ELSEVIER

Computer Networks 31 (1999) 2139–2152

COMPUTER NETWORKS

www.elsevier.com/locate/comnet

An efficient message scheduling algorithm for WDM lightwave networks

Maode Ma [a], Babak Hamidzadeh [b], Mounir Hamdi [a,*]

[a] *Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China*
[b] *Department of Electrical and Computer Engineering, University of British Columbia, Vancouver BC, Canada*

Accepted 3 May 1999

Abstract

Two important issues that need to be addressed when designing medium access control (MAC) protocols for Wavelength Division Multiplexing networks are message sequencing and channel assignment. Channel assignment addresses the problem of choosing an appropriate data channel via which a message is transmitted. This problem has been addressed extensively in the literature. On the other hand, message sequencing, which addresses the order in which messages are sent, has rarely been addressed. In this paper, we propose a new reservation-based message scheduling algorithm called RO-EATS that addresses both the channel assignment and message sequencing during its scheduling process. We formulate an analytical model and conduct extensive simulations to evaluate the performance of this algorithm. We compare the performance results of a well-known algorithm which only addresses the channel assignment issue with those of our new algorithm. The comparison shows that our new algorithm gives significant improvement over scheduling algorithms that do not consider message sequencing. As a result, we anticipate that these research results will lead to new approaches to message scheduling on WDM networks.

Keywords: Optical network; Wavelength Division Multiplexing (WDM); Medium access control protocol; Scheduling algorithm

1. Introduction

With the proliferation of the World Wide Web (WWW) in all aspects of networking, current local and wide area networks can barely cope with the huge demand for network bandwidth. As a result, there is a world-wide effort in upgrading current networks with high-bandwidth fiber-optic links that can potentially deliver Tera-bits/s. *Wavelength Division Multiplexing* (WDM) is an effective technique for utilizing the large bandwidth of an optical fiber. By allowing multiple messages to be simultaneously transmitted on a number of channels, WDM has the potential to significantly improve the performance of optical networks. The nodes in such a network can transmit and receive messages on any of the available channels by using and tuning one or more tunable transmitter(s) and/or tunable receiver(s). Several topologies have been proposed for WDM networks [1,2]. Of particular interest to us in this paper is the single-hop topology where a WDM optical network is configured as a broadcast-and-select network in which all the inputs from the

* Corresponding author. Tel: +852 2358 6984; Fax: +852 2358 1477; E-mail: hamdi@cs.ust.hk

1389-1286/99/$ - see front matter
PII: S1389-1286(99)00085-7

various nodes are combined in a passive star coupler, and the mixed optical information is broadcast to all destinations [3].

To unleash the potential of single-hop WDM passive star networks, efficient medium access control (MAC) protocols are needed to efficiently allocate and coordinate the system resources [1]. MAC protocols in a single-hop WDM passive star network environment can be divided into two main classes, namely pre-allocation-based protocols and reservation-based protocols. Pre-allocation-based techniques use all channels of a fiber to transmit messages. These techniques assign transmission rights to different nodes in a static and pre-determined manner. Examples of preallocation-based protocols can be found in [3–6]. On the other hand, reservation-based techniques allocate a channel as the control channel to transmit global information regarding messages to all nodes in the network. Once such information is received, all nodes invoke the same scheduling algorithm to determine when to transmit/receive a message and on which data channel. Examples of reservation-based protocols can be found in [7–11]. Reservation-based techniques have a more dynamic nature and assign transmission rights based on the run-time requirements of the nodes. In this paper, we focus our attention on reservation-based techniques.

Two important issues that need to be addressed when designing (MAC) protocols for WDM networks are message sequencing and channel assignment. Channel assignment addresses the problem of choosing an appropriate data channel via which a message is transmitted. This problem has been addressed extensively in the literature. On the other hand, message sequencing, which addresses the order in which messages are sent, has rarely been addressed. In this paper, we propose a new algorithm that handles both the channel assignment and message sequencing issues pertaining to scheduling. Most of the existing reservation-based approaches schedule messages individually and independently of one another. They ignore that the way to choose the order of the message transmission may affect the performance of the network. To the best of our knowledge, only our previous paper [12] has addressed the issue of sequencing messages in WDM networks. Using that protocol, the order of transmission is determined by the message length. This algorithm has been shown to significantly improve the performance of a WDM network. However, if the difference of the message lengths are small, the algorithm will not be effective. Also it fails when messages are blocked due to avoiding receiver collisions.

In this paper, we propose a reservation-based protocol for scheduling variable-length messages in single-hop WDM passive star networks that overcomes the above deficiencies. The proposed technique addresses channel assignment and ordering message transmission. Our technique is more globally optimizing than existing approaches, since it not only shares global information among receiving and transmitting nodes, but it simultaneously considers multiple messages from different transmitting nodes. The scheduling algorithm is invoked when all nodes in the network have received the message control information. This approach not only provides more information, but it also reduces the number of times the scheduling algorithm need be invoked. This reduction results in lower scheduling overheads and permits more time for transmitter and receiver tuning.

Our algorithm is composed of two phases. The first phase decides the messages transmission order. The second phase is channel assignment. For the first phase, we use the global information on the receivers to impose a priority on the transmission ordering. This, as will be shown, reduces the average delay in the network. The second part of our algorithm is based on the *Earliest Available Time Scheduling* (EATS) algorithm, which has received a lot of attention as an effective algorithm for channel assignment in WDM networks [9]. We call our new algorithm *Receiver Oriented-Earliest Available Time Scheduling* (RO-EATS).

We formulate an analytical model and conduct extensive simulations for evaluating the performance of the RO-EATS algorithm. We then compare the performance results of the EATS algorithm with the performance results of our new algorithm. The comparison shows that our new algorithm significantly improves the performance of the EATS algorithm.

The difference between the EATS and the RO-EATS algorithms is that EATS only addresses channel assignment; whereas RO-EATS addresses both channel assignment and message sequencing. Mes-

sage sequencing in RO-EATS algorithm fully employs the information of the states of the destination nodes, which gives a significant improvement. Furthermore, the analytical model of RO-EATS is another main contribution of this paper.

The remainder of this paper is organized as follows. Section 2 specifies our system model and the problem to be addressed. Section 3 presents our new scheduling algorithm and the techniques involved. Section 4 provides our analytical performance model of RO-EATS. Section 5 presents the performance results from simulation experiments and theoretical analysis of RO-EATS compared to EATS. Finally, Section 6 concludes the paper with a summary of the results and a discussion of future work.

