A High-Throughput Scheduling Technique, With Idle Timeslot Elimination Mechanism

Panagiotis G. Sarigiannidis, *Student Member, IEEE,* Georgios I. Papadimitriou, *Senior Member, IEEE,* and Andreas S. Pomportsis

***Abstract*—A new media-access-control protocol is introduced in this paper. The authors consider a wavelength-division-multiplexing (WDM) network with star topology. A single-hop WDM system is considered, so that there is a full connectivity between every node-pair in just one hop. The protocol adopted is pretransmission coordination-based, so the protocol coordinates nodes before the actual transmission. The coordination is achieved with one demand (or traffic) matrix, which saves the predetermination of the timeslots each node transmits. Each transmission frame (or cycle) has two phases: the control phase and the data phase. In order to eliminate the possible delay added by the schedule computation between the two phases of each frame, they consider a traffic prediction scheme, which is based upon the hidden Markov chain model. The control phase functions as a learning period in which the predictor is trained. The training is based on the traffic of the network. During the data phase, each station transmits its packets based on the predicted reservations, which are the predictor's output. In the same frame, the predictor computes the reservations for the next frame. They show that their protocol, although suffering from small packet delay loss, introduces a new method of computing the reservations of the demand matrix and brings some performance improvement in terms of channel utilization and results in higher network throughput, which is proven by extensive simulations.**

***Index Terms*—Demand matrix, optical wavelength-division-multiplexing (WDM) networks, reservation, scheduling, traffic prediction.**

I. Introduction

IN THE world of networking, with enormous requirements in speed, only optical technology can correspond with effectiveness because of the incapacity of electronic switching. The development of high-level optical networks is employed by a wavelength-division-multiplexing (WDM) technique to achieve tremendous capacities [1]–[5]. The great thing about WDM networking, which explains the huge research and commercial interest in this technology, is its compliance with the limited, compared to fiber technology, speed of the stations' electronic circuits [6], [7]. In this way, WDM offers an excellent way of exploiting the huge bandwidth of optical fibers by introducing concurrence among multiple users transmitting. The main attraction of optical switching is that it enables routing of optical data signals without the need for conversion to electrical signals and, therefore, is independent of data rate and data protocol. The multiplexing and the demultiplexing of the available channels (or wavelengths) are performed with optical implementation, resulting in gigabit data rates [8]. WDM networks have been considered for local area network (LAN), metropolitan area, and wide area networks. If we specialize in the optical WDM networks, then it is obvious that a list of variable classes can be found. These classes are WDM point to point link networks, WDM broadcast and select networks, wavelength routing networks, and passive optical networks [9]. Among all these classes, the broadcast and select category seems to achieve a significantly higher performance than the others [10]. Regarding the physical topology of the broadcast and select networks, we could report the four most popular topologies, which are star, bus, tree, and ring [11]. The passive star topology using a broadcast and select star coupler has been shown to dominate in LAN area, due to its single point of failure, its completely passive nature, its direct optical connection of all node pairs, and its slight decrease in the signal power [12]. Networks of this category are equipped with one optical passive star coupler, which connects all nodes via two-way optical fibers. If a network consists of N nodes then the star coupler has N inputs and N outputs. If a node wishes to transmit data to a (different) destination node of the network, it sends the optical signal, with the aim of a given available channel to the passive star. The inserted signal will have its power equally divided among all output ports of the star coupler, which will appear exactly on the same wavelength as it entered. This is due to the fact that the star coupler is generally just a passive device, without power, and so, we manage it with reliability and easiness [6], [9]. This procedure is the first part of the broadcast and select characterization of the WDM network architecture. The second procedure includes the reception of the signal form each node by filtering. A multicast service takes place when more than one node choose to accept the transmitted signal, and a unicast service takes place when only one node chooses to accept the transmitted signal, while the rest of the nodes reject it. Another fundamental distinction is between singlehop and multihop systems [13]. If the actual number of intermediate nodes that a data or control packet needs to pass through, in order to reach its final destination, is exactly one, then the system is called single hop; otherwise (more than one intermediate nodes), it is called multihop. Another point of differentiation among broadcast and select WDM networks is the number and type of transmitters and receivers assumed for each node in the system [14]. In the most cases, each node has one transmitter and one receiver. A fixed-tuned transmitter (FT) is capable of transmitting on a specific wavelength only.

Manuscript received March 29, 2006; revised August 25, 2006.
The authors are with the Department of Informatics, Aristotle University, 54124 Thessaloniki, Greece (e-mail: gp@csd.auth.gr).
Digital Object Identifier 10.1109/JLT.2006.884992

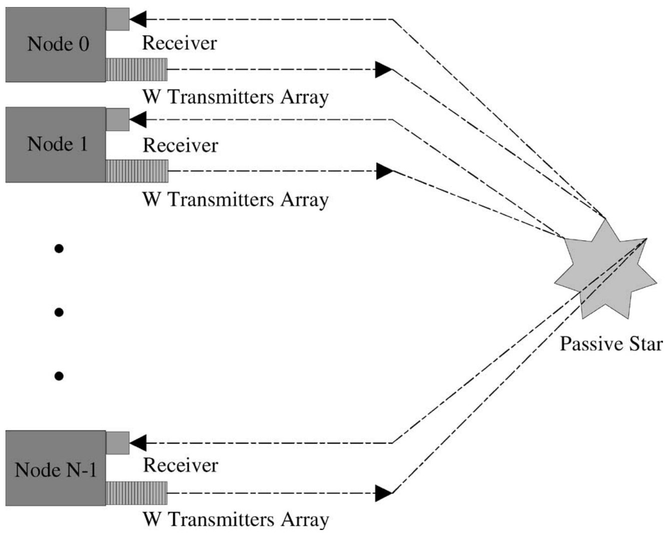

Fig. 1. Broadcast and select network with N nodes and W channels. Each node has a TT and a fixed receiver.

A fixed-tuned receiver (FR) is capable of receiving on a specific wavelength only. Consequently, an FT node can send data only to nodes, which are capable to filter the transmitted wavelength. A tunable transmitter (TT) is configurable to change in order to transmit in all the available channels of the network. A tunable receiver (TR) is configurable too, returning receiving in all the available channels. As a conclusion, we could refer four possible configurations for a WDM broadcast and select system: an FT–FR system, an FT–TR system, a TT–FR system, and a TT–TR system.

This paper focuses on a single-hop TT–FR WDM broadcast and select system with N nodes and W channels (Fig. 1). A new scheduling algorithm is presented and is called wait for fullness (WFF). WFF introduces a new schedule computation mechanism, which is called the cleanup mechanism. The innovation of WFF lies in the way it modifies the scheduling algorithm, so that it minimizes (or even eliminates) the number of idle timeslots, thus significantly improving channel utilization while maintaining a high network throughput. WFF incorporates the same prediction mechanism as predictive online scheduling algorithm (POSA) which, when combined with the pipelining of the schedule computation time and transmission time, explains the significant performance improvement that is observed. The adopted network encloses N nodes, which can transmit in W channels. The data transmission is accomplished by an array of fixed transmitters (or a TT), which can be tuned in each of the W channels. Also, the data reception is accomplished by a fixed receiver, tuned to a specific channel, and allows the node to accept data in the specific wavelength. Each node can receive data on a dedicated channel, which is referred to as the node's home channel. Thus, the network can support three different connections: 1) broadcasting, when the transmission targets all the channels of the network; 2) multicasting, when the transmission targets a part of the channels of the network; and 3) unicasting, when the transmission targets only one node of the network [3], [5]. The connection of the stations is accomplished through a passive star coupler that facilitates the transfer of data from the transmitters to the receivers. Each output port of the star coupler is connected to the corresponding receiver by means of a two way optical fiber.

The rest of this paper is organized as follows: Section II analyzes the architecture of the network, Section III describes the distinction among the media-access control (MAC) protocols, while Section IV outlines the structural elements of the network. Section V analyzes the three online protocols [online interval-based scheduling algorithm (OIS), POSA, and check and sort POSA (CS-POSA)] with the previous progress and work to be improved. Section VI presents the new algorithm and is followed by the figures and the detailed comparisons between the performances of the three algorithms, POSA, CS-POSA, and WFF in Section VII. Finally, concluding remarks are presented in Section VIII.

II. Network Behavior

It is be very useful for the study of the behavior of the network, if we also examine the arithmetic relation between the nodes and the available channels. If we consider that the channels of the network are more than the nodes $(W > N)$, then it is clear that during the data transmission a number of channels remain inactive, since each node accepts data only in a predetermined channel. Practically, this kind of network structure has no meaning. The second case concerns networks where the number of nodes and available channels are equal $(N = W)$. In this case, we have a type of ideal communication, optical self-routing is achieved [15], since each node sends its data to a separate channel of communication. Of course, we have to exclude the case where two nodes tune concurrently at the same channel in order to transmit data to the same destination node. This parallel transmission at the same time causes channel conflict, and the data is destructed. This case is idealized when each node transmits simultaneously in different channel in the network. Naturally, the implantation of systems that support an equal number of users and channels is a complicated and costly process. The most usual relation between the nodes and the channels is where the nodes are more than the channels $(N > W)$. This case is frequent in the local networks that we examine. Here, therefore, the channels are limited; consequently, each node is forced to share a number of channels with certain other nodes. More specifically, if we suppose that we allocate N nodes and W channels, then a number of nodes equal to N/W share the same channel. For example, if we allocate ten nodes and two channels, then five nodes share the same number of channels. It is obvious that the lack of channels leads to unavoidable time delay of packets, since certain nodes wait in order to transmit without collisions or certain nodes are supposed to retransmit the same data if some conflict happens. Finally, we should report the case where the network suffers from available channels $(N \gg W)$. It is easily understood that the network presents a lot of delays in the packet switching, and it requires very organized coordination between nodes and channels. In this paper, we consider that we have more nodes than channels. In this case, we need a common strategy implemented in the network and adopted by all nodes in order to communicate with each other.

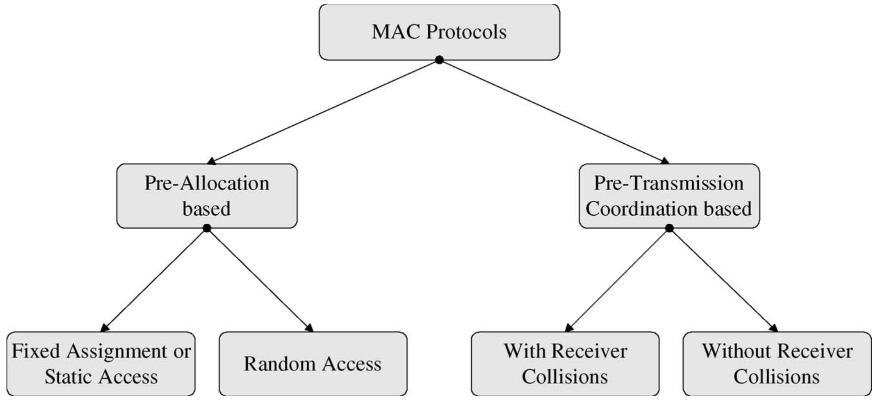

Fig. 2. Categorization of MAC protocols.

