 2002.
- [7] Shepherd FB, Vetta A. Lightning fibers and a dark network. IEEE J Sel Areas Commun 2004;22(9):1583–8.
- [8] Wang B, Hou JC, Han C-C. Design and analysis of a channel access protocol for WDM-based single-hop optical networks. Photonic Network Commun 2003;6(3):289–308.
- [9] Sivalingam KM, Subramaniam S. Emerging optical network technologies: architectures, protocols and performance. Berlin: Springer; 2004.
- [10] Mukherjee B. Optical WDM networks. Berlin: Springer; 2006.
- [11] Chan C, Sherman KL, Zimgibl M. A fast 100-channel wavelength-tunable transmitter for optical packet switching. IEEE Photonics Technol Lett 2001;13(7):729–31.
- [12] Shrikhande K, et al. Performance demonstration of a fast tunable transmitter and burst-mode packet receiver for HORNET. In: Optical fiber communications conference. Anaheim, CA, 2001.
- [13] Bogineni K, Dowd PW. A collisionless multiple access protocol for a wavelength division multiplexed starcoupled configuration: architecture and performance analysis. IEEE/OSA J Lightwave Technol 1992;10(11):1688–99.
- [14] Chen MS, Dono NR, Ramaswami R. A media access control protocol for packet switched wavelength division multi-access metropolitan area networks. IEEE J Sel Areas Commun 1990;8(6): 1048–57.
- [15] Chipalkatti R, Zhang Z, Acampora AS. Protocols for optical star-coupled network using WDM: performance and complexity study. IEEE J Sel Areas Commun 1993;11(4):579–89.
- [16] Bogineni K, Sivalingam KM, Dowd PW. Low-complexity multiple access protocols for wavelength division multiplexed photonic network. IEEE J Sel Areas Commun 1993;11(4):590–603.
- [17] Borella MS, Mukherjee B. Efficient scheduling of nonuniform packet traffic in a WDM/TDM local lightwave network with arbitrary transceiver tuning latencies. IEEE J Sel Areas Commun 1996;14(5): 923–34.
- [18] Ganz A, Gao Y. Time-wavelength assignment algorithms for high performance WDM star based systems. IEEE Trans Commun 1994;42(2–4):1827–36.
- [19] Sivalingam KM, Wang J, Wu J, Mishra M. An interval-based scheduling algorithm for optical WDM star networks. J 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. J Photonic Network Commun 2001;3(3):271–86.
- [21] Sarigiannidis PG, Papadimitriou GI, Pomportsis AS. A new prediction and channel sorting based scheduling algorithm for WDM star Networks. In: Enschede, The Netherlands, Proceedings of the 12th annual symposium of the IEEE/CVT; 2005.
- [22] Sarigiannidis PG, Papadimitriou GI, Pomportsis AS. CS-POSA: a high performance scheduling algorithm for WDM star networks. Photonic Network Commun 2006;11(2):209–25.
- [23] 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.
- [24] Sarigiannidis PG, Papadimitriou GI, Pomportsis AS. WFF: a high performance scheduling algorithm for WDM star networks that minimizes idle timeslots. In: IEEE SCVT'05, Enschede, The Netherlands, November 2005.
- [25] Papadimitriou GI, Tsimoulas PA, Obaidat MS, Pomportsis AS. Multiwavelength optical LANs. New York: Wiley; 2003.