2. WDM system model

In this paper, we consider message transmission in a single-hop WDW optical network whose nodes are connected to a passive star coupler via two-way fibers. Each direction of the fiber supports $C+1$ WDM channels with the same capacity and there are N nodes in the network. The C channels, referred to as data channels, are used for message transmission. The remaining channel, referred to as the control channel, is used to exchange global information among nodes about the messages to be sent. The control channel is the basic mechanism for implementing the reservation scheme. Each node in the network has two transmitters and two receivers. One transmitter and one receiver are fixed and are tuned to the control channel. The other transmitter and receiver are tunable to any data channel.

The nodes are assumed to generate messages with variable lengths which can be divided into several equal-sized packets. The basic time interval on the data channels is the transmission time of one packet. The nodes are divided into two non-disjoint sets of source (transmitting) nodes s_i and destination (receiving) nodes d_j. A queue for buffering messages to be transmitted is assumed to exist at each source node s_i.

A Time Division Multiple Access (TDMA) protocol is used on the control channel to avoid collision of the control packets belonging to different nodes. According to this protocol, each node can transmit a control packet during a predetermined time slot. The basic time interval on the control channel is the

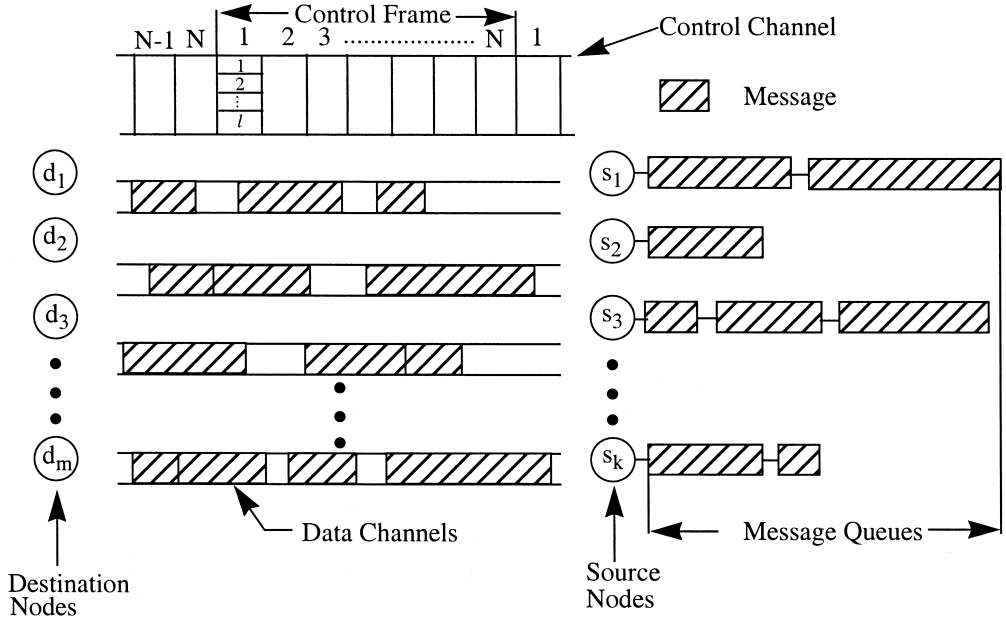


Fig. 1. Data and control channel configuration, and message queues at transmitting nodes.

transmission time of a control packet. N control packets make up one control frame on the control channel. Thus, each node has a corresponding control packet in a control frame, during which that node can access the control channel. The length of a control packet is a system design parameter and depends on the number of messages l about which each node is allowed to broadcast, and the amount of control information on each message (e.g. the address of the destination node, message length). Fig. 1 illustrates the basic concepts of the logical model of our system.

Fig. 1 is a logical model of the WDM optical network specified above. This logical model presents the mechanism of the system structure which only affects the protocol and scheduling algorithm design. In Fig. 1, we use s_i to show the set of source nodes and d_j to show the set of destination nodes. At each source node, there is a queue for messages awaiting transmission. Messages in each queue generally have 3 attributes: the source node; the message length; and the destination node. Each queue can be of any length. The special case is that each queue has only one message at its head. There are $C + 1$ channels in Fig. 1. One of them is the control channel, which is used to transmit control frames. The control frame contains the information of the messages at each node. One packet in the control frame represents the information of the messages at one node. If l messages can be scheduled at the same time, the packet will contain l messages' information at that node. The other C channels are the data transmission channels, which can be used by any source node. In our system, we use a star coupler to connect the nodes and channels, which is not shown in this figure as we focus on the protocol and scheduling algorithm.

3. Scheduling algorithm

Message transmission and reception in this system model works as follows: A node must transmit a control packet on the control channel in its assigned time slot before sending a message to its destination node. After one round-trip propagation delay, the destination node and the other nodes in the network receive the control packet. Then the distributed scheduling algorithm is invoked by each node to determine the data channel and transmission time slots for each message in the control frame. Once a message is scheduled, the sending node's transmitter will tune to the scheduled data channel and send the message at the scheduled time. The receiver of the message destination node should tune to that channel to be ready to receive the message. After a propagation delay, the message will arrive at the destination node.

We form our new *receiver oriented* scheduling algorithm based on the basic channel assignment algorithm, EATS [9]. Our proposed receiver oriented algorithm RO-EATS first considers the earliest available receiver among all the nodes in the network and then selects a message which is destined to this receiver from those which are ready and identified by the control frame. After that, a channel is selected and assigned to the selected message by the EATS algorithm. This new algorithm makes full use of the global information on each message and the system states of the network to improve the performance of the WDM network.

The basic idea of the EATS algorithm is to assign a message to a data channel that has the earliest available time slots among all the channels in the network. Once the data channel is assigned, the message is scheduled as soon as that channel becomes available. In order to keep a record of the channel and receiver usage and their states, two tables are used and reside on each node, which are denoted as *Receiver Available Time* array, RAT, and *Channel Available Time* array, CAT. RAT records the non-available times of the receiver on each node from the current time in the packet slot unit. CAT records the non-available times of each channel from the current time. Both of them are dynamically decreasing with time units. With this global information on each node, the distributed EATS works as follows: Transmit a control packet on the control channel; Choose a channel with the earliest available time; Calculate the transmission time of a message based on two tables; and Update the two tables according to newly scheduled message.