III. MAC Protocols

A MAC protocol comprises all the methods and the ways of accessing the available media [16]. In simple terms, a good protocol is essential to coordinate transmissions between various nodes in the network. Assuming a TT–FR system, it is possible that two or more nodes might transmit on the same channel at the same time. Then, a channel collision occurs and the network performance is degraded. A MAC protocol usually aims to prevent channel collisions form occurring and specify how selections should be made when necessary. The two major categories of protocols are the preallocation based and the pretransmission coordination based (Fig. 2). The basic distinction among MAC protocols is the existence of a control channel [17]. In the case where in the network there is at least one channel dedicated to the coordination of channels and their transmission time, then the protocol is pretransmission coordination. In preallocation schemes, the available wavelengths are used only as data channels, and no wavelength serves as a shared control channel. The wavelengths are preassigned to transmitters or receivers in a fixed manner, and each wavelength comprises the so-called home channel of the node to which it is preallocated. It must also be noted that in the absence of a dedicated control channel, transmission control can be performed using control packets. There are two quite common home-channel allocation schemes [18]:

1) for interleaved allocation scheme $\lambda_i = i \bmod W$;
2) for neighbor allocation scheme $\lambda_i = \lfloor i/\lceil N/W \rceil \rfloor$

where node i is assigned with a channel λ_i as its home channel, where $0 \leq i \leq N-1$, and $0 \leq \lambda_i \leq W-1$. Thus, assuming for example TT–FR node architecture, a source node determines the home channel of a destination node through the destination node number (i), the number of nodes (N), and the number of the total available wavelengths (W).

The category of preallocation-based protocols can be further divided into fixed-assignment protocols and random access protocols [19]. Fixed-assignment (or static access) schemes predetermine the exact time instants the nodes are allowed to access the shared channels. On the other hand, random access protocols allow nodes to access the shared channels in a random way. In the first kind, channel collisions are avoided, while in the second, there is the possibility of the existence of collisions, whose frequency depends on the offered traffic. The other category of pretransmission coordination-based protocols [20] differs from preallocation ones in the fact that they designate at least one wavelength to be used as a control channel. This set of control channels, which is in the most cases only one, is used to coordinate access to the remaining data channels. As we can note, in this category, data transfer actually consists of two independent phases. During the first phase, known as the reservation phase, the MAC protocol coordinates the data to the appropriate channels in order to prepare the transmission of the next phase. During the second phase, known as data transmission phase, the actual transmission of the data is performed, based on the coordination assignments of the reservation phase. Channel collisions may be allowed to occur in both phases, either in control or data channels. In this point, one more differentiation can be recorded under the criterion of the existence of collisions. If we assume TRs at nodes for data reception, then it is possible that two or more nodes transmit to the same destination node at the same time on different wavelengths. In that case, we say that a receiver collision occurs. Therefore, if the pretransmission protocol allows receiver collisions, then the protocol is known as pretransmission coordination based with receiver collisions; otherwise, the protocol is known as pretransmission coordination based without receiver collisions [21]. In this paper, we focus on pretransmission coordination based without receiver collisions, due to the consideration of TT–FR system. This choice can be justified by three reasons: First, the pretransmission-based schemes support better utilization of the available channels and leads to higher performance levels; second, these schemes result in a collision-free system, as they avoid channel (by channel coordination) and receiver (by a TR system) collisions; finally, they offer more flexibility for a large number of nodes [22].

IV. System Architecture

As mentioned, the scheduling algorithm does not use a centralized scheduling scheme to decide how transmissions and receptions should be made. Besides, in centralized scheduling schemes, the star coupler plays a complex role, since the passive star includes a scheduler as well. Typically, this scheduler accepts requests from nodes and puts them in communication order. In this paper, we consider a distributed scheduling algorithm, in which each node has to maintain a global status information for each cycle of the reservation and transmission phases. In other words, each node updates its common information, according to information obtained by the shared channel. A very common global information [23] is the $N \times W$ traffic or demand matrix $D = [d_{n,w}]$. Each row n represents the node and each column w the channel. Furthermore, time is divided in timeslots. Each cycle of the reservation phase followed by a data phase is called frame. The frame then stores for every node the number of timeslots required for transmission to a specific channel. Then, the nodes transmit the requested data during the current frame at different moments. For example, if we assume for a specific cell of demand matrix $d_{1,2}$ a time amount of five timeslots ($d_{1,2} = 5$), then the node n_1 for channel w_2 demands five timeslots for transmission for the next frame. It is supposable that the size of frames is not stable but depends on the total requests of the nodes of the network. Alternatively, the size of each transmission frame is variable and depends on the load of the network. It is obvious that a demand matrix with minor mean timeslots requests produces a short frame. On the contrary, a heavy traffic produces a long frame.

Scheduling algorithms can generally be classified as offline and online [24]. An online scheduling algorithm computes a schedule based on available partial information. Offline schemes do not compute the schedule until the entire demand matrix is known. Offline algorithms [25] have the advantage of efficiency, since they get a better and overall view of the total requests of all nodes. On the other hand, online algorithms typically require less computation time, and so, there are reduced delays between reservation and transmission. We use an online schedule matrix to carry out a more simple algorithm. Also, a further drawback of offline algorithms is their high complexity, which varies from $O(N^2W)$ to $O(N^3W^3)$ [18]. The high levels of complexity do not conform to optical fiber technology and lead to respective long time delays. Besides, these high levels of complexity cannot be compensated by sophisticated hardware, because of the immaturity of optical components. Some examples of offline algorithms are satellite-switched/time-division multiple access [26], MULTI-FIT [27], single reservation algorithm (SRA) [28], and transmission assignment algorithm (TAA) [29].

V. Previous Online Algorithms

Three online algorithms are analyzed in this section: the OIS [30] algorithm, the POSA [31] algorithm, and the CS-POSA algorithm [32], [33]. OIS is a very simple online algorithm, which exploits the advantages of the algorithms that do not need the entire demand matrix but a part of it, because it starts computing the schedule as soon as the first node's set of requests has been known. Execution of the online scheduling algorithm begins upon reception of the first request, and so, the entire demand matrix is not required. This scheme incorporates online scheduling on the basis of available time intervals on channels and for each examined node that requests reservation. All requests of each node are examined sequentially one by one. The main idea of OIS is the maintaining of two sets of intervals for each frame. The first set consists of the list of available intervals for each channel, and the second one consists of the intervals of the node, whose reservation is currently being scheduled. The intervals show the unallocated time on a specific channel or node. Whenever a node sends timeslots requests for transmission, the list of available intervals are checked for their availability of the requested number of free timeslots on the channel. Since OIS is a collision free protocol, the algorithm has to prevent the schedule of two or more nodes on the same channel at the same time.

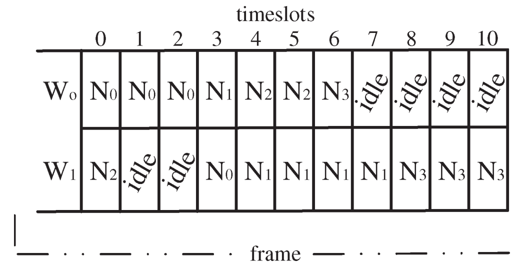

Fig. 3. Final schedule of OIS/POSA, according to demand matrix D.

Assume the following demand matrix D 4×2, with four nodes (N_0, N_1, N_2, N_3) and two transmission channels (W_0, W_1). The set of rows represents the nodes of the network, and the set of columns represents the channels of the network

$$D = \begin{bmatrix} 3 & \dots & 1 \\ 1 & \dots & 4 \\ 2 & \dots & 1 \\ 1 & \dots & 3 \end{bmatrix}.$$

The algorithm does not require knowledge of the rows of all four nodes in order to function but only the ones of N_0 [3,1] (row 1). In Fig. 3, the final schedule can be observed, constructed by OIS for demand matrix D. The frame examined lasts 11 timeslots. The number of timeslots, in which the channels remain idle, determines the performance of the algorithm. It is obvious that channel W_0 transmits data for seven continuous timeslots, while the rest of the four remain idle. Channel W_1 transmits for nine timeslots and remains idle for the other two. Conclusively, we can mention that OIS is defined by three advantages [30]:

1) simplicity in computation actions of the schedule matrix;
2) low complexity, linear with the number of nodes and is given by $O(NW^2K)$;
3) immediate initiation of the construction of the schedule matrix, which verifies its online identity.

These three elements of OIS are set as targets, which should also complement the algorithm which is proposed in this paper, as far as the construction of the schedule matrix is concerned.

The POSA [31] is a very powerful scheduling tool as having a very effective traffic prediction mechanism. More specifically, POSA is an extension of OIS and has a main aim to reduce drastically the computation time of the schedule. While OIS processes the requests for the transmission of all the nodes in the reservation phase, and then it switches to the data phase, by realizing the transmissions that have been registered in the reservation phase, POSA maintains a very powerful prediction system according to the history of the recent actual reservations. In this way, in each frame, it predicts the requests of each node for the following frame, and simultaneously, it transmits

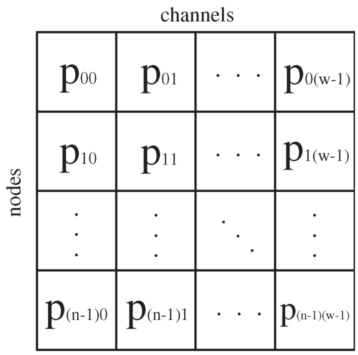

Fig. 4. Form of the predicted demand matrix.

according to the predictions of the previous frame. At the same time, it inputs the actual requests of the current frame to the history queues of the predictor, whose operation is mainly based upon the hidden Markov chain model. However, we have to mention that POSA uses the same scheduling algorithm with OIS but with one difference: OIS constructs the final schedule matrix based on the actual requests of each node, in contrast to POSA that computes the schedule matrix based on the predicted requests, as output of the traffic prediction system. POSA operates in three phases. During the first phase, the algorithm remains in a training condition state. At the beginning, all the nodes broadcast their reservation requests, and the transmission schedule matrix is computed. At the same time, the requests of the nodes are inserted as learning information in the predictor so that it forms suitable history queues for the following prediction phase. The second phase of POSA is just a switching phase. The algorithm stops learning and starts predicting. Finally, POSA enters the most important phase: the prediction phase. Assuming that the network consists of N nodes and W channels, the reservations arrive at the algorithm in the form of the demand matrix $D = [d_{n,w}]$. The aim of the predictor is to predict the contents of the matrix D for each line $n \in N$ and for each column $w \in W$. Thus, the predictor can be divided into $N \times W$ independent predictors $p_{i,j}(0 \leq i \leq N-1, 0 \leq j \leq W-1)$. Each predictor outputs a value form 0 to K, where K denotes the upper bound on a node's request on a channel (Fig. 4). K is a constant value, so it is possible to construct a probabilistic-based deterministic predictor. See [31] in order to find a complete description of the predictor. Of course, if we assume that eight predictors produce the following predicted demand matrix D (the same as OIS), then the final schedule will be the same as OIS's final schedule (Fig. 3), due to the fact the both OIS and POSA use the same scheduling algorithm. Conclusively, we can say that POSA is defined by four basic characteristics [31]:

1) the prediction mechanism;
2) the pipelining of the schedule computation phase of the frame with the reservation and the data transfer phases;
3) an adjusting period of the algorithm so that the history queues of each algorithm is filled in;
4) an overall complexity of the prediction system of

$$O\left[\frac{(K+1+V)(NW)}{P}\right]$$

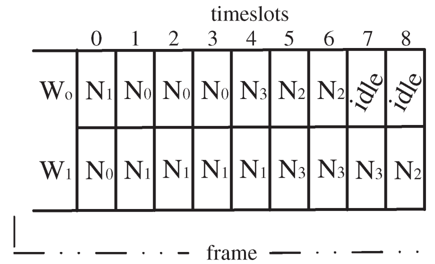

Fig. 5. Final schedule of CS-POSA according to demand matrix D.

