Efficient Message Scheduling for WDM Optical Networks with Minimizing Flow Time

Xiaohong Huang and Maode Ma

Abstract: **In this paper, we propose an efficient sequencing technique, namely minimum flow time scheduling (MFTS), to manage variable-length message transmissions for single-hop passive star-coupled WDM optical networks. By considering not only the message length but also the state of the receivers and the tuning latency, the proposed protocol can reduce the average delay of the network greatly. This paper also introduces a new channel assignment technique latency minimizing scheduling (LMS), which aims to reduce the scheduling latency. We evaluate the proposed algorithm, using extensive discrete-event simulations, by comparing its performance with shortest job first (SJF) algorithm. We find that significant improvement in average delay could be achieved by MFTS algorithm. By combining the proposed message sequencing technique with the channel selection technique, the performance of the optical network could be further improved.**

Index Terms: **Medium access control protocol, optical network, photonic switching, scheduling algorithm, wavelength division multiplexing (WDM).**

I. INTRODUCTION

Wavelength division multiplexing (WDM) is emerging as a promising technology for the next generation of high-speed communication networks. The nodes in such a network can transmit and receive messages on any of available channels using one or more tunable transmitters and/or tunable receivers. Several system structures have been proposed in [1] and [2]. A typical and simple structure is the one with a single-hop topology, which directly connects the network nodes to a passive star coupler [1].

Based on the hardware structure of a WDM optical network, efficient media access control protocols are needed to schedule the message transmission through the multiple channels of an optical fiber. Most of the protocols proposed for this WDM optical network can be divided into two different classes, namely pre-allocation-based [3]–[7] and reservation-based [8]–[14] techniques. Pre-allocation-based techniques assign transmission rights to different nodes in a static manner. While reservation-based techniques reserve one of the channels as a control channel to transmit global information among nodes in the network. Once such information is received, all nodes invoke same scheduling algorithm to determine time slots and a data channel for a message transmission. Reservation-based techniques are more dynamic in nature and assign transmission rights to the messages based on the run-time requirements of the nodes in the network. In this paper, we focus our attention on reservation-based techniques.

Traffic scheduling plays an essential role in photonic switching system. There are two fundamental aspects that the scheduling algorithm should efficiently solve, namely, channel assignment and message sequencing. The channel assignment aspect of a scheduling algorithm addresses the problem of selecting an appropriate channel and a time slot on that channel to transmit a message, while message sequencing addresses the order in which messages are selected for transmission. The channel assignment aspect of this problem has been addressed extensively in the literature. These scheduling algorithms schedule the messages individually and independently. They ignore that the way to choose the order of the message transmission may affect the performance of the network. However, the performance of the network could be further improved by the way of considering both aspects, as shown in [9] and [10]. By the protocols in [9], the order of transmission is determined by the message length. And the protocol in [10] decides the sequence of the message transmission by the information of the receiver's states. These algorithms have been shown to improve the network performance of a WDM network. However, all these algorithms just focus on one characteristic of the messages. They have not considered the whole flow time of the messages in a network. Flow time is defined as a time period that a message exists in a network, which consists of message waiting time and message transmission time. Further, the message waiting time depends on tuning overhead of transceivers and the run-time availability of destination node and transmission channel. In this paper, we propose a scheduling algorithm that determines the order of message transmissions based on the flow time of the messages, which is called minimum flow time scheduling (MFTS). This algorithm works by taking not only the message length but also the availability of the channels and the receivers into account. In this way, the algorithm can coordinate the transmission and reception of the messages efficiently while minimizing the average message delay of the network.

This paper also provides an insight into the channel assignment techniques. Earliest available time scheduling (EATS) is an algorithm which has received a lot of attention as an effective algorithm for channel assignment in WDM networks [11]. Based on this algorithm, an algorithm, named minimum scheduling latency (MSL), is proposed in [12] which is designed to reduce the scheduling latency of EATS. In this paper, we introduce a channel assignment algorithm, namely latency minimizing scheduling (LMS), which is the revised version of the MSL algorithm.

We compare the performance of MFTS with shortest job first (SJF) in a mathematical way. The results prove that our new message sequencing technique can achieve better performance

Manuscript received August 21, 2003; approved for publication by Krishna Sivalingam, Division III Editor, February 24, 2004.

The authors are with Nanyang Technological University, Singapore, email: huangxiaohong@pmail.ntu.edu.sg, emdma@ntu.edu.sg.

1229-2370/04/$10.00 © 2004 KICS

in terms of average message delay than SJF. In addition, we conduct extensive discrete-event simulations to evaluate the new algorithms. The results continuously demonstrate the superior performance of MFTS to SJF algorithm. The experiments also show that MFTS combined with LMS can further enhance the performance.

The remainder of this paper is organized as follows: Section II briefly describes the related research on scheduling variable-length messages in single-hop passive star-coupled WDM optical networks. Section III specifies our WDM system model. Section IV presents our scheduling techniques. Section V discusses the performance comparison between MFTS and SJF and Section VI shows the results from simulation experiments. Finally, Section VII concludes the paper with a summary.

II. RELATED RESEARCH

In this section, we present a brief survey of scheduling algorithms for the reservation-based protocols in single-hop passive star-coupled WDM optical networks. In general, there are two types of scheduling algorithms. One type only addresses the channel assignment aspect of scheduling. The other type addresses the combination of the message sequencing and the channel assignment for scheduling.

In [11], one protocol with several scheduling algorithms has been proposed. The protocol employs some global information of the network to prevent both data channel collisions and receiver collisions. By using global information, its ability to avoid both collisions makes this protocol a milestone in the development of MAC protocols for passive star-coupled WDM optical networks. In this paper, three data channel assignment algorithms have been proposed, namely earliest available time scheduling (EATS), contiguous destination scheduling (CDS), and transmitter tune ahead scheduling (TTAS). The fundamental one is EATS algorithm. This algorithm assigns a message to the data channel that has the earliest available time among all the channels. Once the data channel is assigned, the message is scheduled to be transmitted as soon as that channel becomes available. Based on EATS, a novel algorithm, called MSL, is proposed in [12]. This paper illustrates that if the message is destined to a very busy node, its transmission time can be scheduled much later than the data channels' earliest available time. In this case, the channel scheduling latency becomes quite large. The logic behind this algorithm is that, instead of always selecting the earliest available channel, it selects the available channel optimally by taking the destination availability into account. If the message is not destined to a busy node, the earliest available channel is selected. Otherwise, among all the channels, find a new channel subset S in which any channel can achieve the same earliest scheduled transmission time as the earliest available channel. And then, the one with the longest channel available time in S is selected. With this algorithm, the scheduling latency can be reduced. Therefore, the performance in terms of average message delay can be improved.