Our scheme for choosing a suitable message is based on the states of the receivers according to RAT. The objective of this scheme is to avoid lots of

messages going to one or a few nodes at the same time, and to raise the channel utilization. In our network architecture, channels and receivers can be thought as two kinds of resources in series, and exclusively occupied by messages. The CAT and RAT record the states of these resources, and represent global information available to all the nodes in the network. Two consecutive messages with the same destination node may not fully use the available channels when the EATS algorithm is employed. This is because when there are two consecutive messages going to same destination node, the first message will occupy one of the channels and the receiver at the destination will tune to that channel; while the second message has to wait until the first message has been successfully transmitted and received. During the transmission of the first message, the second message cannot be transmitted because the receiver is waiting for the first message, although there may be other available channels that can transmit the second message. In this situation, the second message will be blocked and some channels, which are not occupied, cannot be used. As a result, the average message delay, which is a metric of the system performance, will be degraded and the utilization of the transmission channels reduced. Our new algorithm prevents scheduling two consecutive messages to the same destination node. This algorithm always checks the table of RAT to see which node is the least used as a destination and chooses the message to transmit which is destined to this node. This message choosing scheme has the ability to sequence messages presented in the control frame according to the states of the receivers.

The complexity of RO-EATS algorithm can be evaluated based on its operations. It has two sequence procedures. One of them is to sort the RAT table, and the other is to sort the CAT table. The number of items of RAT is the number of nodes in the network. The number of items of CAT is the number of channels in the network. Let us assume that the number of nodes is always larger than the number of channels. So we consider only the number of nodes when we estimate the complexity of the algorithm. The complexity of a typical sorting algorithm is $O(n\log_2 n)$, where n can be mapped to the number of the network nodes in our case. The worst case running time of the algorithm is two times the sequence procedure and the complexity of the algorithm is still $O(n\log_2 n)$. The bandwidth of the transmission link is defined as the number of bits transmitted per unit time. In our network, we assume that one packet can be transmitted in one time slot. So the network bandwidth can be expressed as a function of the number of packets transmitted. The scheduling algorithm will not cost in terms of bandwidth as long as the scheduling procedure can be completed while messages in one control frame are being transmitted. The transmission time of all messages of one control frame can be approximated by $m * n$, where n is the number of nodes in the network, and m is the mean message length. The condition that our scheduling algorithm will not produce cost in terms of bandwidth can be formulated as $2 * n\log_2 n \le m * n$. This clearly shows that the mean message length can be a system design parameter. When it is large enough, our scheduling algorithm will not introduce any cost to the message transmission in the network.

The RO-EATS algorithm can be expressed in detail as follows:

We assume that there are M nodes and C channels. The messages have variable lengths that follow an exponential distribution. The messages can be transmitted from source node i to destination node j, where $i \neq j$, and $i, j \in M$. The *Receiver Available Time* (RAT) Table can be expressed as an array of M elements, one for each node. $RAT[j] = n$, where $j = 1, 2, \ldots, M$, means that node i will be free after n time slots. The *Channel Available Time* (CAT) Table can be expressed as an array of C elements, one for each channel. $CAT[k] = m$, where $k = 1, 2, \ldots, C$, means that channel k will be available after m time slots.

3.1. The RO-EATS algorithm

Begin
  Transmit a control packet on the control channel;
  Wait until the control packet returns;
message choosing scheme:
  Sort $RAT[j]$ in non-decreasing order by the value of $RAT[j]$ to form a new array $RAT'[l]$;
  Check whether the current message is destined

to node j, where $RAT'[l] = RAT[j]$ and $l = 0$;
If no, jump to **n1**; if yes, jump to channel assignment scheme;

n1:

The current message waits until next time to be checked;
Check other messages waiting in the frame whether one of them destined to node j, where $RAT'[l] = RAT[j]$ and $l = 0$;
If no, jump to **n2**; if yes, jump to channel assignment scheme;

n2:

$l = l + 1$;
Check the messages waiting in the frame whether one of them destinated to node j, where $RAT'[l] = RAT[j]$;
If no, jump to **n2**; if yes, jump to channel assignment scheme;

channel assignment scheme:

Sort $CAT[k]$ in non-decreasing order by the value of $CAT[k]$ to form a new array $CAT'[h]$;
Use the channel k to transmit the selected message, where $CAT'[h] = CAT[k]$ and $h = 0$;
Calculate $r = RAT[j] + T$, $t_1 = \max(CAT[k], T)$, $t_2 = \max(t_1 + R, r)$; where T is the transmitters' tuning time, R is the propagation delay.
Schedule the message transmission time at $t = t_2 - R$;
Update $RAT[j] = t_2 + m$, $CAT[k] = t_2 - R + m$, where m is the message length.

End.

4. A numerical example

In this section, we discuss the details of the proposed scheduling techniques in the context of an example. Fig. 2 presents a simplified system model of Fig. 1. Fig. 2 shows a network of 4 nodes and a set of 10 messages to be transmitted at the source nodes s_i through 3 channels C_i. In this figure, the boxes represent messages. We label each message in the queue by m_i. Each message has basically three attributes. The first one is the source node of a message. If a message is attached to a node, the name of the node is considered as the messages's first attribute. For example, in Fig. 2, message $m1$ is attached to node $n1$. The $m1$s source node is then $n1$. The second attribute is the length of the message, which is represented by the first number of the data pair above the message in the figure. And the second number of the pair is the third attribute of a message, which is the destination receiver of this message.

We start our discussion by observing the behavior of the EATS algorithm on this example. As mentioned earlier, this algorithm is a basic channel assignment algorithm and does not sequence messages in any particular order. EATS starts by assigning the message represented in the first control packet of a frame to the data channel with the earliest available time. It then proceeds to assign the message represented by the second control packet of a frame to the next channel with the earliest available time, and so on.

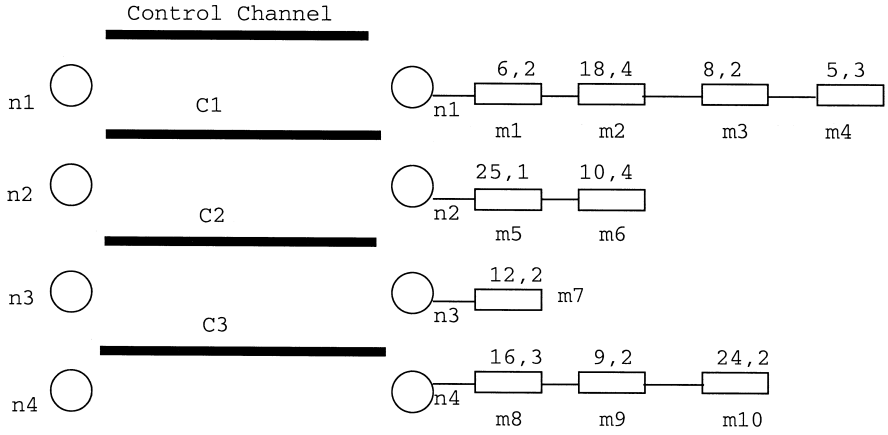


Fig. 2. Message queues at transmitting nodes.