(i.e., $P = (NW/p)$), where p is a constant, which shows that the entire algorithm can be considered to run in constant asymptotic time. The above four characteristics are set as targets that should accompany the new proposed algorithm too.

The CS-POSA [32], [33] protocol is an improved edition of POSA. On the one hand, it retains the same algorithm that constructs the schedule matrix as OIS, and on the other hand, it adopts the prediction mechanism of the requests of each node as it has been presented above. CS-POSA introduces the operation of a new function before the construction of the schedule matrix. More specifically, after it has accepted the prediction requests from the prediction mechanism, it makes two continuous steps: First, it adds the total requests of each node for each channel and enters them in a vector. This vector is the traffic picture of each node and is the key point of CS-POSA. Then, it sorts the vector in a declining order so that in the first place, there are the requests of the node with the biggest total of requests and, in the last place of the vector the one with the least total of requests. After this procedure is completed, the same algorithm with OIS is applied, starting from the node with the most requests in all channels and finishing with the node with the least requests. In order to understand better the operation of CS-POSA, a specific example is examined. The following traffic matrix has been constructed by eight individual predictors (we consider the same demand matrix as this in OIS). Before CS-POSA constructs the schedule matrix, it adds each row of the traffic matrix D in a new vector S that will register the total amount of requests by each node:

$$D = \begin{bmatrix} 3 & \dots & 1 \\ 1 & \dots & 4 \\ 2 & \dots & 1 \\ 1 & \dots & 3 \end{bmatrix}, \quad S = \begin{bmatrix} 3+1=4 \\ 1+4=5 \\ 2+1=3 \\ 1+3=4 \end{bmatrix}, \quad S' = \begin{bmatrix} 5 \\ 4 \\ 4 \\ 3 \end{bmatrix}.$$

Therefore, vector S consists of the total amount of the requests of the four nodes for the two transmission channels. Table S is a mirror of the activity that each node has. After this, 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. In this way, vector S changes in the ordered vector S'. That denotes that the requests of node N_1 is examined first, then those of node N_0, then those of node N_3, and finally those of node N_2. For the same demand matrix, CS-POSA constructs the following schedule matrix (Fig. 5). It is clear that the schedule matrix of CS-POSA spends a total of nine timeslots from which two out of 16 subtimeslots, i.e., a percentage of 12.5% is wasted.

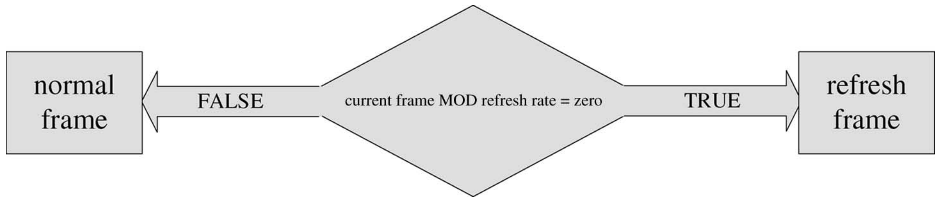

Fig. 6. Frame condition, which resulting a normal or a refresh frame.

VI. WFF Algorithm

As we examined in the previous section, OIS and POSA use exactly the same scheduling algorithm. This operation, even if it is very simple and at the same time it allows the creation of the schedule without complicated processes, may not always compute the optimal schedule. That means that the constructed schedule matrix consists of a lot of idle or wasted timeslots. In other words, there are many idle time periods of the channels. This of course increases the length of the constructed schedule and leads to low channel utilization and, moreover, to low network performance. For the predictor system with which POSA carries, we can assert that it contributes in a great degree to the increase of the network throughput. At the same time, however, it lacks, as OIS, in channel utilization, since it does not introduce a mechanism that decreases the idle timeslots. Finally, CS-POSA leads to an increase of channel utilization, but the benefit is limited by the load of the network and fluctuates at low levels. This is due to the fact that CS-POSA intervenes before the construction of the schedule matrix, by shifting the order of the service of each node. This modification improves the performance of the network but does not affect the birth rate of idle timeslots. In this paper, we further extend the above work by presenting a novel schedule mechanism in order to reduce the number of idle timeslots of the final constructed matrix.

In this section, we present a new scheduling method that is based on the two previous algorithms that were discussed, namely OIS [30] and POSA [31]. This new protocol attempts a synthesis of the main features of OIS and POSA and results in a performance improvement, in terms of channel utilization and network throughput. WFF introduces a new set of schedule computation mechanisms called the cleanup mechanism and the refresh function. The innovation of WFF lies in the way it modifies the scheduling algorithm so that it minimizes (or even eliminates) the number of idle timeslots, thus significantly improving channel utilization, while maintaining a high network throughput. WFF incorporates the same prediction mechanism as POSA which, when combined with the pipelining of the schedule computation time and transmission time, explains the significant performance improvement that is observed.

A. Cleaning Mechanism

The innovative cleanup mechanism of WFF acts after the construction of the schedule. It is actually a procedure during which the timeslots that contain at least one idle channel (subtimeslot) are located and logically erased so that the total number of idle timeslots is minimized and the channel utilization is increased. The function of the cleaning mechanism can be divided into the following four steps [34].

1) Locate the timeslots that contain at least one idle channel subtimeslot (referred to as idle timeslots).
2) Logically erase these timeslots and construct the schedule matrix without these timeslots.
3) Reschedule the requests that were contained in the deleted timeslots.
4) At regular, predetermined intervals, perform the refresh function, and schedule all stored (in queues) packets to put an upper bound on the incurred service delay.

The discovery of idle timeslots is a pretty simple procedure. After constructing the schedule matrix according to OIS/POSA algorithm, the rows of the matrix (corresponding to channels) are scanned one by one for all columns (i.e., for all timeslots). When a slot containing at least one idle channel is located, it is logically erased, which means that the transmissions it contains are performed in one of the following frames. This means that the requests that were rescheduled are added to the new requests that the stations send for the following frame and that the actual data continue to be stored in queues while their transmission is being scheduled. Of course, if the cleanup mechanism finds an idle channel, w_i (row i) does not continue to the rest of the channels next to i but moves to the next timeslot (column).

B. Refresh Function

It is obvious that if the algorithm always functioned according to the cleanup mechanism, the channel utilization would be approximately equal to 100% since the number of idle slots would be almost zero. This ideal level of channel utilization and the corresponding high network throughput come at the cost of a significant delay, since the number of packets waiting in queues constantly increases, and their scheduling is left to be done later. For this reason, the cleaning mechanism includes a process called refresh function, during which the contents of the waiting queues are emptied (all waiting packets are forced to be scheduled for transmission). This cleanup process is performed at regular frame intervals (i.e., every time a constant number of frames have elapsed). For the frame that the refresh process is performed, WFF functions as OIS/POSA, and the specific frame is named as refresh frame. Fig. 6 shows a scheme that results a normal or a refresh frame. In this way, if we suppose that the refresh rate is equal to 4 and the current frame is 2000, then frame 2000 is the refresh frame, frame 2001 is the normal frame, frame 2002 is the normal frame, frame 2003 is the normal frame, frame 2004 is the refresh frame, and so on. Obviously, during a normal frame, the cleanup function is

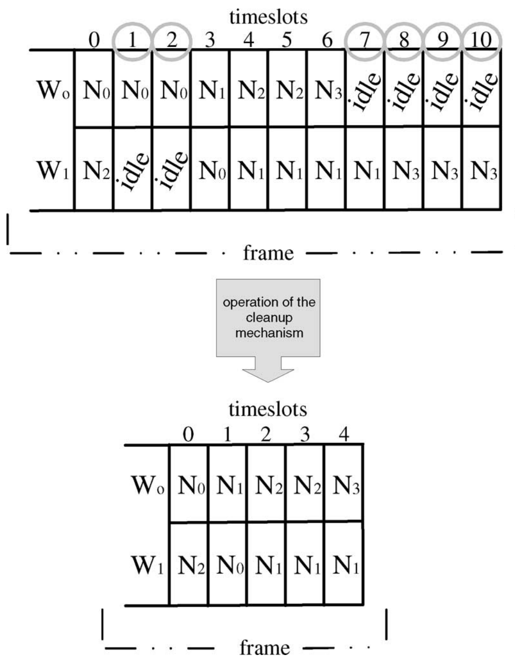

Fig. 7. Operation of the cleanup mechanism for the frame f.

operated, while during a refresh frame, the refresh function is executed. Finally, we have to pinpoint that the refresh rate is defined from the beginning and has the same value for all nodes.

C. Example Comparison

The function of the cleanup mechanism is shown with an example. Consider a network with four nodes $(N_0, N_1, N_2,$ and $N_3)$ that transmit data using two channels (W_0 and W_1). Assume that a total of 16 requests are submitted for frame f and that these requests are distributed, where D_f is the demand matrix for frame f (which is the same demand matrix D as in Section V).

$$D_f = \begin{bmatrix} 3 & \dots & 1 \\ 1 & \dots & 4 \\ 2 & \dots & 1 \\ 1 & \dots & 3 \end{bmatrix}$$

It is clear that node N_0 requests three timeslots, for channel W_0 and one timeslot for channel W_1. Node N_1 requests one timeslots for channel W_0 and four timeslots for channel W_1. Node N_2 requests two timeslots for channel W_0 and one timeslot for channel W_1. Finally, node N_3 requests one timeslot for channel W_0 and three timeslots for channel W_1. The function of OIS or POSA produces the schedule matrix, as plotted in Fig. 3. The CS-POSA algorithm constructs the schedule, which is observed in Fig. 5. At this point, the cleanup mechanism is activated in the constructed schedule of OIS/POSA (Fig. 3). The timeslots that include at least one idle channel are identified and marked in Fig. 7. These timeslots are 1, 2, 7, 8, 9, and 10. The second step of the cleanup mechanism is to exclude these timeslots (or equivalently the corresponding transmission requests) from the schedule matrix. Note that the excluded requests are not overlooked. The third step of the cleanup mechanism is the rescheduling of the excluded requests in the frame that follows (together with the nodes' actual requests for the next frame). Therefore, two timeslots for node N_0, on channel W_0 along with one timeslot for node N_1 and three timeslots for node N_3 on channel W_1, will be rescheduled. The matrix that contains the rescheduled requests for frame $f+1$, which is denoted as D'_f, is shown below:

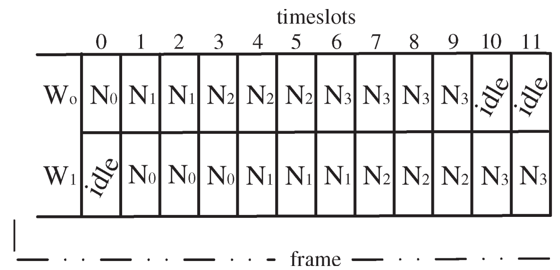

Fig. 8. Final schedule matrix of OIS/POSA for the frame $f+1$.

$$D'_f = \begin{bmatrix} 2 & \dots & 0 \\ 0 & \dots & 1 \\ 0 & \dots & 0 \\ 0 & \dots & 3 \end{bmatrix}.$$

Summarizing the schedules computed for frame f, it can be observed that the schedule length for OIS and POSA is equal to 11 timeslots and that the schedule length for CS-POSA is equal to 9 (as described in Section V), while the schedule length for WFF is merely five timeslots. The number of idle slots for OIS and POSA is equal to 6, and the number of idle slots for CS-POSA is equal to 2, while there are naturally no idle timeslots for WFF.