The above scheduling algorithms schedule message transmission individually and independently. They attempt to schedule each message immediately after receiving the control information about that message. However, in [9], the authors noticed that the performance of the network could be further improved by the way of exploiting more existing global information of the network and the transmitted messages. From this point of view, a general scheduling scheme, which combines the message sequencing techniques, such as longest-job-first (LJF) algorithm and shortest-job-first (SJF) algorithm, with the channel assignment algorithms in [11], is proposed to schedule variable-length message transmission. Sequencing messages using SJF, where shorter messages are scheduled to be transmitted before longer messages, presents the best performance in terms of message delay. In [10], as an example of the general scheduling scheme in [9], a novel scheduling algorithm is proposed. The new algorithm, named receiver-oriented earliest available time scheduling (RO-EATS), decides the sequence of the message transmission by the information of the receiver's states. The scheduling can decrease the message waiting delay caused by the scheme to avoid receiver collision. It first considers the earliest available receiver among all the nodes in the network, and then selects a message for transmission, which is destined to this receiver from those which are ready and identified by the control frame.

The works in [9] and [10] have shown that useful information for improving the schedule quality exists when messages are considered together as a batch, rather than individually, for scheduling. The algorithms that address the transmission sequence as well as the channel assignment aspect of the scheduling work better than those schedule messages transmitted individually since they not only share the global information about each message among receiving and transmitting nodes, but also consider multiple messages from different transmitting nodes simultaneously. The results of [9] and [10] have demonstrated that significant improvement in terms of average message delay for one batch of messages can be achieved by the combination of message sequencing and channel assignment. One other good point of such scheduling is that the scheduling algorithm is invoked when all the nodes in the network have received control information about one batch of messages, resulting in the reduction of the frequency in needing the scheduling algorithm to be invoked. This reduction leads to lower scheduling overheads and permits more times for transceivers' tuning.

In this paper, we proposed a new scheduling algorithm with the combination of message sequencing and channel assignment. Our new algorithm is going to further elaborate the characteristics of the specified WDM optical networks.

III. SYSTEM MODEL

The network architecture being considered in this paper is the single-hop passive star-coupled WDM optical network with a passive star coupler (PSC). The usable bandwidth is divided into $C+1$ wavelengths (λ_0, $\lambda_1, \cdots, \lambda_C$) with the same capacity. In what follows, we call each wavelength as the corresponding channel. The C channels ($\lambda_1, \cdots, \lambda_C$), referred to as data channels, are used for transmission of messages. The remaining channel (λ_0), referred to as 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-based schemes. The number of nodes N is independent of the number of channel C, but normally assumed to

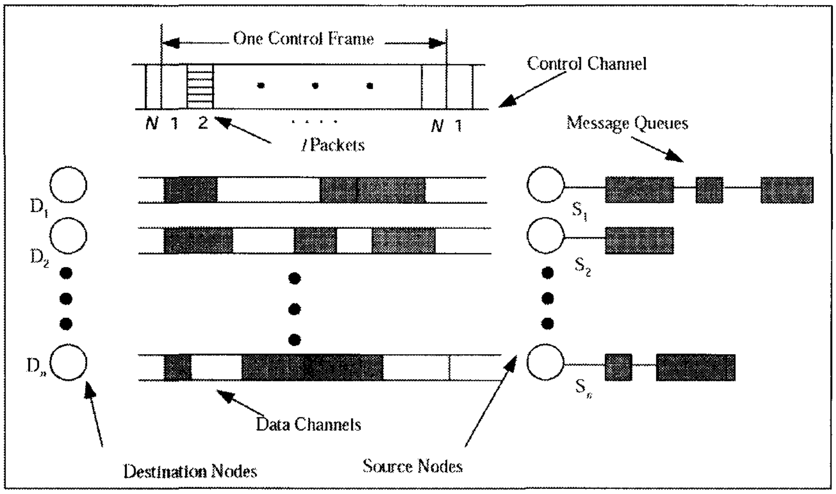

Fig. 1. Data and control channel configuration and message queues at nodes.

be much greater than C. In the network, each node is equipped with two transmitters and two receivers. One transmitter and one receiver are fixed and are tuned to the control channel (λ_0). The other transmitter and receiver are tunable and can be tuned to any data channel to access messages on those channels.

The network operates in a slotted mode, with a slot time equal to the transmission time of one fixed-length packet. The nodes are assumed to generate variable-length messages, which can be divided into several fixed-length data packets. In the network, the nodes are divided into two non-disjoint sets of source nodes S_i and destination nodes D_j. A queue of messages is assumed to exist at each source node S_i, which is shown in Fig. 1. On the control channel, time is also divided into control frames. A control frame consists of N control slots, each of which equals to the transmission time of a control packet. A time division multiplexing access (TDMA) protocol is used on the control channel to avoid channel collision. Each node has a corresponding control packet in a control frame, during which that node can access the control channel. A control packet contains the information such as source-node id, destination node address and number of data packets in a message.

The procedure of message transmission and reception in this system model works as follows: When a source node has a message at the head of its queue, the source node S_i first sends a control packet during time slot i on the control channel to all other nodes. After a round-trip propagation delay, all of the nodes will have information contained in a control frame about messages to be transmitted at all nodes. Then an identical copy of a distributed scheduling algorithm is invoked by all the nodes, which sorts the message using messages sequencing technique and assigns the messages represented in the control frame to the appropriate data channels to be transmitted at given time slots.

IV. SCHEDULING ALGORITHM

From the Section II, we have the conclusion that significant improvement can be achieved when two scheduling issues are considered simultaneously. In this section, we present a new message sequencing algorithm and a new channel assignment algorithm, which can work seamlessly during the scheduling process. Their details will be presented as follows.