Fig. 3 shows each of the control frames and their packets which transmit global information about the messages. The numbers on top of the control packets of each frame designate the corresponding source nodes. The numbers inside the packets indicate the message lengths. Each control packet in a frame transmits information about the message at the head of the queue at its corresponding source node. So the frame shown in Fig. 3a contains control information about messages $m1$, $m5$, $m7$ and $m8$ in control packets 1, 2, 3 and 4, respectively.

In this example, we assume that the data channels are initially idle. EATS can schedule each message after its corresponding control packet reaches all nodes, or it can schedule all messages represented by a frame after the entire frame has been transmitted by all nodes. The effect of these alternatives is semantically the same and leads to the same schedules. As Fig. 4 shows, EATS initially provides the schedule $\{(m1, C1), (m5, C2), (m7, C3), (m8, C1)\}$ which assigns message $m1$ to data channel $C1$, message $m5$ to data channel $C2$, message $m7$ to data channel $C3$, and message $m8$ to data channel $C1$. Although $m7$ is assigned to channel $C3$ it cannot be transmitted at the time slot 1. It has to wait until $m1$ has been transmitted because $m7$ has the same receiver as $m1$. Message $m7$ will be started at time slot 7. Message $m2$ of frame 2 is then assigned to channel $C3$ which has the earliest available time among the three channels at which time $m2$ can be scheduled.

Similarly, $m6$ is assigned to the channel with the earliest available time, namely $C1$. But it cannot be transmitted at the time slot 23 because it has the same receiver as message $m2$, which has been scheduled on channel $C3$ at the same time. It has to wait until message $m2$ has successfully been transmitted. According to the length of message $m2$, $m6$ has to wait 14 time slots. Message $m9$ is then assigned to $C2$, since this channel has the earliest available time at the time of assignment. Message $m3$ in frame 3 is the next message to be assigned to channels. At the current time, $C2$ is the earliest available channel. Message $m3$ is then assigned to it. In frame 3, there is only one message which has not been scheduled. That is $m10$. The earliest available channel at this time is $C3$. So $m10$ is assigned to this channel. Message $m10$ cannot be transmitted immediately after it is assigned to $C3$ just because $m10$ has the same destination as $m3$, which has already been scheduled on $C2$ before $m10$ was scheduled. As a result, $m10$ has to wait 6 time slots until the transmission of $m3$ has been completed. The last message is $m4$ in frame 4. It can be assigned to $C2$ as it is the earliest available channel at that moment. The result of this channel assignment leads to the schedule shown in Fig. 4.

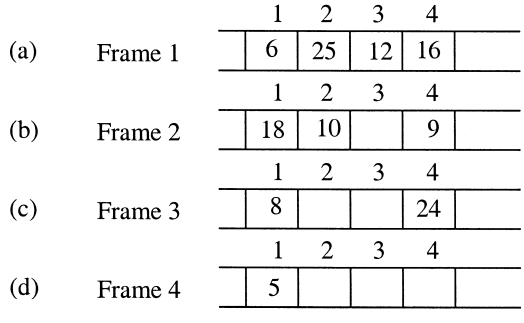


Fig. 3. Control frames using EATS and the corresponding frame scheduling.

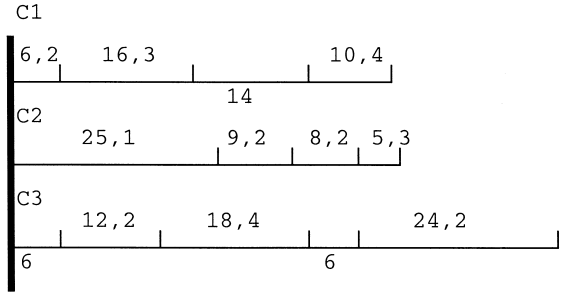


Fig. 4. Message scheduling using EATS.

In Fig. 4, each horizontal axis represents the time of each channel. Each data pair above one segment of the axis represents that segment of time on the channel that has been assigned to the corresponding message already. Each number below one segment of the axis indicates that the same number of time slots on the channel is not occupied by any message. The average delay of the 10 messages of this example using EATS can be calculated as: $(6 + 22 + 46 + 25 + 34 + 42 + 47 + 18 + 36 + 66) / 10 = 342 / 10 = 34.2$.

Our proposed RO-EATS algorithm uses the same control information as EATS, as shown in Fig. 3.

This protocol invokes the distributed scheduler after an entire control frame has been transmitted to all nodes. After receiving a control frame, the scheduler first considers selecting a message suitable to be transmitted according to the algorithm presented in the previous section to avoid a large number of messages destined to one or a few receivers at the same time in order to raise the channel utilization. After the selection of the message to be transmitted is determined, then the message is assigned to a channel by the simple channel assignment algorithm of EATS. The messages in a frame will be selected and scheduled one by one until all messages in the frame have been scheduled.

In our example, when the first control frame is received by every node, the distributed scheduler first sorts the *RAT* to look for the earliest available receiver and then picks up a message from the messages in the frame, which is destined to this node, to be scheduled and then assign a channel to this message. In this way, from the first frame, $m1$ is assigned to $C1$, $m5$ is to $C2$. When message $m7$ is considered to be scheduled, as it is destined to node $n2$, and $RAT[2]$ is larger than $RAT[3]$, $m7$ will remain unscheduled while $m8$ is scheduled first to channel $C3$. At last in this frame, $m7$ is assigned to the channel $C1$ as the channel $C1$ is the earliest available channel and there is no message destined to $n4$ although $RAT[4]$ is the smallest value. In the second frame, $m2$ is first considered and assigned to $C3$ because the value of $RAT[4]$ is the smallest and also is the value of $CAT[3]$ at the current time. Message $m6$ cannot be considered before $m9$ is considered because, currently, $RAT[2]$ is smaller than $RAT[4]$. Message $m9$ is considered to be assigned to $C1$. At last in this frame, $m6$ is allocated to $C2$ but can only be transmitted after time slot 34 because the transmission of $m2$, which has the same destination as $m6$, is scheduled to be finished at that time.

Similarly, in frame 3, $m3$ and $m10$ are destined to same node $n2$ and there are no other messages destined to other nodes. Node $n2$ is the only choice to be selected. Message $m3$ is the first to be considered for scheduling. Based on the EATS principle, $m3$ is assigned to $C1$ as it has the earliest availability. Then $m10$ is considered only based on the channel states. It is assigned to $C3$. But it has to wait one time slot for finishing the transmission of $m3$, which is destined to $n2$ in $C1$. Finally, in frame 4, it has only one message $m4$. The earliest channel available time is the only consideration to assign $C2$ to $m4$. The result of scheduling all messages in the 4 frames with our proposed RO-EATS algorithm is shown in Fig. 5. The average delay of the 10 messages by the RO-EATS algorithm can be calculated as: $(6 + 18 + 27 + 35 + 40 + 25 + 44 + 16 + 34 + 59)/10 = 304/10 = 30.4$.