Assume that the demand matrix D_{f+1} for the next frame (i.e., frame $f+1$) is as in

$$D_{f+1} = \begin{bmatrix} 1 & \dots & 3 \\ 2 & \dots & 3 \\ 3 & \dots & 3 \\ 4 & \dots & 2 \end{bmatrix}$$

According to the cleanup mechanism, the demand matrix for frame $f+1$ would be the result of the addition of tables $D'_f + D_{f+1}$. Therefore, the new demand matrix for WFF for frame $f+1$ is equal to

$$D'_f + D_{f+1} = \begin{bmatrix} 2 & \dots & 0 \\ 0 & \dots & 1 \\ 0 & \dots & 0 \\ 0 & \dots & 3 \end{bmatrix} + \begin{bmatrix} 1 & \dots & 3 \\ 2 & \dots & 3 \\ 3 & \dots & 3 \\ 4 & \dots & 2 \end{bmatrix} = \begin{bmatrix} 3 & \dots & 3 \\ 2 & \dots & 4 \\ 3 & \dots & 3 \\ 4 & \dots & 5 \end{bmatrix}.$$

In the same manner, the OIS/POSA algorithm constructs the schedule matrix, which is shown in Fig. 8 for frame $f+1$. Of course, OIS/POSA and CS-POSA computes the schedule based only on D_{f+1}. The constructed schedule contains three idle timeslots and has frame length equal to 12. Similarly, CS-POSA algorithm constructs the schedule matrix, which is

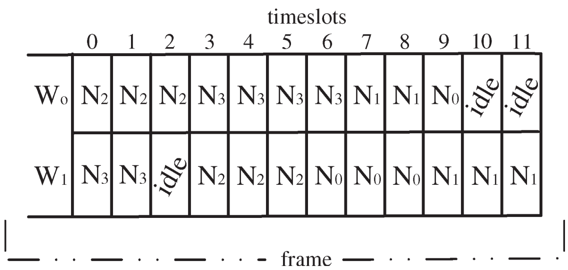

Fig. 9. Final schedule matrix of CS-POSA for the frame $f+1$.

plotted in Fig. 9. It is clear that CS-POSA shifts the order of the nodes before the construction of the schedule. Therefore, the requests of the node N_2 will be examined first, followed by those of N_3, then by those of N_1, and finally by those of N_0. The constructed schedule has frame length equal to 12 and includes three idle timeslots. WFF computes its schedule based on the total demands of $D'_f + D_{f+1}$. If we construct the schedule matrix, we observe that it contains a number of seven idle timeslots (Fig. 10). The rescheduled requests are contained in D'_{f+1}. Therefore, two timeslots, requested by node N_3, on channel W_0 and five timeslots, requested by the same node on channel W_1, will be rescheduled

$$D'_{f+1} = \begin{bmatrix} 0 & \dots & 0 \\ 0 & \dots & 0 \\ 0 & \dots & 0 \\ 2 & \dots & 5 \end{bmatrix}.$$

In conclusion, for frame $f+1$, OIS/POSA and CS-POSA constructed a schedule of 12 timeslots, while WFF constructed a schedule which lasts ten timeslots. There are no idle slots for WFF in contrast to OIS/POSA and CS-POSA, where the number of idle timeslots is three.

We complete our example by considering a last set of demands D_{f+2} for frame $f+2$.

$$D_{f+2} = \begin{bmatrix} 2 & \dots & 2 \\ 3 & \dots & 2 \\ 4 & \dots & 2 \\ 3 & \dots & 0 \end{bmatrix}$$

According to the cleanup mechanism, the demand matrix for frame $f+2$ is the result of the addition of tables $D'_{f+1} + D_{f+2}$. Therefore, the new demand matrix for WFF is equal to

$$D'_{f+1} + D_{f+2} = \begin{bmatrix} 0 & \dots & 0 \\ 0 & \dots & 0 \\ 0 & \dots & 0 \\ 2 & \dots & 5 \end{bmatrix} + \begin{bmatrix} 2 & \dots & 2 \\ 3 & \dots & 2 \\ 4 & \dots & 2 \\ 3 & \dots & 0 \end{bmatrix}$$
$$= \begin{bmatrix} 2 & \dots & 2 \\ 3 & \dots & 2 \\ 4 & \dots & 2 \\ 5 & \dots & 5 \end{bmatrix}.$$

The OIS/POSA algorithm constructs the schedule matrix, which is shown in Fig. 11. The constructed schedule contains six idle timeslots and has frame length equal to 12. Similarly, CS-POSA algorithm constructs the schedule matrix, which is shown in Fig. 12. It is clear that CS-POSA shifts the order of the nodes before the construction of the schedule. Therefore, the requests of the node N_2 will be examined first followed by those of N_1, then by those of N_0, and, finally, by those of N_3. The frame length of the constructed schedule is equal to 12 timeslots and contains six idle timeslots. If we assume that the refresh function is performed during frame $f+2$, we draw some important conclusions from this example. As we mentioned, the refresh function forces the WFF to operate like OIS/POSA, without the process of the cleanup mechanism. Fig. 13 shows the final schedule. The nodes request 55 timeslots in total from frame f to frame $f+2$ (16 for the first frame, 21 for the second frame and 18 for the third). In order to serve these requests, OIS and POSA utilize a total of 35 timeslots (11 in frame f, 12 in frame $f+1$, and 12 in frame $f+2$). CS-POSA utilizes a total of 33 timeslots (nine in frame f, 12 in frame $f+1$, and 12 in frame $f+2$), while WFF utilizes only 27 timeslots (five in frame f, 10 in frame $f+1$, and 14 in frame $f+2$). The total number of wasted timeslots for OIS/POSA is 15 (27%), for CS-POSA, it is 11 (20%), while WFF only wasted six (11%) timeslots (all in the last frame when cleaning reset was performed).

D. Predictor of WFF

Assuming that there are N nodes and W channels, each node possesses a set of W queues. The aim of the predictor is to construct the $N \times W$ demand matrix D with N rows (nodes) and W columns (channels). Each value in the table ranges between 0 and K. Thus, the overall predictor resembles NW separate and independent predictors, whose aim is to accurately predict the expected next number of slots that node n requests to transmit using channel w in the next frame.

The predictor makes the following assumptions concerning its operation [31].

1) The network traffic can be predicted using a traffic model.
2) The operation of each of the different NW predictors is independent and is not influenced by any other.
3) There is a constant upper bound K for the number of slots that each node can request on a specific channel.

There are four critical parameters that determine the effectiveness of the predictor of WFF, as mentioned in [31]. These are as follows.

1) Accuracy of prediction: It is a fact that a high level of prediction accuracy is desirable. In POSA and WFF, this is proven through empirical results.
2) Real time learning: The predictor should be capable of responding dynamically to changes in network activity.
3) Asymptotic time complexity: The WFF algorithm must not have a high level of complexity, such that would not facilitate its real time operation. POSA has an asymptotic time complexity less than $O(NW^2K)$. The same goal is also achieved by WFF.
4) Scalability: In order to be useful, the algorithm must be able to scale efficiently with the number of nodes and channels, as well as with the traffic intensity and the length of the scheduling queue. This goal is also achieved by both POSA and WFF.

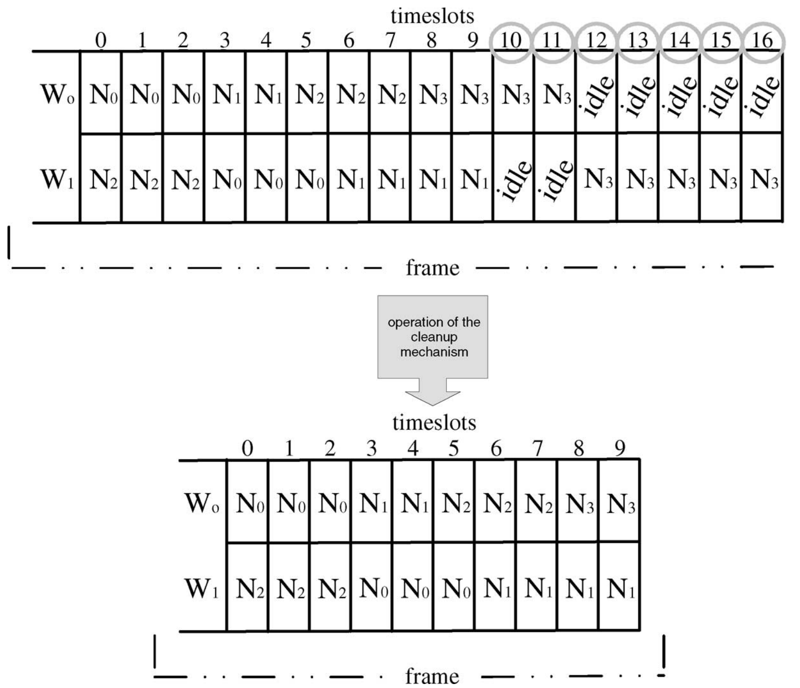

Fig. 10. Operation of the cleanup mechanism for the frame $f+1$.

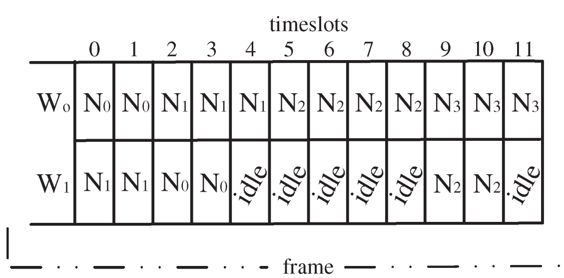

Fig. 11. Final schedule matrix of OIS/POSA for the frame $f+2$.