A. MFTS Algorithm

MFTS is designed to reduce mean message flow time of the system. In the network, for a particular processing sequence, the mean flow time of message transmission can be

$$\begin{aligned}\overline{F} = 1/n\sum_{k=1}^{n} F_{ik} &= 1/n\sum_{k=1}^{n}(W_{ik}+l_{ik})\\ &= 1/n\sum_{k=1}^{n} W_{ik} + 1/n\sum_{k=1}^{n} l_{ik},\end{aligned}$$

where i denotes the sequence, k denotes the position of the message in the sequence, and l_{ik} and W_{ik} denote the length and the waiting time of the message which is queued in the k position of sequence i, respectively. $\sum_{k=1}^{n} l_{ik}$,which does not change no matter how the messages are queued, denotes the sum length for all the messages in the queue. Hence in order to minimize $\overline{F}$, we must minimize $\sum_{k=1}^{n} W_{ik}$.

In order to minimize the mean flow time of the whole system, we have following cases to consider:

- $W_{i1} = 0$ for all sequence since message M_{i1}, which is the message queued in the first place of sequence i, can be processed without waiting for other messages.
- $W_{i2} = F_{i1}$ since M_{i2} must wait only for M_{i1} to be processed. Thus, if we choose M_{i1} to have the shortest flow time $F_{i1} = W_{i1} + l_{i1}$ of all the messages in the list $\{M_1, M_2, \cdots, M_n\}$, we shall minimize W_{i2}.
- $W_{i3} = F_{i1} + F_{i2}$ since M_{i3} must wait only for M_{i1} and M_{i2} to be processed. In order to minimize W_{i3}, we should choose M_{i1} and M_{i2} to have the shortest and next shortest flow time from the list $\{M_1, M_2, \cdots, M_n\}$. If we let M_{i1} still have the shortest, then we do not affect our minimization of W_{i2}. Therefore, we can minimize W_{i2} and W_{i3} simultaneously.
- Continuing in this way, we can build up a schedule by which the kth message has the shortest flow time among those remaining. Simultaneously, the total waiting time $\sum_{k=1}^{n} W_{ik}$ can be minimized. Thus, minimize $\overline{F}$.

Based on the above consideration, we have an idea to generate a new message sequencing algorithm MFTS, which can reduce the mean flow time of the system significantly. In our proposed algorithm, a new priority assignment scheme is introduced, in which we consider not only the message transmission time but also the message waiting time. We have the following formula

$$F_i = W_i + l_i$$

to assign the priority to the messages that are going to be transmitted. By this priority scheme, the message with the least value of F_i will be considered as the message with the highest priority. There is one point we need to address here is that the priority is not static. After we select one with the highest priority, the waiting time of the remainders will be calculated based on the updated global information, and then the new one with the least flow time will be selected again among the rest of the messages. In this way we can assure that the flow time of the system can be kept low all the time.

B. LMS Algorithm

Based on MSL algorithm in [12], we introduce a new channel selection algorithm LMS in this paper. Same as MSL, LMS aims to reduce the schedule latency of EATS algorithm. Here we define scheduling latency as the difference between the time when the channel starts transmission and the time when the channel becomes available.

In order to determine the flow time of each message, we need some global information. One is D_j for each node. $D_j = m$, where $j = 1, \cdots, N$, means that node j's receiver will become free after m time slots. D_j is needed to avoid the receiver collisions. The other is C_k for each channel. $C_k = n$, where $k = 1, \cdots, C$, means that channel k will become available after n time slots. C_k contains the information of each channel and is needed to avoid channel collisions. Moreover, we assume that the round-trip propagation delay is R time units and the transceiver's tuning time is T time units.

We have

$$r = D_j + T, \tag{1}$$

where r is the time when node j can be ready to receive the next message. Then,

$$t_1 = \max(C_k,\ T), \tag{2}$$

$$t_2 = \max(t_1 + R,\ r),$$

where t_1 is the earliest available time that transmitting node can transmit on data channel k and t_2 is the time that destination node's receiver should be ready to receive on data channel k. So, the waiting time of the message can be expressed by

$$W_i = t_2 - R. \tag{3}$$

The flow time of message i can be obtained from

$$F_i = W_i + l_i, \tag{4}$$

where l_i is the message transmitting time.

According to the above relationships, we can easily find that if $(t_1+R) < (D_j+T) = r$, the message needs to wait longer than the channel available time due to the unavailability of destination receiver. Thus, the scheduling latency may exist. In our system, there are three factors that are related to the scheduling latency, namely channel available time, destination available time and transmitter tuning time. The problem of EATS is that it selects the channel according to channel available time regardless of the other two factors. Thus, if a message is destined to a busy node, or the channel available time is much smaller than tuning time, the latency will become large. However, if we choose a channel by taking the destination available time and the transmitter tuning overhead into account, the latency can be reduced. MSL is an algorithm which reduces the latency of EATS, but it just considers destination and channel available time regardless of the tuning time. Based on this algorithm, we introduce a new channel selection algorithm LMS, which considers all these three factors.

C. MFTS-LMS Algorithm

We combine the proposed message sequencing technique with the channel selection algorithm to form the new algorithm. We describe it formally as the following.

`Begin`:
- **Transmit** a control packet on the control channel for every node;
- **Wait** until the control frame returns;

`Start`:
- **Calculate** the flow time of every message i destined to node j not scheduled:

Step 1) Select the channel according to the new channel selection algorithm:

a. If there is a channel subset $\{C_k\}$ in which $[\max(C_k, T) + R] \leq (D_j + T)$, select the maximum $C_i \in \{C_k\}$. If there are multiple channels with the maximum available time, choose the one with the smallest channel number.

b. If there is no such $\{C_k\}$ that makes $[\max(C_k, T)+R] \leq (D_j+T)$, then

b. 1 If there is $C_k \leq T$, select the maximum $C_i \in \{C_k\}$.

b. 2 Else, select the minimum C_i among all the channels.