It is evident from the result of this example that the RO-EATS algorithm improves the average delay compared to the EATS algorithm. This is attributed to RO-EATS's ability to address both message sequencing and channel assignment simultaneously. It is also shown in the scheduling schemes of Figs. 4 and 5, that the finishing times of the two algorithms are different. It is easily seen that the finish time of the 10 messages transmission by the RO-EATS algorithm is much smaller than that of the EATS algorithm. The improvement by our new algorithm is more than 10% compared to the EATS algorithm. This fact shows that when the number of messages to be transmitted is fixed the time used to transmit these messages by our new algorithm is shorter than that of EATS algorithm. In other words, when the time used to transmit the messages is fixed the number of messages to be transmitted by our new algorithm is more than that of the EATS algorithm. This implies that the throughput, which is one important system performance metric, of a WDM network using the RO-EATS algorithm can be expected to be better than that of a WDM network using the EATS algorithm. Our algorithm not only decreases the average message delay but also gives better load balance on the channels.

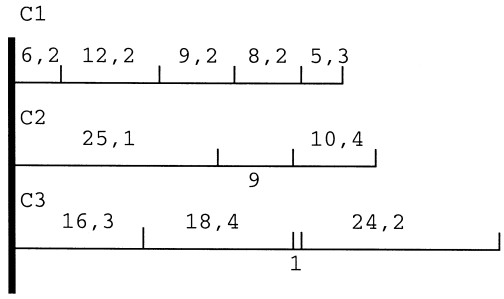


Fig. 5. Message scheduling using the RO-EATS algorithm.

5. Analytical model

To compare the performance of the scheduling algorithms, we give an approximate mathematical model for a WDM network using the RO-EATS algorithm. The goal of this model is to study the performance of a WDM network using the new algorithm under the condition of a limited number of data channels in the network. In a practical WDM network, the number of channels is less than the number of nodes. The performance metric we use is the average message delay in the network.

In order to make the model show the main characteristics of the system, several assumptions are used and can be summarized as follows:

1. Both transmitters/receivers tuning time T, and the propagation delay of messages, R are set to be constant.
2. The message population is finite as we consider only the messages in one control frame, which is the number of nodes in the network, N.
3. For each of the nodes, the message arrival is a Poisson process and follows its own independent message arrival distribution with the same mean value of λ.
4. The message transmitted by a node can be destined to every other node with equal probability.
5. For each of the nodes, the message length is exponentially distributed with same mean value of $1/\mu$.
6. For each of the nodes, the probability that each node has one message is approximately $1/N$.

A WDM optical network using the RO-EATS algorithm can be modeled as a M/G/1 with a priority queueing system. The priority assignment scheme is based on the states of the destination nodes. The population of this queue is bound by the number of nodes as we consider the queue being composed of the messages in one control frame. The server of the queue can be considered as the set of data channels with different service rates. The service rate of a channel depends on the message length it serves. The messages come into each node to form the queue with the same mean arrival message rate λ and probability $1/N$. The mean message arrival rate of the whole queue can be approximated by $(N-k) \times \lambda/N$, where k represents the system state, which means there are k messages in the queue and $k \in \{0,1,\ldots, N-1\}$. The mean service rate of the whole queue is state dependent. It can be expressed by:

$$\mu_k = \alpha\mu' k \quad \text{when } k \le C, \tag{1}$$

$$\mu_k = \alpha\mu' C \quad \text{when } k > C, \tag{2}$$

where μ' can be expressed as:

$$\mu' = \frac{1}{T + R + 1/\mu}. \tag{3}$$

μ_k is the service rate of the whole queue when the system is in state k; and μ is the reciprocal of the mean value of message length; μ' is the reciprocal of the sum of mean transmission time, receivers/transmitters tuning time and propagation delay. C is the number of data channels in the network; α is the probability that any one of the channels is not used because the receiver is unavailable. It is shown that the service rate of the whole queue depends on the number of messages staying in the queue. In this formula, we set α, the probability that any one of the channels is not used because the receiver is unavailable, to be 1. This means that the utilization of the channels will not be affected by the receivers unavailability. This consideration is based on the advantages of the RO-EATS scheduling algorithm. The RO-EATS algorithm intentionally avoids one or a few receivers being used too much and distributes k messages to as many different receivers as possible to ensure all the channels can be fully used. This algorithm can actually minimize both the number of messages blocked by receiver unavailability and number of channels unemployed.

The system traffic intensity or the traffic load P_k, which is the ratio of the messages arrival rate to the service rate of the queueing system, when kth priority messages are served, can be expressed by the following relationship:

$$P_k = \frac{\lambda_k}{\mu_k} = \frac{(N-k) \times \lambda/N}{\alpha\mu' k}. \tag{4}$$

Applying Little's result to our M/G/1 priority queuing system, we can obtain the relationships between the average message delay D_k of the kth priority message, and the average waiting time DW_k of the kth priority message in the queue. Finally, we

can derive the average message delay time D of all messages in the system. In particular, the waiting time DW_k of the kth priority messages can be expressed as follows:

$$DW_k = \frac{\sum_{i=1}^{k} p_i/\mu_i}{2 \times \left(1 - \sum_{i=1}^{k} p_i\right) \times \left(1 - \sum_{i=1}^{k-1} p_i\right)}. \quad (5)$$

The delay time D_k of the kth priority messages can be obtained by adding the message's service time $1/\mu_k$, to the waiting time DW_k of the kth priority message in the queue:

$$D_k = \frac{1}{\mu_k} + DW_k. \quad (6)$$

Based on the above formulae, we can calculate the average delay time of all messages in the system as follows:

$$D = \frac{1}{N} \times \sum_{k=1}^{N} D_k. \quad (7)$$

6. Simulation and analytical results

In this section, we first present the results of the above analytical model of a system using EATS and RO-EATS. We also present the design and simulation experiments of a WDM network using the different algorithms. We compare these results to verify the accuracy of the analytical model.