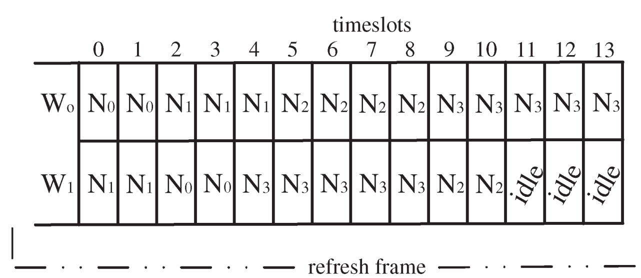

Fig. 13. Operation of the refresh function for the frame $f+2$.

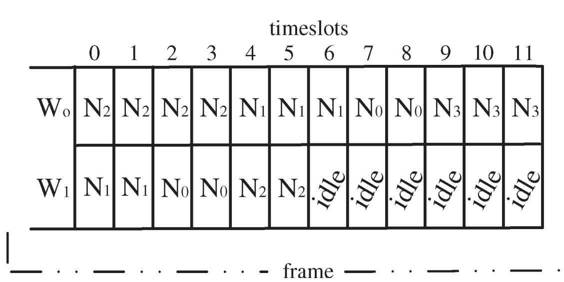

Fig. 12. Final schedule matrix of CS-POSA for the frame $f+2$.

E. Phases of WFF

The WFF protocol operates with a prediction mechanism. This mechanism should learn how to predict, so it is necessary the protocol to provide it training period. Additionally, between the two levels, i.e., between the level of the training and the level of the prediction, there is one more level: the level of change from the one phase to the other. In each level, the protocol acts with a different algorithm as far as the construction of the schedule matrix. In the training level, WFF operates as OIS. It accepts the requests of the nodes, it places them in the history queues, computes the schedule matrix with the help of the OIS algorithm based on the actual requests of the nodes, and then it starts to transmit following the reservations of the schedule matrix. It must be mentioned that during the training level the executing algorithm is OIS and not WFF, since there is not parallelism and prediction.

Conclusively, the training phase is defined by three independent operations:

1) adoption of the OIS/POSA algorithm without any intervening change;
2) collection of the actual requests of each node and construction of the schedule according to them;
3) filling the actual requests into the history queues in order to train the predictor the traffic pattern and improve its prediction accuracy.

In the level of change, WFF switches from the level of training into the level of prediction. We consider that the algorithm has learned the way the network works and constructs the schedule matrix not based on the actual requests of the nodes but based on the output of the prediction. Also, WFF puts the cleanup mechanism into action in order to construct the schedule matrix.

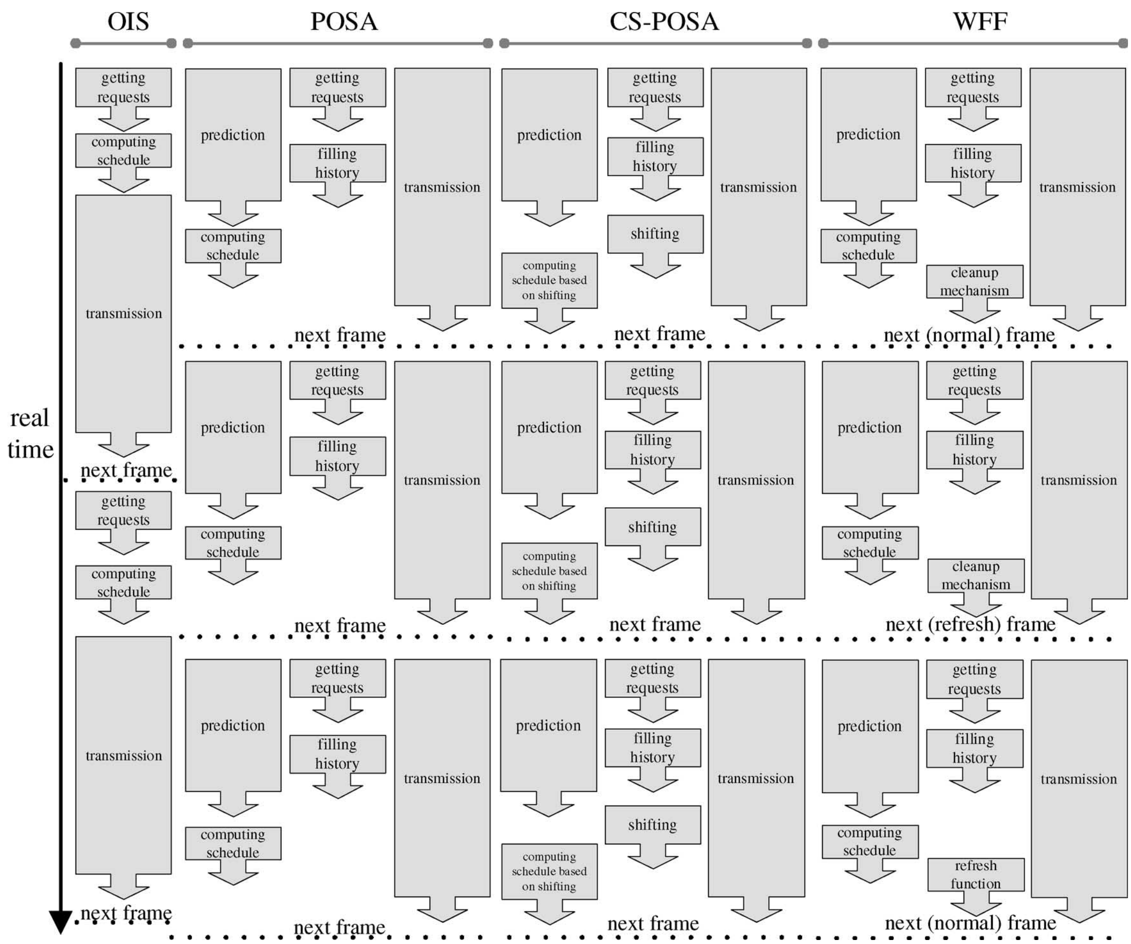

Fig. 14. Structural comparison among OIS, POSA, CS-POSA, and WFF.

Conclusively, the prediction phase consists of three independent phases:

1) adoption of the prediction mechanism from POSA without intervening changes;
2) construction of the schedule matrix with the help of OIS/POSA algorithm, followed by the cleanup mechanism;
3) using of the refresh function in regular time periods.

F. Structural Comparison Among OIS, POSA, CS-POSA, and WFF

It is very useful to see figuratively the structural comparison of the four algorithms that were analyzed before. In Fig. 14, there are all the independent levels that take place in the total execution time of each algorithm. Moving from the top to the bottom, there is the actual execution time. The discontinues lines separate, as far as time is concerned, the frame for each algorithm. The first algorithm under examination is OIS. We can see that in its structure there is not any pipelining. The operation order of the actions is as follows: acceptance of the requests, computation of the schedule, and transmission. Exactly next to it, there are POSA's steps. Here, there is a parallel execution of the POSA's procedures. In a few words, the procedure of the prediction takes place simultaneously with three more: the acceptance of the actual requests, the filling history, and the transmission of the packets. Along with the transmission, there is also the computation of the schedule for the following frame, based on the previous frame predicted set of requests. On the left of POSA, we can see the stages of CS-POSA. The differentiation between the steps of POSA and CS-POSA is obvious. While both of them use the parallelism that the prediction mechanism offers, CS-POSA introduces an extra stage: the procedure of shifting of the nodes under the criterion of the total load of each node. This procedure occurs in parallel with the transmission of the packets, a fact that does not increase the length of the frame length, as far as time is concerned, but it keeps it at the same level as POSA. Finally, on the right, we can observe the operations of WFF. Here, we have the new recommended stage of a cleanup mechanism that acts after the schedule has been constructed. Additionally, it is clear that in the third continued frame, there is a refresh frame, and the operation of the function mechanism has been substituted by the refresh function aiming at emptying the waiting queues.

G. WFF Algorithm Complexity

The algorithm complexity is a defining parameter for the implementation of the protocol. It is understandable that an algorithm, which is applied in very fast optical networks, should be able to cope with the requests of the media and should not add extra delays. Moreover, it should operate in such a way

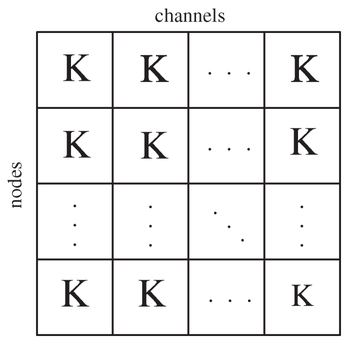

Fig. 15. Demand matrix D_{worst} with the maximum value of requests.

that it can be adjusted to any changes in the number of nodes or channels of the network, without significant variations in its performance. The algorithm complexity and the protocol scalability are crucial factors, which are analyzed below. WFF does not influence at all the prediction system. Therefore, the only operation that influences the algorithm complexity of WFF is the cleanup mechanism and, more specifically, the scanning of the idle timeslots of the final schedule matrix. This final matrix possesses as many rows as the channels of the network and as many columns as the total requested timeslots, in order that the transmission for each frame is completed. In other words, the algorithm must scan a number of columns equal to the frame length. Therefore, the problem is reformed into finding the frame length. The frame length is influenced by the amount of the load of the network. If the load is high, then the frame length is greater, and vice versa. Thus, the worst case for the frame length is when all of the nodes demand the maximum possible requests in timeslots, i.e., the following demand matrix D_{worst} is formed (Fig. 15). In this case, the frame length equals K demands multiplied by N nodes. Therefore, the final number of the cells in the matrix that should be checked is NWK, and the final algorithm complexity is $O(NWK)$.

It should be noted, however, that this asymptotic time for this operation is always less than $O(NWK)$, and in most cases, it is significantly less, as the cell being checked is less than NWK, due to the mean requests of each node being equal to $K/2$ and the algorithm stops scanning cells for the current timeslot if one channel is found idle. Regardless, if we consider that the complexity of the POSA algorithm, which is the same as OIS algorithm, is linear with the number of nodes and is given by $O(NW^2K)$, then we conclude that the total complexity of WFF is much less than OIS, and it does not add extra complexity.

VII. Performance Comparison

In this section, three algorithms—POSA, CS-POSA, and WFF—have been studied and compared in the context of channel utilization and network throughput, under uniform traffic. The system parameters varied are N, number of nodes; W, number of channels; and K, maximum value over all entries in the traffic matrix. Each entry in the matrix is a random number between zero and K (both inclusive). In order for the goal of scalability to be achieved, the value of K is not constant. The maximum number of slots that could be requested by a given node for a given channel during a individual frame K was given by the equation

$$K = \left\lceil \frac{NW}{5} \right\rceil.$$

The value of K is a function of the number of nodes and the number of channels, divided by an integer [31]. This allowed the scalability of the algorithm to be examined, as K changed. The target is to show that the algorithm is scalable not only in terms of N and W, but also in terms of K, as POSA. Furthermore, the relation between network throughput and mean packet delay is examined as the number of nodes and maximum load (the value of maximum K) vary. Also, it should be mentioned that the tuning latency time is considered to be equal to zero timeslots for simplicity reasons. The simulation took place in a C environment. For each experiment, a total of 10 000 frames were generated, 1000 of which belong to the training phase of the algorithms.