Step 2) Calculate $t_1 = \max(C_i, T)$ and $t_2 = \max(t_1 + R, r)$, and obtain the time to start transmission $t = t_2 - R$. So the flow time of every message waiting to be scheduled is $F_i = t + l_i$;

- **Choose** the message with the least flow time F_i;
- **Use** the selected channel to transmit the selected message and schedule the transmission time at t;
- **Update** $D_j = t_2 + l_i, C_i = t_2 - R + l_i$;
- **Return** to **Start** if any messages not scheduled;

`End`

The complexity of MFTS-LMS can be evaluated based on its operations. It has two sequencing procedures. One of the sequence procedures is to sort the messages according to their flow time. The other is to sort the C_k table. The number of the messages could be the number of nodes in the network in the worst case and the number of C_k is the number of channels in the network. Let us assume that the number of nodes (N) is always larger than the number of channels (C). Hence we consider only the number of nodes when we estimate the complexity of the sorting algorithm. The complexity of a typical sorting algorithm is $O(N\log_2 N)$, where N can be mapped to the number of nodes in our case. Given that the worst case running time of the algorithm is two rounds of the sequence procedure, the complexity of the entire algorithm should be $O(N\log_2 N)$.

Table 1. Current global information.

$C_1 = 5,\ C_2 = 13,\ C_3 = 15,\ C_4 = 0$
$D_{j1} = 9,\ D_{j2} = 20,\ D_{j3} = 10,\ D_{j4} = 10,\ D_{j5} = 0$
$T = 2,\ R = 4$

Table 2. Information of messages.

Messages	M_1	M_2	M_3	M_4	M_5
Destination node j	j_1	j_2	j_3	j_4	j_5
Packet length L	7	9	10	11	13

Table 3. Using SJF-EATS algorithm.

Messages	M_1	M_2	M_3	M_4	M_5
Destination node j	j_1	j_2	j_3	j_4	j_5
D_{jk}	9	20	10	10	0
Packet length L	7	9	10	11	13
Selected channel	C_4	C_1	C_2	C_4	C_3
W_i	7	18	13	14	15
F_i	14	27	23	25	28
C_k (updated)	14	27	23	25	28
D_{jk} (updated)	18	31	27	29	32
$\overline{F}$	$(14 + 27 + 23 + 25 + 28)/5 = 23.4$				

The bandwidth of the transmission link is defined as the number of bits transmitted per unit time. Since in our network, we assume that one packet can be transmitted in one time slot, 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 is being transmitted. The transmission time of all messages of one control frame can be approximately by $m \times N$, where N is the number of nodes in the network and m is the mean message length. Therefore, the condition that our scheduling algorithm will not produce cost in terms of bandwidth can be formulated as $2 \times N\log_2 N \leq m \times N$. 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.

D. Example

In order to illustrate the new algorithm effectively, we present an example in this subsection. The current global information is given in Table 1. And the information of the messages is given in Table 2, which contains five messages $M_1, M_2, \cdots, M_5$, destined to nodes $j_1, j_2, \cdots, j_5$, respectively. In this example, the number of data channel (C) is equal to 4, tuning time of the transceiver (T) is 2, and the propagation delay (R) is set to 4.

We start our discussion by observing the behavior of SJF algorithm in this example. According to SJF algorithm, messages M , $M_2, \cdots, M_5$ are sorted to M_1, M_2, M_3, M_4, and M_5, which is shown in Table 3. The channel selection algorithm used in this example is EATS. Firstly, running of the EATS algorithm will assign M_1 to C_4. From Table 2, the destination available time for M_1, namely D_{j1}, equals to 9. Comparing $(D_{j1} + T)$ with $(C_4 + R)$, it is easily observed that the waiting time of the message M_1 is dependent on the destination available time and equal to 7 time slots according to (2) and (3). The flow time of the message M_1 is 14 time slots (7 time slots for waiting plus 7 time slots for transmission). After scheduling, the global information should be updated ($C_4 = 14$ and $D_{j1} = 18$). At the current time, C_1 becomes the earliest available channel. Message M_2 is then assigned to it. According to (2) and (3), the waiting time of message M_2 is also dependent on the destination available time and equal to 18 time slots. Thus, the flow time of M_2 is 27 time slots (18 time slots for waiting plus 9 time slots for transmission). Similarly, the global information should be updated ($C_1 = 27$ and $D_{j2} = 31$). The flow times of the rest of the messages can be calculated by following the same way. And the final result is shown in Table 3. The average flow times of these five messages can be calculated as: $(14 + 27 + 23 + 25 + 28)/5 = 23.4$.

Next we apply our new algorithm MFTS combing with EATS to our example. Firstly, we calculate the flow times of all these five messages based on the global information which is shown in Table 1. Currently, channel C_4 has the earliest available time. According to MFTS algorithm, we have to calculate the flow time of each message, and then assign the highest priority to the one with the least flow time. For message M_1, the destination available time (D_{j1}) is equal to 9. Comparing $(D_{j1} + T)$ with $(C_4 + R)$, we can have that the waiting time for M_1 is equal to 7 time slots according to (2) and (3). Then M_1's flow time is equal to 14 time slots (7 time slots for waiting plus 7 time slots for transmission). For M_2, the destination available time (D_{j2}) is 20. So M_2's waiting time is equal to 18 time slots. The flow time of M_2 is 27 time slots (18 time slots for waiting plus 9 time slots for transmission). Following the same way, we can get the flow times of all the five messages at the first round of sorting, which are 14 (M_1), 27 (M_2), 18 (M_3), 19 (M_4), and 15 (M_5). Then M_1 will be the one with the least flow time and it will be scheduled first. After scheduling, the global information should be updated ($C_4 = 14$ and $D_{j1} = 18$). Then, based on the new global information, we can calculate the flow times of the rest of the four messages. At this time every message will select C_1 for transmission due to its earliest available time. Following the method as above, we can get the flow times of these four messages, which are 27 (M_2), 18 (M_3), 19 (M_4), and 18 (M_5). In this case, both M_3 and M_5 have the least flow time. We can select M_3 in the second round of sorting and assign the higher priority to it. By following this way, we finally sort the five messages to M_1, M_3, M_4, M_2, and M_5. The average flow time can be acquired from $(14 + 18 + 24 + 27 + 28)/5 = 22.2$, which is shown in Table 4. Until now we can find that MFTS can achieve better performance in terms of mean flow time than SJF. One feature of MFTS is that it doesn't assign priority at the beginning of scheduling. Instead it assigns the priority during the scheduling process.

At last, we discuss the performance result of MFTS-LMS algorithm with this example. In the first round of sorting, the respective flow time of every message will be calculated based on LMS algorithm. For example, for M_1, there are C_1 and

Table 4. Using MFTS-EATS algorithm.