Fig. 6 shows the results of the analytical models of the system using RO-EATS and EATS. The mathematical equations that calculate the average message delay of the network using RO-EATS is presented in Section 5; while the mathematical model of the system using EATS appears in [9]. We consider the case when the average message arrival rate at each node λ is varying. The average delay of the messages in the network is studied under a limited number of channels and a given number of nodes. The parameters of the network are set as follows. The receiver and transmitter tuning times are set to 0. The propagation delay of the transmission link is

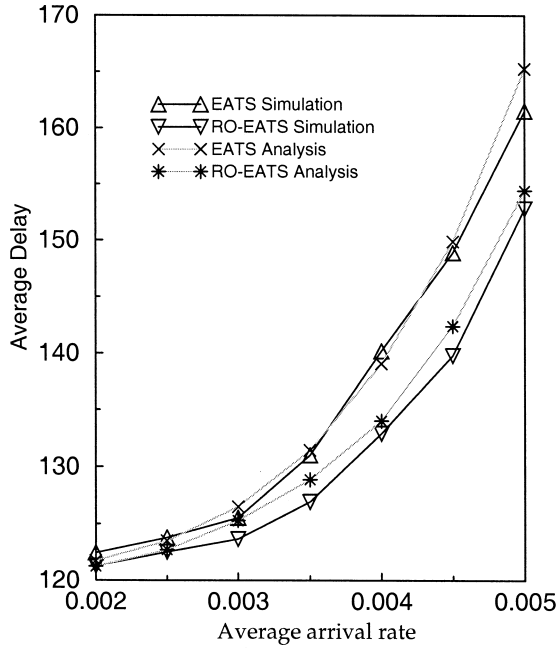


Fig. 6. Simulation and analytical results of the two algorithms.

set to 100. The number of nodes is fixed at 50. The number of channels is 4. The average message length is 20 time slots. The message length follows an exponential distribution. The message arrival is a Poisson process. When the mean message arrival rate for each node varies from 0.002 to 0.005, the average message delay of the system using EATS ranges from 121 to 165 time slots; while that for RO-EATS ranges from 122 to 155 time slots. After the critical point of the average arrival rate, namely at 0.0045, the difference between the average delay of the two algorithms reaches up to more than 16% of the average delay caused by queueing, tuning, and transmission. Hence, it can be seen that RO-EATS has improved the system performance a lot over that of EATS.

We conduct a series of simulation experiments in three groups by using a discrete-event-simulator. The behavior of the candidate algorithms is observed over a simulation period of 10,000 time units. Each point in the performance graphs is the average of 10 independent runs.

From Figs. 6–8, we show the results of the first simulation experiments, which studies the effect of

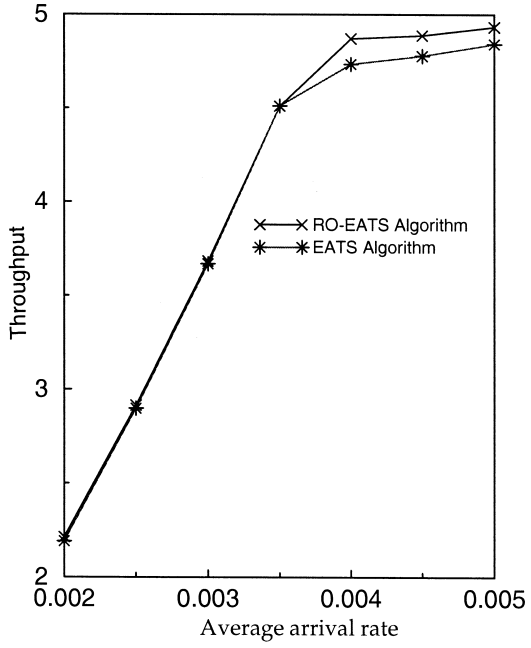


Fig. 7. Throughtput versus average arrival rate.

the mean message arrival rate on the average message delay and system throughput. In Fig. 6, we use exactly the same values of the parameters for the simulation to validate them. We present the results of the simulation experiments of a WDM network using RO-EATS and using EATS. When the mean message arrival rate of each node varies from 0.002 to 0.005, the average delay of the messages using the EATS algorithm ranges from 123 to more than 160 time slots; while that for the RO-EATS algorithm ranges from 122 to slightly more than 150 time slots. The difference between the average delay of the two algorithms reaches up to nearly 20% of the average delay caused by queueing, tuning, and transmission at the critical point. The simulation results confirms the fact that RO-EATS can improve the system performance of the network quite a bit.

Fig. 6 also shows that the results of the mathematical models are quite close to the results of simulation experiments with the same system parameters.

We have conducted other simulation experiments to show the improvements on other performance metrics by RO-EATS. One of the results is to show another performance metric – the throughput as a function of the mean messages arrival rate. The parameters we use are the same as those of the above experiments. The results of this simulation experiment are shown in Fig. 7.

In Fig. 7, the system throughputs of the two algorithms are shown as a function of packets per unit time slot. We can see that the throughput of the WDM network using RO-EATS is larger than the throughput of the WDM network using EATS, especially when the average message arrival rate reaches the point of 0.004. This implies that the system using RO-EATS has better capacity than the system using EATS to balance the work-load. This shows that RO-EATS not only improves the average message delay but also improves the WDM channels throughput.

We combine the simulation results shown in Figs. 6 and 7 and put them into Fig. 8 to show the relationship between the average message delay and the system throughput. The parameters we use here are the same as the parameters used in Figs. 6 and 7. From this figure, we can see that the performance of RO-EATS is better than EATS in the sense that at a certain value of system throughput, the average message delay of the system using RO-EATS is always

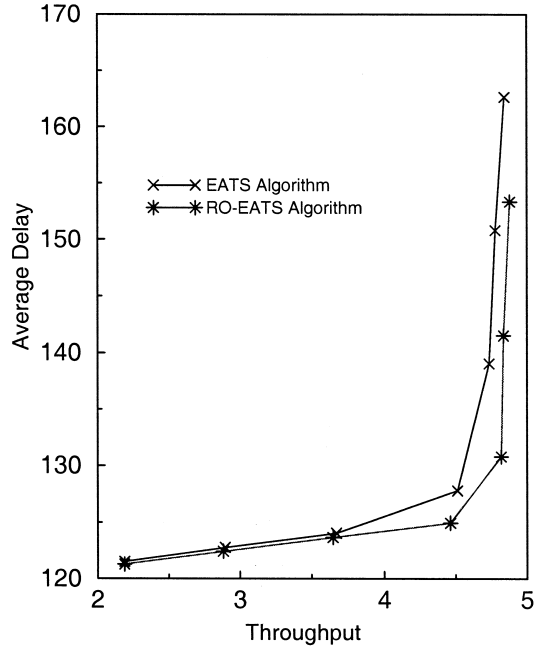


Fig. 8. Average message delay versus throughput.

less than that of the system using EATS. This is shown clearly when the throughput approaches 4.5.