A. Performance Metrics

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

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

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

4) Throughput is symbolized by Γ and denotes the average number of bits transmitted per transmission frame per channel. It is measured in megabits per second. So

$$\Gamma = \frac{lR}{W(C + Ll/S)}$$

l denotes the packet length in bits, C the computation time in microseconds, and S the transmission rate in megabits per second. Since the three algorithms, which are examined, do not waste computation delay due to pipelining throughput, the relation finally becomes

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

5) 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, the data packet has been produced at the time moment t_1 and in the schedule matrix it has been set to be transmitted

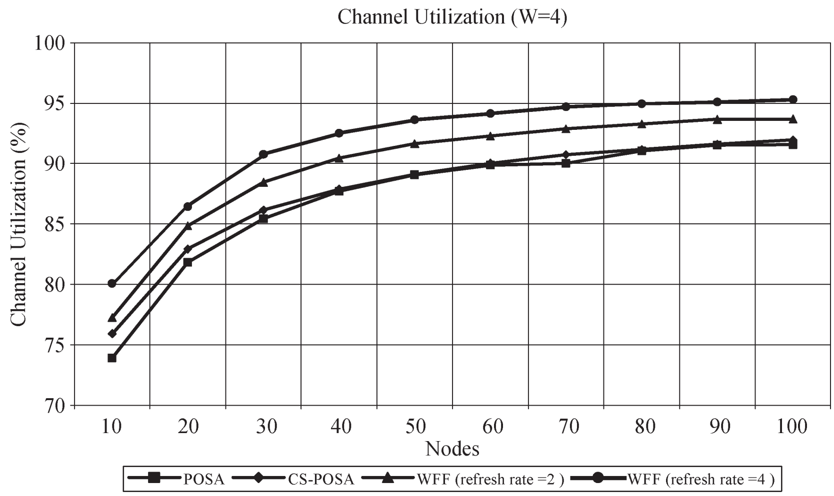

Fig. 16. Channel utilization with four channels.

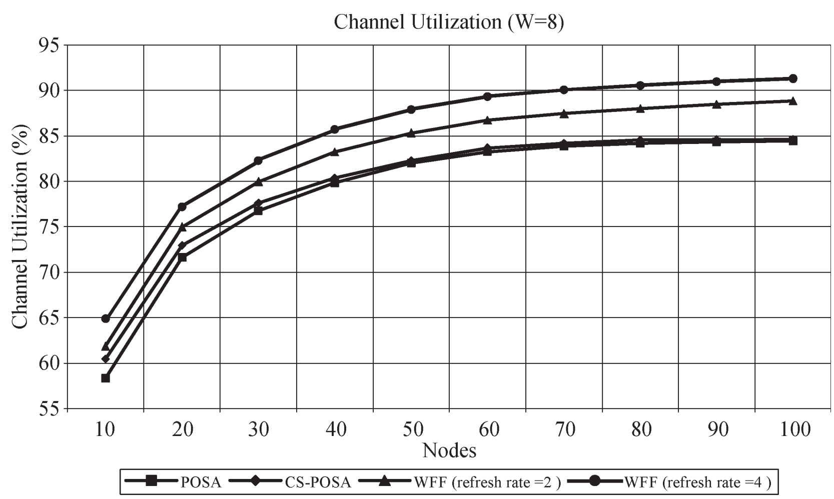

Fig. 17. Channel utilization with eight channels.

at the time moment t_2, where $t_2 - t_1 = t$ timeslots, then $D = t$.

6) The line speed (S) has been set in 2.4 Gb/s.

Two sets of simulated experiments were conducted on the simulation model for $N \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, \text{and } 100\}$, $W \in \{4, 8\}$, and $K = \text{FLOOR}(NW/5)$, in terms of channel utilization and network throughput. In order to study the relation between throughput and delay, another two sets of experiments were simulated. In the first case, the number of nodes increases $N \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, \text{and } 100\}$, $W \in \{4, 8\}$, and $K = \text{FLOOR}(NW/5)$, while in the second case, the impact of varying K is studied: $K \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, \text{and } 100\}$, $N = 30$, $W \in \{4, 8\}$. Finally, the relation between throughput and load is studied for $K \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, \text{and} 100\}$, $N = 30$, and $W \in \{4, 8\}$.

B. Scheduling Utilization Results

Channel utilization (or schedule utilization results) obtained by using POSA, CS-POSA, and WFF algorithm are plotted in Figs. 16 and 17. Fig. 16 shows the comparison using four channels and Fig. 17 shows the comparison using eight channels. It is obvious that WFF is more improved than POSA and CS-POSA, regardless of the refresh rate. Certainly, WFF with refresh rate equal to 4 obtains a significant difference over POSA and CS-POSA. The graphs plot the numerical view of the three algorithms from 10 to 100 nodes: $N \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, \text{and } 100\}$, $K = \text{FLOOR}(NW/5)$, and WFF remains better for four and eight channels. The biggest difference in scheduling utilization between POSA and WFF with refresh rate 2 appears to be on the level of 4.40% for 100 nodes and eight channels, while the biggest difference between POSA and WFF with refresh rate 4 seems to be on the level of 6.87% for 100 nodes and eight channels. Also, the biggest difference in scheduling utilization between CS-POSA and WFF with refresh rate 2 appears to be on the level of 4.25% for 100 nodes and eight channels, while the biggest difference between CS-POSA and WFF with refresh rate 4 seems to be on the level of 6.72% for 100 nodes and eight channels. In any case, it is useful to point out that the difference between WFF with refresh rate 2 and the other two algorithms does not fall below 1.33% for each channel and for each number of nodes. The most important conclusion from this comparison between the three algorithms when measuring the schedule utilization is that WFF remains constantly better than POSA and CS-POSA for each number of nodes and channels.

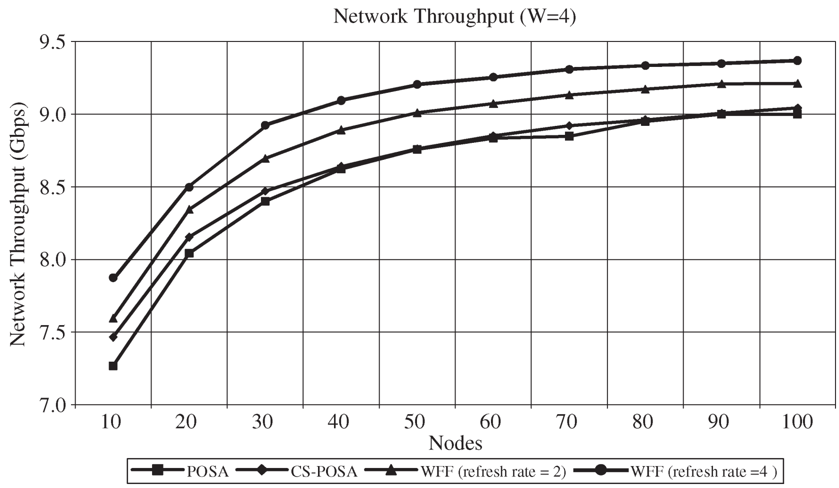

Fig. 18. Network throughput with four channels.

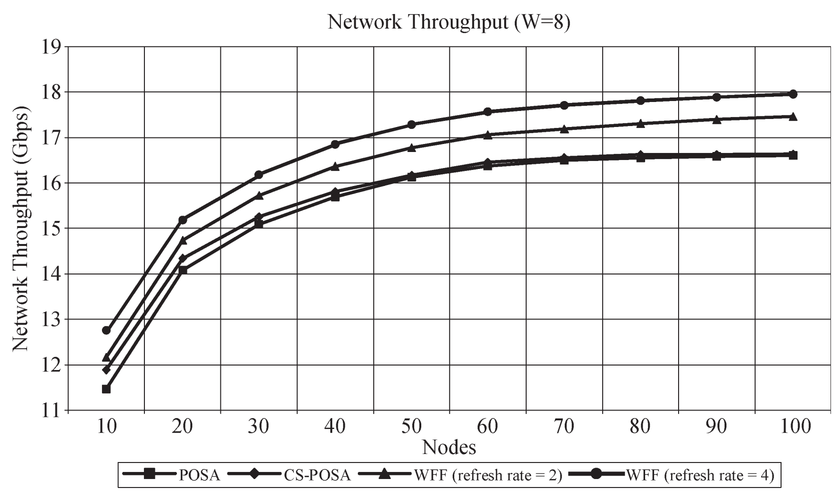

Fig. 19. Network throughput with eight channels.

C. Throughput Results

The results of the simulation, in terms of network throughput do not differ much from the results of the schedule utilization since the performance of the network throughput appears, as a direct result of the performance of the schedule utilization. Two sets of simulated experiments were conducted on the simulation model for $N \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, \text{and } 100\}$, $W \in \{4, 8\}$, and $K = \text{FLOOR}(NW/5)$, in terms of network throughput. The results from the comparison between the three algorithms are presented in the two following figures (Figs. 18 and 19) for four and eight channels. The difference is obvious, while the more the channels are increased, the more the difference is increased. It is worth mentioning that again WFF is constantly better that POSA and CS-POSA for any channel, regardless of the number of the nodes in the network. Also, it is clear for the WFF algorithm that the more the refresh rate is increased, the more the maximum throughput is obtained. The maximum difference between POSA and WFF for four channels reaches 328 Mb/s (refresh rate $= 2$) and 606 Mb/s (refresh rate $= 4$), while for eight channels, the maximum difference seems to be equal to 865 Mb/s (refresh rate $= 2$) and 1350 Mb/s (refresh rate $= 4$). The maximum difference between POSA and CS-POSA for eight channels reaches 251 Mb/s (refresh rate $=$ 2) and 453 Mb/s (refresh rate $= 4$), while for eight channels, the maximum difference is equal to 764 Mb/s (refresh rate $= 2$) and 1321 Mb/s (refresh rate $= 4$).

D. Throughput Versus Delay as the Number of Nodes Increases

The relation throughput-delay is very important since it influences the general performance of the network. Higher throughput means that more packets travel through the communication media. Higher delay means that the packets retain more time in the node's queues. The delay that we analyze refers to the mean time delay of the packets in the waiting queues until their transmission starts. Of course, this time is either minimal and is considered negligible by many studies, or it is not taken into consideration in the simulations. In the specific study, however, it is regarded as a very important factor, since the performance of the algorithm in the schedule operation is studied. This time is measured in timeslots, and the real value of each timeslot depends on the line speed of the medium. In this paragraph, the relation between throughput and delay is studied as the number of nodes is increased:

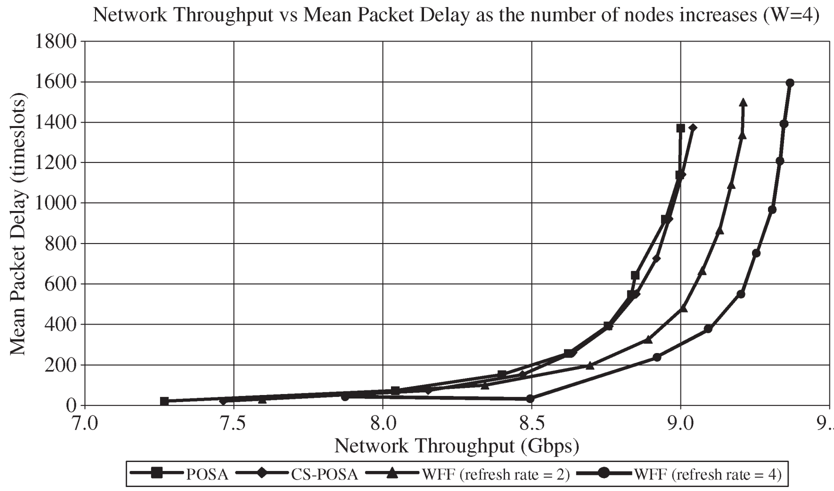

Fig. 20. Network throughput versus mean packet delay, as nodes are increased with four channels.