Messages	M_1	M_3	M_4	M_2	M_5
Destination node j	j_1	j_3	j_4	j_2	j_5
D_{jk}	9	10	10	20	0
Packet length L	7	10	11	9	13
Selected channel	C_4	C_1	C_2	C_4	C_3
W_i	7	8	13	18	15
F_i	14	18	24	27	28
C_k (updated)	14	18	24	27	28
D_{jk}(updated)	18	22	28	31	32
$\overline{F}$	$(14+18+24+27+28)/5 = 22.2$				

Table 5. Using MFTS-LMS algorithm.

Messages	M_1	M_5	M_3	M_4	M_2
Destination node j	j_1	j_5	j_3	j_4	j_2
D_{jk}	9	0	10	10	20
Packet length L	7	13	10	11	9
Selected channel	C_1	C_4	C_2	C_1	C_3
W_i	7	2	13	14	18
F_i	14	15	23	25	27
C_k (updated)	14	15	23	25	27
D_{jk}(updated)	18	19	27	29	31
$\overline{F}$	$(14+15+23+25+27)/5 = 20.8$				

C_4 that satisfy $[\max(C_k, T) + R] \leq (D_{j1} + T)$, so we will select C_1 according to LMS. After that, according to (3) and (4), we will calculate the flow time of M_1, which equals to 14 time slots. While for M_2, all of the four channels satisfy $[\max(C_k, T) + R] \leq (D_{j2} + T)$. Then in order to reduce the scheduling latency, channel C_3 with the largest channel available time will be selected. And the flow time of the message is 27 time slots. By following the same way, we get the flow times of five messages that are 14 (M_1), 27 (M_2), 18 (M_3), 19 (M_4), and 15 (M_5), respectively in the first round of sorting. Thus, the message M_1 with the least flow time will be assigned with the highest priority. After scheduling M_1, we start the second round of calculation on the rest of the messages based on the updated global information. Similarly, the flow times of the remaining messages can be obtained, which are 27 (M_2), 18 (M_3), 19 (M_4), and 15 (M_5). Therefore, at this round, the message M_5 with the least flow time will be selected. According to LMS, the channel C_4 with the least channel available time will be assigned to this message. Finally, we can sort the five messages to M_1, M_5, M_3, M_4, and M_2. The final result is listed in Table 5. From the table, the average flow time can be easily calculated from $(14 + 15 + 23 + 25 + 27)/5 = 20.8$. Comparing this result with the result of MFTS-EATS above, we can find that LMS algorithm works better than EATS.

V. PERFORMANCE COMPARISON

In this section, we aim to compare the performance of MFTS and SJF in terms of mean flow time by adjacent pairwise interchange [15], [16]. We consider an SJF sequence in which there must be a pair of adjacent messages M_1 and M_2, with M_1 preceding M_2 due to $l_1 \leq l_2$, where l is the transmission time of the message. Then, we construct a new sequence, S', in such a way that only messages M_1 and M_2 are sorted according to MFTS algorithm with other messages in S unaltered. We aim to prove that a strict improvement of performance can be achieved by the new sorting method. If we can show that the performance of the SJF sequence can be improved with respect to mean flow time by sorting an adjacent pair of messages in the way of MFTS algorithm, we can continue the sorting until eventually the MFTS sequence is constructed. The important part of our comparison is to prove the new sequence S' can achieve smaller mean flow time than original sequence S.

All the situations are depicted in Fig. 2. We assume that in sequence S, M_1 precedes M_2. And in this figure, we assume that after sorting according to MFTS, M_2 precedes M_1. In Fig. 2, A denotes the set of messages that precede messages M_1 and M_2 in both sequences and B denotes the set of messages that follow M_1 and M_2 in both sequences. As we have mentioned above, messages of A and B will be left unchanged after the new sorting. C_k denotes the channel available time at which message M_1 begins to hold the channel in S and at which message M_2 begins to hold the channel in S'. In the sequence S, $F_1(S)$ is the flow time of message M_1. $F_1(S)$ also equals to the channel available time at which message M_2 begins to hold the channel in S, which is denoted by C'_k. The flow time of message M_2 in S is denoted by $F_2(S)$. Similarly, $F_1(S')$ is the flow time of message M_1 and $F_2(S')$ is the flow time of message M_2 in sequence S'.

We temporarily adopt the notation $F_k(S)$ to represent the flow time of message k under schedule S and $F_k(S')$ to represent the flow time of message k under schedule S'. Then,

$$\sum_{k=1}^{N} F_k(S) = \sum_{k\in A} F_k(S) + F_1(S) + F_2(S) + \sum_{k\in B} F_k(S),$$
$$\sum_{k=1}^{N} F_k(S') = \sum_{k\in A} F_k(S') + F_2(S') + F_1(S') + \sum_{k\in B} F_k(S')$$

And because $\sum_{k\in A} F_k(S) = \sum_{k\in A} F_k(S')$ and $\sum_{k\in B} F_k(S) = \sum_{k\in B} F_k(S')$, we can get

$$\sum_{k=1}^{N} F_k(S) - \sum_{k=1}^{N} F_k(S') = F_1(S) + F_2(S) - F_2(S') - F_1(S').$$

Therefore, we need to prove $F_1(S) + F_2(S) - F_2(S') - F_1(S') = F_S - F'_S \geq 0$.

Proof: In sequence S, since $l_1 \leq l_2$, M_1 will be served before M_2. In sequence S', the message with less flow time will be served first, so we need to compare the flow time of these two messages.

i) If $F_1 \leq F_2$, MFTS can get the same sequence of these two messages by SJF, so the mean flow time of S' is the same with that of S.
ii) If $F_1 > F_2$, according to MFTS, M_2 will precede M_1. The situation is same with S' shown in Fig. 2.