The second group of simulation experiments we did is to show the effect of the number of channels on the average message delay and system throughput. The results are shown in Figs. 9 and 10. The parameters we use in the simulation are as follows: The number of transmitter nodes is 50. The average message length is 20 time slots. The mean message arrival rate of each node is fixed at 0.0045. The number of channels is varied from 4 to 10. Fig. 9 shows that with increasing number of channels, the average message delay of the two algorithms decreases. When the number of channels is high, the average message delays of the two algorithms are kept quite low. However, RO-EATS outperforms EATS when the number of channels is low.

Fig. 10 shows the relationship between the system throughput and the number of channels available in the system. In Fig. 10, we can see that the system using RO-EATS always has larger throughput than the system using EATS for any number of channels.

The third group of simulation experiments is to show how the average message delay and system throughput are affected when the average message length is varied. The results of these experiments are

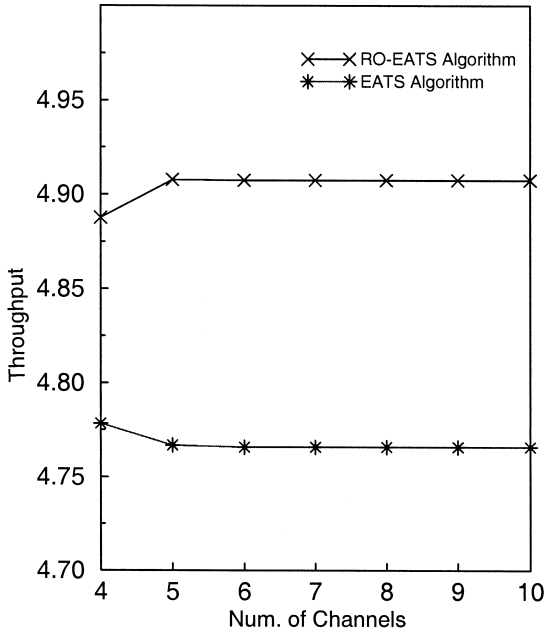


Fig. 10. System throughput versus number of channels.

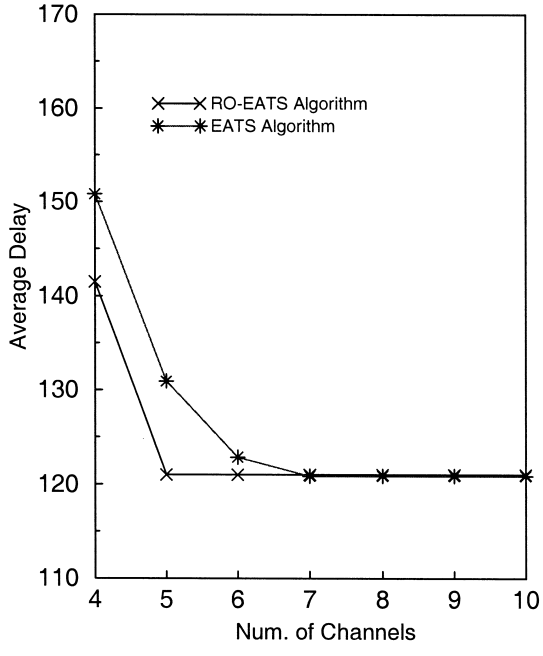


Fig. 9. Average message delay versus number of channels.

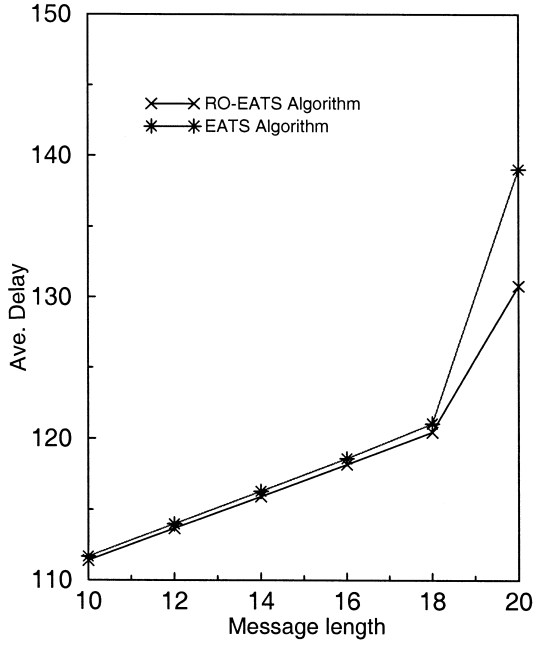


Fig. 11. Average message delay versus average message length.

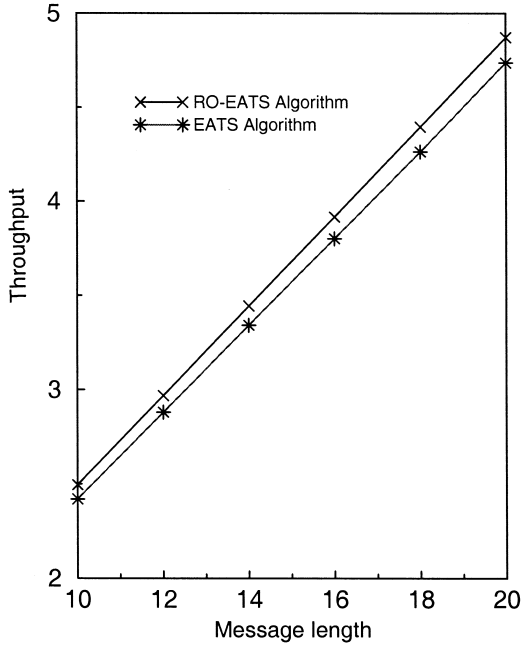


Fig. 12. System throughput versus average message length.

shown in Figs. 11 and 12 respectively. The parameters we use in this group of experiments are as follows: The number of nodes is 50 and number of channels is 4. The average message arrival rate is 0.004 for each node. The average message length varies from 10 to 20 with step of 2 time slots. In Fig. 11, we can see that as the average message length increases, the average message delay also increases. It also shows that the system using RO-EATS has smaller average message delay than that of the system using EATS. This fact has been clearly shown when the average message length reaches 20 time slots. In Fig. 12, we can see that the system using RO-EATS always has higher throughput than the system using EATS when the average message length increases.

The results shown in Figs. 11 and 12 reveal that RO-EATS has better performance than EATS no matter how the average message length changes.

Finally, we have to mention that it was shown in [9] that EATS significantly outperforms a large number of highly-regarded scheduling algorithms such as [3,5,7]. Needless to say our proposed scheduling algorithm, RO-EATS, will significantly outperform those scheduling algorithms as well.