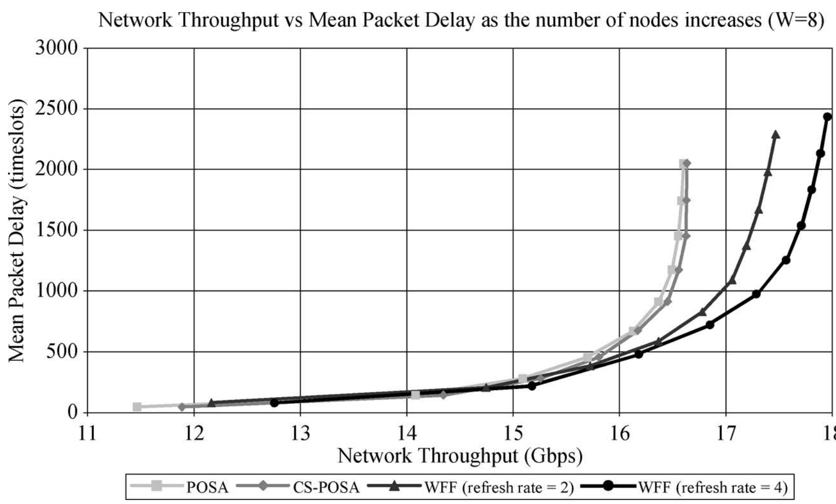

Fig. 21. Network throughput versus mean packet delay, as nodes are increased with eight channels.

$N \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, \text{and } 100\}$, $W \in \{4, 8\}$, and $K = \text{FLOOR}(NW/5)$. The results from the comparison between the three algorithms are presented in Fig. 20 for four channels and in Fig. 21 for eight channels. The behavior of the three algorithms is presented, i.e., the relation throughput-delay, altering the number of the nodes. It is clear that for each channel, the performance difference of the algorithms, as far as throughput is concerned, is greater, benefiting WFF. At the same time, however, WFF presents a slight increase in the mean packet delay. This increase seems to be higher as the refresh rate is higher. However, in each of the above cases, we cannot claim that this increase is disinclined for the operation of the algorithm, since it moves in low levels. This is the tradeoff of the algorithm, i.e., it gains in channel utilization and network throughput, while it suffers a little in mean packet delay.

E. Throughput Versus Delay as Maximum K Increases

We use a specific network model with $K \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, \text{and } 100\}$, $N = 30$, and $W \in \{4, 8\}$: The results from the comparison between the three algorithms are presented in Fig. 22 for four channels and in Fig. 23 for eight channels, by altering the values of the workload of the network, i.e., of K. In the graphs, it is obvious that there is a constant difference between the algorithms in the context of throughput as the mean packet delay increases. In other words, it can be observed that for each value of K, all the algorithms have almost the same mean packet delay, while WFF is seen as improved in the context of throughput. For example, for $K = 30$ and $W = 4$, WFF is better than POSA and than CS-POSA, in terms of throughput (8.7 Gb/s for WFF with rate $= 2$, 8.9 Gb/s for WFF with rate $= 4$, 8.3 Gb/s for POSA, and 8.4 Gb/s for CS-POSA) but suffers in terms of mean packet delay (235 timeslots for WFF with rate $= 2$, 283 timeslots for WFF with rate $= 4$, 195 timeslots for POSA, and 197 timeslots for CS-POSA). Also, for $K = 30$ and $W = 8$, WFF is again better than POSA and CS-POSA in network throughput (15.8 Gb/s for WFF with rate $= 2$, 16.3 Gb/s for WFF with rate $= 4$, 15.3 Gb/s for POSA, and 15.4 Gb/s for CS-POSA), while it keeps a little higher delay than POSA and CS-POSA (281 timeslots for WFF with rate $= 2$, 350 timeslots for WFF with rate $= 4$, 201 timeslots for POSA, and 202 timeslots for CS-POSA).

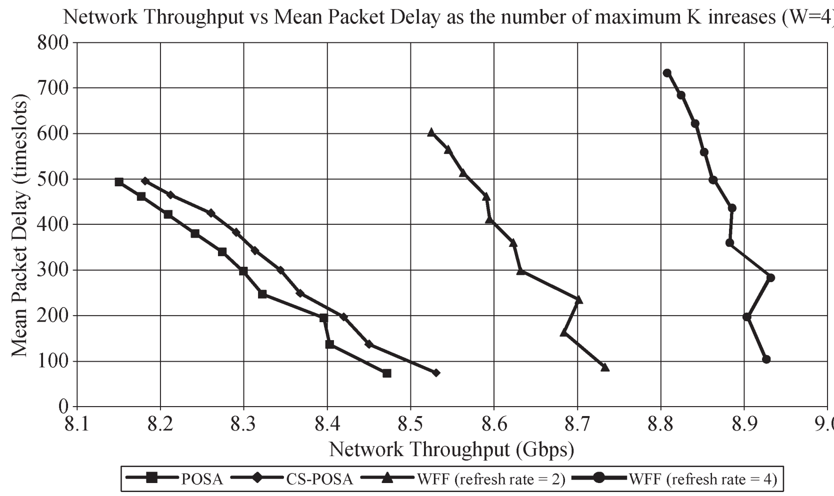

Fig. 22. Network throughput versus mean packet delay, as the maximum K increases with four channels.

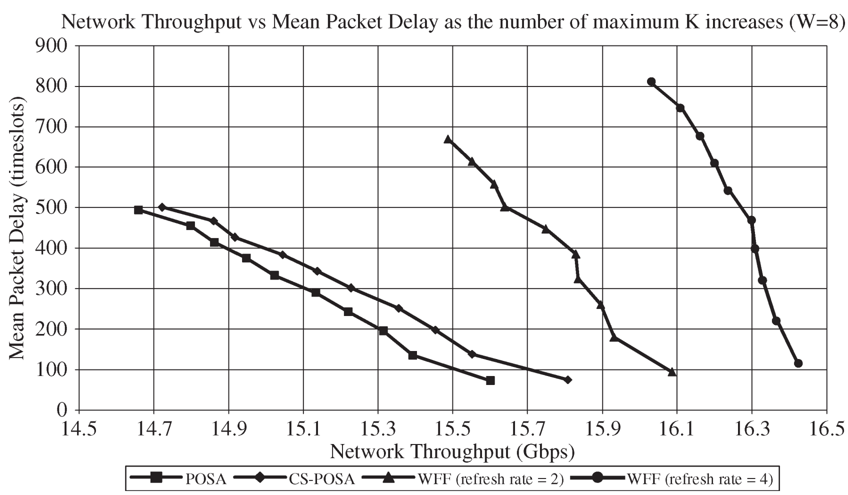

Fig. 23. Network throughput versus mean packet delay, as the maximum K increases with eight channels.

For each value of K and W, WFF improves the network throughput, while the improvement is better as the channels increase but suffers a few percent in terms of mean packet delay, due to the fact that some of the packets wait a certain amount of time for rescheduling. Therefore, it can be concluded that WFF does not lack significantly in delay, which means that the improvement that it brings to the network in the context of throughput is stable.

F. Throughput Versus Load Results

The results from the comparison of the three algorithms are shown in Fig. 24 for four channels and in Fig. 25 for eight channels. Fig. 24 presents the behavior of the algorithms, i.e., the relation between throughput and load, changing the values of the workload of the network, i.e., of K. More specifically, the simulation parameters are $K \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, \text{and} 100\}$, $N = 30$, and $W \in \{4, 8\}$. It must be mentioned that while the workload of the network is increased, the throughput is decreased for all algorithms. For example, for WFF with refresh rate 4 and $W = 4$, when K equals to 10, the throughput equals to 8.92 Gb/s. When K equals to 50, the throughput equals approximately 8.88 Gb/s. Finally, when K equals 100, the throughput is decreased reaching 8.80 Gb/s. This phenomenon is not often met in the category of the networks examined. Nevertheless, it appears that all the algorithms examined—OIS [30], POSA [31], CS-POSA [32], [33]—are all 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 schedule matrix. If there was an open space of nine slots in the constructed schedule matrix and the packet that arrived was of ten timeslots size duration of transmission, then the algorithm could not 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 ten timeslots. This leads to a decrease of the channel utilization as the unused timeslots are increased and the throughput is decreased. Finally, we can observe that the decrease of the WFF algorithm is less

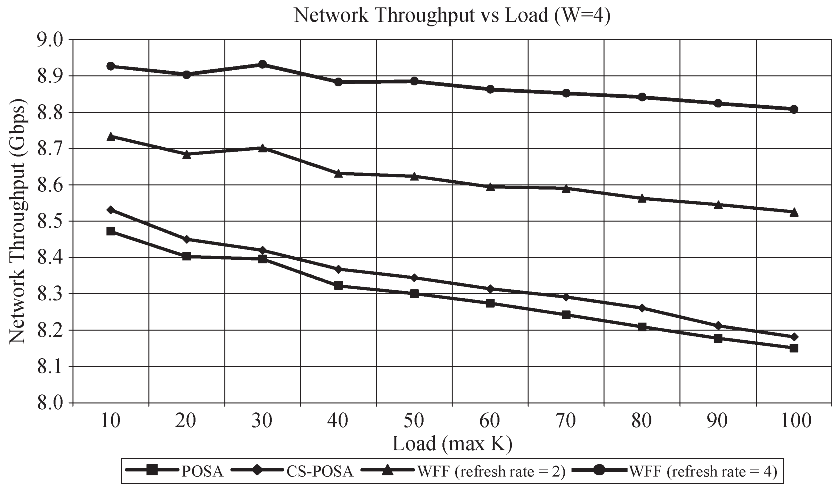

Fig. 24. Network throughput versus load with four channels.