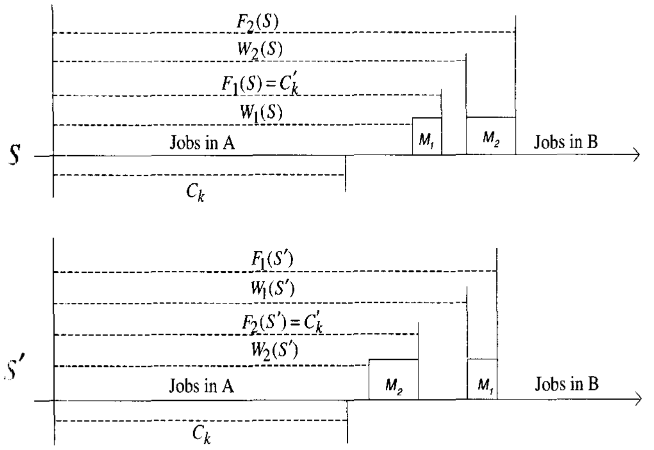

Fig. 2. Sorting of adjacent messages.

In order to prove that $F_S - F_S' \geq 0$ when $F_1 > F_2$, we make the following assumptions: The number of channels is 1; the receiver available times for M_1 and M_2 are D_1 and D_2, respectively; the tuning time T equals to 0. And we have the conditions that $l_1 \leq l_2$ and $F_1 > F_2$.

To obtain the flow time of these two messages, let's look at the relationship between $C_k + R$ and D_j. According to (2) and (3), we can easily get that if $C_k + R > D_j$, the flow time depends on channel available time C_k, otherwise, the flow time is determined by destination available time D_j. Based on the above assumptions, we have following properties that:

a. It is impossible for $C_k + R > D_1$ when $F_1 > F_2$.

The proof is as follows. In this case, we can have $W_1 = C_k$ and $F_1 = C_k + l_1$. Then we try to get the value of F_2. For every message that is sorted at time C_k, the least waiting time equals to C_k. So the waiting time for M_2 will not be less than C_k, namely $W_2 \geq C_k$. Then $F_2 = W_2 + l_2 \geq C_k + l_2$. And from $l_1 \leq l_2$, it is clearly that $(F_2 = W_2 + l_2) \geq (F_1 = C_k + l_1)$, which contradicts with $F_1 > F_2$. Therefore, $C_k + R > D_1$ cannot be true.

b. If $C_k + R \leq D_1$, the flow time of message M_1 depends on the receiver available time D_1, and then the flow time of M_1 is $F_1 = D_1 - R + l_1$.

c. If $C_k + R \leq D_2$, the flow time of the message M_2, namely F_2, depends on the receiver available time D_2, and $F_2 = D_2 - R + l_2$.

d. If $C_k + R > D_2$, the flow time of the message M_2, namely F_2, depends on channel available time C_k, and $F_2 = C_k + l_2$.

According to the above properties, there will be two cases to evaluate F_S and F_S'.

Case 1: $C_k + R \leq D_1$ and $C_k + R \leq D_2$.

From $F_1 > F_2$ and property b and c, we have the relationship as follows,

$$D_1 - R + l_1 > D_2 - R + l_2 \Rightarrow D_1 + l_1 > D_2 + l_2. \quad (5)$$

In S:

M_1 will precede M_2. So the C_k' for M_2 equals to $F_1(S)$, as shown in Fig. 2. And from property b, we have $F_1(S) = D_1 - R + l_1$, then, $C_k' + R = F_1(S) + R = D_1 - R + l_1 + R = D_1 + l_1$ From (5), we can get $C_k' + R > D_2 + l_2 > D_2$.

So the flow time of message M_2 in sequence S is dependent on channel available time C_k', and $F_2(S) = C_k' + l_2 = F_1(S) + l_2$.

Then, the flow time of the two messages, M_1 and M_2, is $F_S = F_1(S) + F_2(S) = 2(D_1 - R + l_1) + l_2$.

In S':

In S', M_2 precedes M_1. From property c, we have $F_2(S') = D_2 - R + l_2$. The C_k' for M_1 equals to $F_2(S')$, and $C_k' + R = F_2(S') + R = D_2 - R + l_2 + R = D_2 + l_2$. Then, in order to get the flow time of M_1 in sequence S', we need to compare $C_k' + R$ with its destination available time D_1.

1) If $D_2 + l_2 > D_1$, $C_k' + R > D_1$. And, $F_1(S') = C_k' + l_1 = F_2(S') + l_1 = D_2 - R + l_2 + l_1$. So, $F_S' = F_2(S') + F_1(S') = 2(D_2 - R + l_2) + l_1$.
2) If $D_2 + l_2 < D_1$, $C_k' + R < D_1$, and $F_1(S') = D_1 - R + l_1$. So, $F_S' = F_2(S') + F_1(S') = D_2 - R + l_2 + D_1 - R + l_1$.

Comparison:

1) $F_S - F_S' = 2(D_1 - R + l_1) + l_2 - 2(D_2 - R + l_2) - l_1 = 2(D_1 + l_1) - 2(D_2 + l_2) + l_2 - l_1$. From (5) and $l_2 \geq l_1$, we have $F_S - F_S' > 0$, namely, $F_1(S) + F_2(S) - F_2(S') - F_1(S') > 0$.
2) $F_S - F_S' = 2(D_1 - R + l_1) + l_2 - (D_2 - R + l_2 + D_1 - R + l_1) = D_1 - D_2 + l_1$. From (5), we can easily get $F_S - F_S' > 0$, namely, $F_1(S) + F_2(S) - F_2(S') - F_1(S') > 0$.

So, in this case, sequence S' can get less flow time F than S.

Case 2: $C_k + R \leq D_1$ and $C_k + R > D_2$

From $F_1 > F_2$ and property b and d, we have

$$D_1 - R + l_1 > C_k + l_2. \quad (6)$$

In S:

M_1 will be served first. From property b, we get $F_1(S) = D_1 - R + l_1$. Then, the channel available time for M_2 will be C_k', which equals to $F_1(S)$. And from $C_k + R \leq D_1$ and $C_k + R > D_2$, we can get $D_1 \geq D_2$. So, $C_k' + R = F_1(S) + R = D_1 - R + l_1 + R = D_1 + l_1 > D_2$. Thus, $F_2(S) = C_k' + l_2 = F_1(S) + l_2$. Consequently, we obtain $F_S = F_1(S) + F_2(S) = 2(D_1 - R + l_1) + l_2$.

In S': For the S', M_2 precedes M_1. From property d, we can easily get $F_2(S') = C_k + l_2$. As for M_1, the channel available time for it, C_k', equals to $F_2(S')$, then we need to compare $(C_k' + R)$ with D_1, where $C_k' + R = F_2(S') + R = C_k + l_2 + R$.