7. Conclusions

In this paper, we proposed a new reservation-based algorithm for scheduling variable-length messages in a single-hop WDM passive star network, denoted RO-EATS. Unlike many existing reservation-based techniques, the proposed algorithm addresses *both* message sequencing and channel assignment aspects of the scheduling problem. The message selection scheme which we adopt is used to impose an order on the message sequences according to the destinations of the messages and the states of the receivers. Our proposed algorithm, RO-EATS, is shown to reduce the average message delay and increase the system throughput of a WDM network. We formulated an analytical model and conducted extensive simulations to evaluate its performance. In particular, we compared the performance of our RO-EATS algorithm with that of the EATS algorithm. The results of our comparisons show that our scheduling algorithm has significant improvements over EATS. Since EATS was shown to outperform various recently-proposed scheduling algorithms, our proposed algorithm will significantly outperform those algorithms as well.

As part of our future work, we plan to extend our research to study real-time applications in the same WDM optical network using RO-EATS. We also plan to apply other priority schemes to the other channel assignment algorithms to form more algorithms which address both message sequencing and channel assignment aspects of the scheduling problem to improve the efficiency of WDM lightwave networks.

Acknowledgements

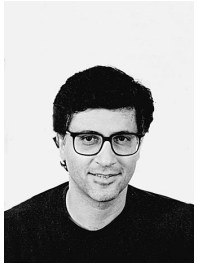
This research work was supported in part by the Hong Kong Research Grant Council under the Grant RGC/HKUST 692/96E.

References

[1] B. Mukherjee, WDM-based local lightwave networks – Part I: single-hop systems, IEEE Network (May 1992) 12–27.
[2] B. Mukherjee, WDM-based local lightwave networks – Part II: multi-hop systems, IEEE Network (July 1992) 20–32.
[3] K. Bogineni, K.M. Sivalingam, P.W. Dowd, Low-complexity multiple access protocols for wavelength-division multi-

plexed photonic networks, IEEE J. Selec. Commun. 11 (4) (1993) 590–603.
[4] A. Gantz, Y. Gao, Time-wavelength assignment algorithms for high performance WDM star based systems, IEEE Trans. Commun. 42 (1994) 1827–1836.
[5] G.N. Rouskas, M.H. Ammar, Analysis and optimization of transmission schedules for single-hop WDM networks, Proceedings of IEEE INFOCOM, March 1993, pp. 1342–1349.
[6] M.S. Borella, B. Mukherjee, Efficient scheduling of nonuniform packet traffic in a WDM/TDM local lightwave network with arbitrary transceiver tuning latencies, Proceedings of IEEE INFOCOM, 1995, pp. 129–137.
[7] K. Bogineni, P.W. Dowd, A collisionless multiple access protocol for a wavelength division multiplexed star-coupled configuration: architecture and performance analysis, J. Lightwave Technol. 10 (11) (1992) 1688–1699.
[8] R. Chipalkatti, Z. Zhang, A.S. Acampora, Protocols for optical star-coupler network using WDM: performance and complexity study, IEEE J. Selec. Commun. 11 (4) (1993) 579–589.
[9] F. Jia, B. Mukherjee, J. Iness, Scheduling variable-length messages in a single-hop multichannel local lightwave network, IEEE/ACM Trans. Networking 3 (4) (1995) 477–487.
[10] C.S. Li, M.S. Chen, F.F.K. Tong, POSMAC: a medium access protocol for packet-switched passive optical networks using WDMA, J. Lightwave Technol. 11 (5/6) (1993) 1066–1077.
[11] N. Mehravari, Performance and protocol improvements for very high-speed optical fiber local area networks using a passive star topology, J. Lightwave Technol. 8 (4) (1990) 520–530.
[12] B. Hamidzadeh, M. Ma, M. Hamdi, Message sequencing techniques for on-line scheduling in WDM networks, Proceedings of GLOBECOM, 1997, vol. 2, pp. 868–872.

Maode Ma received the B.E degree in Computer Engineering from Tsinghua University, Beijing, China, in 1982, the M.E. degree in Computer Engineering from Tianjin University, Tianjin, China, in 1991. He is currently a Ph.D. candidate in the Department of Computer Science of the Hong Kong University of Science and Technology since 1994. From 1982 to 1985, he was a project engineer in computer industry in China. From 1986 to 1991, he was a system engineer for the Department of Computer Engineering of Tianjin University. From 1991 to 1994, he was a faculty member with the Department of Computer Engineering of Tianjin University. His current research interests include WDM optical computer networks, QOS of the computer communications, real-time system scheduling, system modelling and simulation, application of queuing theory, Markov decision process, and application of discrete event system algebra.



Babak Hamidzadeh received M.S. and Ph.D. degrees in Computer Science and Engineering from The University of Minnesota in 1989 and 1993, respectively. In that period, he also worked as a research associate at The Systems and Research Center of Honeywell Inc., and as a research scientist at The Research and Technology Center of Alliant Techsystems Inc. for over 3 years. From 1993 to 1996 he was an Assistant Professor of Computer Science and Computer Engineering at The Hong Kong University of Science and Technology. Currently, he is an Assistant Professor of Electrical and Computer Engineering at The University of British Columbia. He is also a member of IEEE Computer Society. His areas of research include real-time computing, parallel and distributed processing, multimedia, and communication networks.

Mounir Hamdi received the B.Sc. degree with distinction in Electrical Engineering (Computer Engineering) from the University of Southwestern Louisiana in 1985, and M.Sc. and Ph.D. degrees in Electrical Engineering from the University of Pittsburgh in 1987 and 1991, respectively. While at the University of Pittsburgh, he was a Research Fellow involved with various research projects on interconnection networks, high-speed communication, parallel algorithms, switching theory, and computer vision. In 1991 he joined the Computer Science Department at Hong Kong University of Science and Technology as an Assistant Professor. He is now an Associate Professor of Computer Science and the Director of the Computer Engineering Programme. His main areas of research are ATM/IP packet switching architectures, high-speed networks, wireless networking, and parallel computing. Dr. Hamdi has published over 100 papers on these areas in various journals, conference proceedings, and book chapters. He is on the editorial board of the IEEE Communications Magazine and Parallel Computing, and has been on the Program Committee of more than 30 international conferences and workshops. He was guest editor of a special issue of Informatica on ''Optical Parallel Computing''. He co-founded and co-chaired the International Workshop on High-Speed Network Computing. Dr. Hamdi received the best paper award at the 12th International Conference on Information Networking. Dr. Hamdi received the ''Best Ten Lecturers Award'' and ''Teaching Excellence Award'' from the Hong Kong University of Science and Technology. Dr. Hamdi is a member of IEEE and ACM.