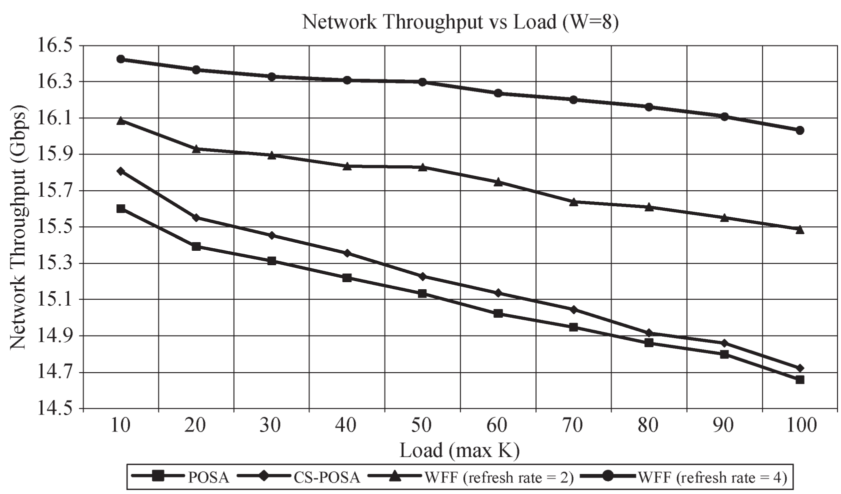

Fig. 25. Network throughput versus load with eight channels.

than the other protocols. For example, observing Fig. 25, it can be seen that the difference between the maximum and the minimum value of throughput is equal to 390 Mb/s for WFF (refresh rate 4), 600 Mb/s for WFF (refresh rate 2), 941 Mb/s for POSA, and 1018 Mb/s for CS-POSA. Therefore, WFF seems to be more stable, as network traffic varies. Also, WFF seems to be more stable, as refresh rate is increased.

VIII. Conclusion

This paper describes a predictive online collision free coordination-based MAC protocol for a star broadcast and select WDM optical network. Its architecture is TT–FR and supports direct access to all nodes via one hop. The proposed algorithm maintains a prediction system that allows the network the pipelining of the schedule computation phase with the reservation and the data transfer phases. Also, it maintains a very simple algorithm to construct the final schedule matrix, which has time complexity linear to the number of the nodes. The novel mechanism (the cleanup mechanism) tries to scan and find idle timeslots and deletes them logically. This way, the schedule length is less than other online protocols, and the number of idle timeslots is decreased. Our results show that the proposed algorithm is superior, in terms of utilization and throughput, but suffers a little in mean packet delay. Also, the suggested algorithm keeps complexity at low levels and at the same time supports higher network efficiency.

References

[1] G. I. Papadimitriou, C. Papazoglou, and A. S. Pomportsis, "Optical switching, 'Switch fabrics, techniques, and architectures'," *J. Lightw. Technol.*, vol. 21, no. 2, pp. 384–405, Feb. 2003.

[2] S. Yao, B. Mukherjee, and S. Dixit, "Advances in photonic packet switching: An overview," *IEEE Commun. Mag.*, vol. 38, no. 2, pp. 84–94, Feb. 2000.

[3] P. E. Green, *Fiber Optic Networks*. Eaglewood Cliffs, NJ: Prentice-Hall, 1993.

[4] A. S. Acampora and M. J. Karol, "An overview of lightwave packet networks," *IEEE Network Mag.*, vol. 3, no. 1, pp. 29–41, Jan. 1989.

[5] B. Mukherjee, "WDM optical networks: Progress and challenges," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 10, pp. 1810–1824, Oct. 2000.

[6] C. S. R. Murthy and M. Gurusamy, *WDM optical networks: Concepts, design, and algorithms*. Englewood Cliffs, NJ: Prentice-Hall PTR, 2002.

[7] C. A. Brackett, "Dense wavelength division multiplexing network: Principles and applications," *IEEE J. Sel. Areas Commun.*, vol. 8, no. 6, pp. 948–964, Aug. 1990.

[8] T. E. Stern and K. Bala, *Multiwavelength Optical Networks: A Layered Approach*. Reading, MA: Addison-Wesley, May 1999.

[9] G. I. Papadimitriou, P. A. Tsimoulas, M. S. Obaidat, and A. S. Pomportsis, *Multiwavelength Optical LANs*. New York: Wiley, 2003.

[10] G. I. Papadimitriou, A. N. Miliou, and A. S. Pomportsis, "OCON: An optically controlled optical network," *Comput. Commun.*, vol. 22, no. 9, pp. 811–824, Jun. 1999.
[11] P. Green, "Progress in optical networking," *IEEE Commun. Mag.*, vol. 39, no. 1, pp. 54–61, Jan. 2001.
[12] S. Ramaswami and K. N. Sivarajan, *Optical Networks: A Practical Perspective*, 2nd ed. San Mateo, CA: Morgan Kaufmann, 2002.
[13] B. Mukherjee, *Optical Communication Networks*. Reading, MA: Addison-Wesley, 1997.
[14] G. I. Papadimitriou, "Centralized packet filtering protocols: A new family of MAC protocols for WDM star networks," *Comput. Commun.*, vol. 22, no. 1, pp. 11–19, Jan. 1999.
[15] K. Bogineni and P. W. Dowd, "A collisionless multiple access protocol for a wavelength division multiplexed star-coupled configuration: Architecture and performance analysis," *J. Lightw. Technol.*, vol. 10, no. 11, pp. 1688–1699, Nov. 1992.
[16] A. Ganz and Z. Koren, "WDM passive star—Protocols and performance analysis," in *Proc. IEEE INFOCOM*, Bal Harbour, FL, Apr. 7–11, 1991, pp. 991–1000.
[17] K. M. Sivalingam, K. Bogineli, and P. W. Dowd, "Pre-allocation media access control protocols for multiple access WDM photonic network," in *Proc. ACM SIGCOMM*, Aug. 1992, pp. 235–246.
[18] ——, "Acknowledgement techniques of random access based media access protocols for a WDM photonic environment," *Elsevier Comput. Commun.*, vol. 16, no. 8, pp. 458–471, Aug. 1993.
[19] K. M. Sivalingam and P. W. Dowd, "A multi-level WDM access protocol for an optically interconnected multiprocessor system," *J. Lightw. Technol.*, vol. 13, no. 11, pp. 2152–2167, Nov. 1995.
[20] K. M. Sivalingam and J. Wang, "Media access protocols for WDM networks with on-line scheduling," *J. Lightw. Technol.*, vol. 14, no. 6, pp. 1278–1286, Jun. 1996.
[21] K. M. Sivalingam, J. Wang, X. Wu, and M. Mishra, "Improved on-line scheduling algorithms for optical WDM networks," in *Proc. DIMACS Workshop Multichannel Opt. Netw.*, New Brunswick, NJ, Mar. 1998, pp. 43–61.
[22] A. Ganz, "End-to-end protocols for WDM star networks," presented at the Proc. Workshop Protocols High-Speed Netw., pp. 219–235, Zurich, Switzerland, May 1989, IFIP.WG6.1–WG6.4.
[23] J. Diao and P. L. Chu, "Packet rescheduling in WDM star networks with real-time service differentiation," *J. Lightw. Technol.*, vol. 19, no. 12, pp. 1818–1828, Dec. 2001.
[24] E. Modiano, "Random algorithms for scheduling multicast traffic in WDM broadcast-and-select networks," *IEEE Trans. Netw.*, vol. 7, no. 3, pp. 425–434, Jun. 1993.
[25] E. Modiano and R. Barry, "A novel medium access control protocol for WDM based LANs and access networks using a master/slave scheduler," *J. Lightw. Technol.*, vol. 18, no. 4, pp. 461–468, Apr. 2000.
[26] Y. Ito, Y. Urano, T. Muratani, and M. Yamaguchi, "Analysis of a switch matrix for an SS/TDMA system," *Proc. IEEE*, vol. 65, no. 3, pp. 411–419, Mar. 1997.
[27] M. S. Borella and B. Mukherjee, "Efficient scheduling of nonuniform packet traffic in a WDM/TDM local lightwave network with arbitrary transceiver tuning latencies," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 5, pp. 923–934, Jun. 1996.
[28] M. Azizoglou, R. A. Barry, and A. Mikhtar, "Impact of tuning delay on the performance of bandwidth-limited optical broadcast networks with uniform traffic," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 5, pp. 935–944, Jun. 1996.
[29] P. S. Henry, "Very high capacity lightwave networks," in *Proc. IEEE ICC*, 1988, pp. 1206–1209.
[30] K. M. Sivalingam, J. Wang, J. Wu, and M. Mishra, "An interval-based scheduling algorithm for optical WDM star networks," *J. Photon. Netw. Commun.*, vol. 4, no. 1, pp. 73–87, 2002.
[31] E. Johnson, M. Mishra, and K. M. Sivalingam, "Scheduling in optical WDM networks using hidden Markov chain based traffic prediction," *J. Photon. Netw. Commun.*, vol. 3, no. 3, pp. 271–286, Jul. 2001.
[32] P. G. Sarigiannidis, G. I. Papadimitriou, and A. S. Pomportsis, "A new prediction and channel sorting based scheduling algorithm for WDM star networks," in *Proc. 12th Annu. Symp. IEEE/CVT*, Enschede, The Netherlands, Nov. 2005.
[33] ——, "CS-POSA A high performance scheduling algorithm for WDM star networks," *Photon. Netw. Commun.*, vol. 11, no. 2, pp. 209–225, Mar. 2006.
[34] ——, "WFF: A high performance scheduling algorithm for WDM star networks that minimizes idle timeslots," in *Proc. IEEE SCVT*. Enschede, The Netherlands, Nov. 2005.

Panagiotis G. Sarigiannidis (S'06) received the B.S. degree in computer science from the Department of Informatics of Aristotle University, Thessaloniki, Greece, in 2001, where he is currently working toward the Ph.D. degree in optical networks.

His research interests include optical networks and optical switching.

Georgios I. Papadimitriou (M'89–SM'02) received the Diploma and Ph.D. degrees in computer engineering, both from University of Patras, Patras, Greece, in 1989 and 1994, respectively.

From 1989 to 1994, he was a Teaching Assistant with Department of Computer Engineering of the University of Patras and a Research Scientist with Computer Technology Institute, Patras. From 1994 to 1996, he was a Postdoctorate Research Associate with Computer Technology Institute. From 1997 to 2001, he was a Lecturer with Department of Informatics, Aristotle University, Thessaloniki, Greece. Since 2001, he has been an Assistant Professor with Department of Informatics, Aristotle University. He is the author of more than 120 refereed journal and conference papers. He is coauthor of the books *Multiwavelength Optical LANs* (Wiley, 2003) and *Wireless Networks* (Wiley, 2003), and coeditor of the book *Applied System Simulation* (Kluwer, 2003). His research interests include optical networks, wireless networks, high-speed local area networks, and learning automata.

Dr. Papadimitriou is Associate Editor of five IEEE journals, including the IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS—PART C, the IEEE TRANSACTIONS ON BROADCASTING, the IEEE SENSORS JOURNAL, the IEEE NETWORK and the IEEE COMMUNICATIONS MAGAZINE.

Andreas S. Pomportsis received the B.S. degree in physics and the M.S. degree in electronics and communications, both from the University of Thessaloniki, Thessaloniki, Greece, the Diploma degree in electrical engineering from Technical University of Thessaloniki, and the Ph.D. degree in computer science from University of Thessaloniki, in 1987.

Currently, he is a Professor with Department of Informatics, Aristotle University of Thessaloniki. He is coauthor of the books *Multiwavelength Optical LANs* (Wiley, 2003) and *Wireless Networks* (Wiley, 2003). He has published over 150 journal and conference papers. His research interests include computer networks, learning automata, computer architecture, parallel and distributed computer systems, and multimedia systems.