1) If $C_k + l_2 + R > D_1$, we can have $F_1(S') = C_k' + l_1 = C_k + l_2 + l_1$. Thus, $F_S' = F_2(S') + F_1(S') = 2(C_k + l_2) + l_1$.
2) If $C_k + l_2 + R < D_1$, we have $F_1(S') = D_1 - R + l_1$. Then, $F_S' = F_2(S') + F_1(S') = C_k + l_2 + D_1 - R + l_1$.

Comparison:

1) $F_S - F_S' = 2(D_1 - R + l_1) + l_2 - 2(C_k + l_2) - l_1$. From (6) and $l_1 \leq l_2$, we have $F_S - F_S' > 0$.
2) $F_S - F_S' = 2(D_1 - R + l_1) + l_2 - (C_k + l_2 + D_1 - R + l_1) = D_1 - R + l_1 - C_k$. From $C_k + R \leq D_1$, we have $F_S - F_S' > 0$.

In this case, sequence S' can get less flow time F than S, too. Therefore, sequence S' can achieve less flow time than sequence S in both cases, which proves that SJF algorithm can be improved with respect to mean flow time by sorting an adjacent pair of messages in the way of MFTS algorithm. It follows that the overall MFTS sequence can achieve better mean flow time than SJF. □

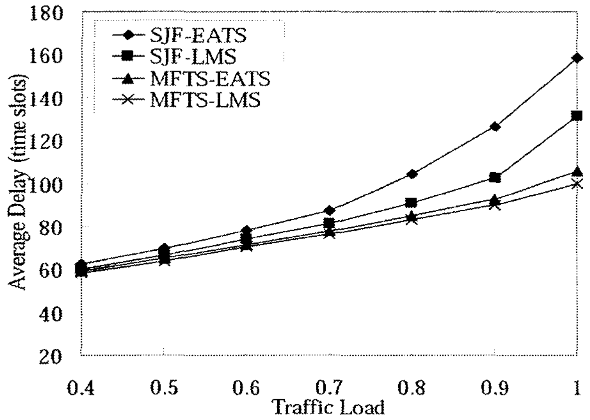

Fig. 3. Average delay versus traffic load.

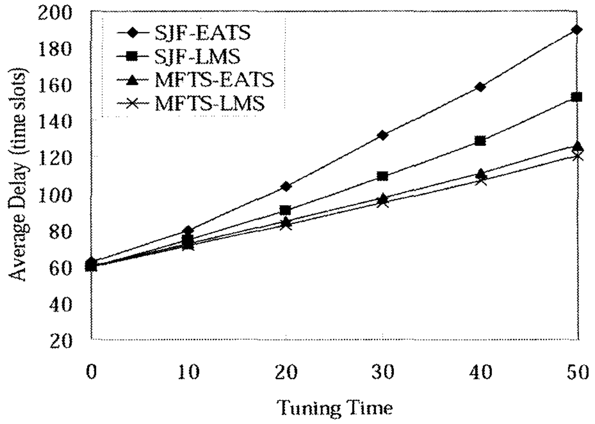

Fig. 4. Average delay versus tuning time.

VI. SIMULATION RESULTS

In this section, we carry out the performance study of the proposed scheduling algorithm MFTS. We also compare it with the SJF algorithm by extensive simulation experiments. The objectives of the simulation are twofold. First, we demonstrate that MFTS achieves better performance in terms of average delay than SJF. Second, we demonstrate that the new channel selection algorithm LMS works better than EATS. The following subsections provide a discussion of the design of these experiments and their results.

A. Experimental Design

The parameters involved in the system design include the number of nodes (N), which is set to 50. Round-trip propagation delay (R) equals to 10 time units. Message lengths vary according to an exponential distribution with a mean value as 20 time slots. Packet arrivals at each source node comply with independent and identical Poisson processes. Traffic load across all nodes ranges from 0.4 to 1. The destination nodes are selected according to a uniform probability distribution. To evaluate the performance of computer networks, average message delay is normally adopted as one major metric, as shown in [8]–[14]. In our current research work, we follow the same way as other researchers to take average message delay as the metric to evaluate the performance of the scheduling algorithms over the networks. The message delay is defined as the duration from the time a message is scheduled to the time the message finishes transmission.

B. Experimental Results

Fig. 3 compares the average message delay using four algorithms under varying loads in a system. In this set of experiments, we assume that the number of channels (C) is 4 and the tuning time (T) is 20. Since the propagation delay (R) is always part of the message delay in all of the algorithms, it will not be included in the message delay. Thus, the delay incurred by queuing, tuning time overhead, and message transmission time will be our focus. From the figure, we can see that in the low traffic load, the difference among these four algorithms is not significant. However, as the traffic load increases, MFTS-EATS and MFTS-LMS significantly outperform the other two algorithms. For example, when traffic load is 1.0, the delay with MFTS-EATS is only 66% of that with SJF-EATS. This is because the MFTS algorithm sorts the messages according to the flow time, therefore resulting in delay reduction. Also, the figure shows the difference between EATS and LMS. It is easy to see that LMS can work better than EATS in terms of delay due to the minimized scheduling latency.

Fig. 4 compares the characteristic of average message delay under varying tuning time. In this set of experiments, the number of channels (C) is fixed to 4 and the traffic load is set to 0.8. It is easy to see that the delay increases with the tuning time for these four algorithms. This is because that when the tuning time increases, the source nodes take longer time to complete a message transmission. Among these four algorithms, MFTS consistently demonstrates its superior performance to SJF in terms of delay. The difference in message delays becomes more significant when tuning time increases. This is expected since MFTS considers the flow time of the message rather than only the message length. The LMS is introduced mainly to minimize the scheduling latency. Therefore, MFTS-LMS and SJF-LMS can achieve lower delay than MFTS-EATS and SJF-EATS, respectively.

Fig. 5 shows the effect of varying channel number on the average delay for four algorithms. We assume that the tuning time (T) is fixed to 20 and the traffic load is set to 0.8. It is clear that the delay decreases for all of the algorithms when the number of channels increases. Once again, we observe that MFTS achieves better performance than SJF in terms of delay while using the same channel selection algorithm, especially when the number of channels is small. This is because, when the number of channel is small, the competition for channels becomes drastic and the effect of the message sequencing becomes important. MFTS considers message flow time rather than message length, so it has lower delay. When the number of channels becomes large, the performance of all the algorithms will flatten out and further increase in the number of channels will not induce any change, which means that data channels are no longer a bottleneck. From the figure, we can see that combing with the same

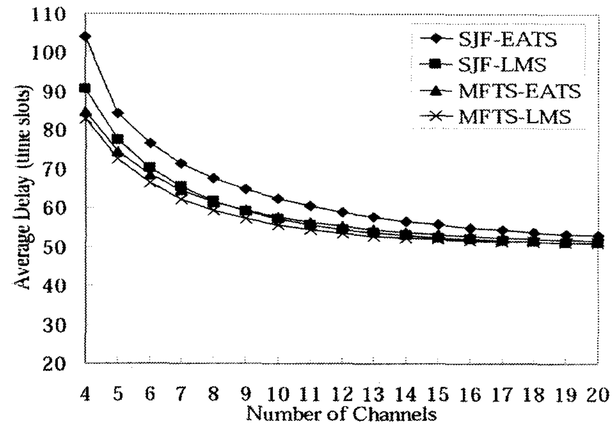

Fig. 5. Average delay versus number of channels.

channel selection algorithm, MFTS can achieve better performance than SJF, while combining with the same message sequencing algorithm, LMS can achieve better performance than EATS.

VII. CONCLUSIONS

In this paper, we have proposed a new scheduling algorithm MFTS for the reservation-based MAC protocol in single-hop passive star-coupled WDM optical networks, which considers the message waiting time and message transmission time simultaneously. The studies have shown that MFTS can reduce the average message delay of the network dramatically. In addition, we have introduced a new channel assignment algorithm LMS, which considers tuning time of the transceivers in the scheduling. It has been shown that LMS works better than EATS. We have also evaluated the performance of different message sequencing techniques combing with different channel selection algorithms. Overall, the MFTS-LMS algorithm obtains the best performance in the comparison.

REFERENCES

[1] B. Mukherjee, "WDM-based local lightwave networks- part I: Single-hop systems," *IEEE Network*, pp. 12–27, May 1992.

[2] B. Mukherjee, "WDM-based local lightwave networks- part II: Multi-hop systems," *IEEE Network*, pp. 20–32, July 1992.

[3] I. Chlamatac and A. Ganz, "Channel allocation protocols in frequency-time controlled high speed networks," *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 430–440, Apr. 1988.

[4] A. Ganz and Y. Gao, "Time-wavelength assignment algorithms for high performance WDM star based systems," *IEEE Trans. Commun.*, vol. 42, no. (2/3/4), pp. 1827–1836, Feb./Mar./Apr. 1994.

[5] G. N. Rouskas and M. H. Ammar, "Analysis and optimization of transmission schedules for single-hop WDM networks," *IEEE/ACM Trans. Networking*, vol. 3, no. 2, pp. 211–221, Apr. 1995.

[6] M. Azizoglu, R. A. Barry, and A. Mokhtar, "Impact of tuning delay on the performance of bandwidth-limited optical broadcast networks with uniform traffic," *IEEE J. Select. Areas Commun.*, vol. 14, no. 6, pp. 935–944, June 1996.

[7] H. Y. Tyan *et al.*, "On supporting temporal quality of service in WDMA-based star-coupled optical networks," *IEEE Trans. Comput.*, vol. 50, no. 3, pp. 197–214, Mar. 2001.

[8] K. Bogineni and P. W. Dowd, "A collisionless multiple access protocol for a wavelength division multiplexed star-coupled configuration: Architecture and performance analysis," *IEEE/OSA J. Lightwave Technol.*, vol. 10, no. 11, pp. 1688–1699, Nov. 1992.

[9] B. Hamidzadeh, Maode Ma, and M. Hamdi, "Efficient sequencing techniques for variable-length messages in WDM networks," *IEEE/OSA J. Lightwave Technol.*, vol. 17, no. 8, pp. 1309–1319, Aug. 1999.

[10] Maode Ma, B. Hamidzadeh, and M. Hamdi, "An efficient message scheduling algorithm for WDM lightwave networks," *Computer Networks*, vol. 31, no. 20, pp. 2139–2152, Sept. 1999.

[11] F. Jia, B. Mukherjee, and J. Iness, "Scheduling variable-length messages in a single-hop multichanel local lightwave network," *IEEE/ACM Trans. Networking*, vol. 3, no. 4, pp. 477–487, Aug. 1995.

[12] J. Diao and P. L. Chu, "Packet rescheduling in WDM star networks with real-time service differentiation," *IEEE/OSA J. Lightwave Technol.*, vol. 19, no. 12, pp. 1818–1828, Dec. 2001.

[13] H. C. Lin and C. H. Wang, "A hybrid multicast scheduling algorithm for single-hop WDM networks," *IEEE/OSA J. Lightwave Technol.*, vol. 19, no. 11, pp. 1654–1664, Nov. 2001.

[14] H. C. Lin and P. S. Liu, "Dynamic control frames in reservation-based packet scheduling for single-hop WDM networks," in *Proc. IEEE ICPP 2003*, 2003, pp. 87–95.

[15] S. French, *Sequencing and Scheduling: An Introduction to the Mathematics of the Job-shop*, New York: Wiley, 1950.

[16] K. R. Baker, *Introduction to Sequencing and Scheduling*, New York: Wiley, 1974.

Xiaohong Huang received her BE degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2000. She is currently the Ph.D candidate in the school of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. Her research interests include WDM optical networks, photonic packet switching, scheduling algorithm, and differentiated services.

Maode Ma received his BE degree in automatic control from Tsinghua University in 1982, ME degree in computer engineering from Tianjin University in 1991, and Ph.D. degree in computer science from Hong Kong University of Science and Technology in 1999. He started his professional career as an engineer in computer industry in 1982. Starting from 1991, he was an assistant professor in the department of computer engineering at Tianjin University. Since 2000, Dr. Ma has joined the school of electrical and electronic engineering at Nanyang Technological University in Singapore as an assistant professor. Dr. Ma has published more than 20 academic papers in the area of WDM optical networks and other relevant fields. His current research interests are performance analysis of computer networks, optical networks, and wireless networks.