Pipelined Versus Non-pipelined Traffic Scheduling in Unidirectional WDM Rings*

Xijun Zhang[1] and Chunming Qiao[2]

[1] Department of Electrical Engineering (EE)
University at Buffalo (SUNY), Buffalo, NY 14260
xz2@eng.buffalo.edu
[2] Departments of Computer Science and Engg. (CSE) and EE
University at Buffalo (SUNY), Buffalo, NY 14260
qiao@computer.org

Abstract. Two traffic scheduling schemes, namely, non-pipelined and pipelined scheduling, are considered for uniform traffic in unidirectional WDM rings. Their performances in terms of schedule length and throughput are compared. Optimal or near optimal schedules are obtained for a given number of wavelengths, and a given number of transceivers per node as well as their tuning time.

1 Introduction

Due to the lack of sophisticated optical logic devices, communications in all-optical Wavelength Division Multiplexed (WDM) networks are normally carried out in a circuit-switching fashion to avoid conversions of data between electronic and optical domains at intermediate nodes. In addition, in order to reduce the control overhead involved in dynamically establishing connections, one may schedule pre-determined communication patterns. This basically leads to hybrid Time/Wavelength Division Multiplexed access wherein a connection will be scheduled on a particular wavelength (assuming no wavelength conversion at any intermediate node) during a particular time slot.

A typical communication pattern is uniform, all-to-all traffic, which requires each node in a network to send a unique message to every other node. Such a pattern arises in many important parallel computing algorithms. Furthermore, it is usually more beneficial to use a highly tuned schedule for the uniform traffic pattern than attempting to find a schedule for a non-uniform but dense traffic pattern [1, 2]. The schedule length is an important performance metric as it affects, for example, the throughput that can be achieved in the network.

The problem of traffic scheduling in conventional networks has been studied extensively (see, for example, [1–8]). Traffic scheduling in broadcast-and-select WDM networks based on star-couplers has also received much attention (see,

* This research is supported in part by a grant from NSF under contract number MIP-9409864 and ANIR-9801778.

for example, [9–12]). In this paper, we study the problem of scheduling traffic in WDM rings.

Our work (including those in [13–15]) differs from other closely related work (including those in [16, 17]), in that we consider WDM rings with a limited number of wavelengths per link, and/or a limited number of transceivers per node. The traffic scheduling problem addressed in this work is unique (and more challenging) also because the tuning time of each transceiver will be considered. Furthermore, we will consider pipelined transmissions, which is made possible by two unique properties of optical transmissions, namely, unidirectional propagation and predictable propagation delay [18, 19].

The rest of this paper is organized as follows. In Section 2, we give a general description of the WDM ring network to be considered as well as the traffic scheduling problem. Non-pipelined and pipelined transmission scheduling schemes are studied in Section 3 and Section 4, respectively. Numerical results on their performance in terms of schedule length and throughput are given in Section 5. Finally, we conclude the paper in Section 6.

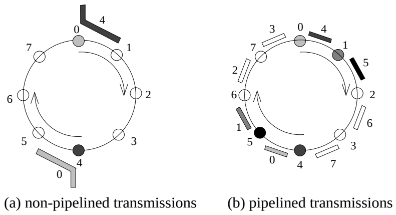
2 Problem Description

We consider a unidirectional (e.g. clockwise) WDM ring having K wavelength channels on each link and N nodes. At each node, there are T fully tunable transceivers whose maximum tuning time is Δ seconds. Since the transceivers can be pre-tuned, we will ignore the *initial* tuning time hereafter and accordingly, consider the effect of Δ only when $T < K$. We assume that there is no wavelength conversion. For simplicity, it is also assumed that the propagation delay between any two adjacent nodes is d seconds, and the network is globally synchronized. We will study the scheduling problem for all-to-all uniform traffic, in which each node has one message of M bits to be sent to each of the other $N - 1$ nodes.

Due to limited network resources, not all the messages can be sent/received simultaneously. Accordingly, our objective is to determine a priori a schedule which includes all the necessary information (such as the time to transmit and receive, and the wavelength to be used). The schedule length, denoted by L_s, is the time needed for all the messages to be received.

In this paper, we will consider two traffic scheduling schemes, namely, non-pipelined and pipelined, as illustrated in Figure 1. More specifically, assume that the transmission rate on each wavelength is B bits per second. In non-pipelined transmission, when it is node i's turn to transmit to another node s hops away, ($s = 4$ in Figure 1), it transmits the entire message. Accordingly, $\frac{M}{B} + s \cdot d$ seconds are needed for the message to be received. During this period, those intermediate nodes cannot transmit on the same wavelength On the other hand, in pipelined transmission, all nodes can transmit on the same wavelength simultaneously. In order to avoid collision, each node only sends a packet of p bits at a time, where $\frac{p}{B} \leq d$, and hence, needs to send q packets for each message, where $p \cdot q = M$.

In the remainder of the paper, we will examine non-pipelined and pipelined traffic scheduling schemes in more detail, determine optimal or near optimal



(a) non-pipelined transmissions (b) pipelined transmissions

Fig. 1. (a) Non-pipelined and (b) Pipelined transmission (the number on each message/packet denotes the destination node).

schedules for each scheme, and compare the performance of the two schemes in terms of the schedule length as well as the network throughput.

3 Non-pipelined Scheduling

The maximum propagation time over all connections in the N-node unidirectional ring is $(N-1) \cdot d$, which will be denoted by t_p. We will study non-pipelined scheduling under the assumption that M is large relative to t_p. A non-pipelined schedule can be regarded as a sequence of *rounds* (i.e. super time-slots). In each round, it is desirable to schedule as many all-optical connections as possible, one for each message to be transmitted, in order to fully utilize the I/O capacity of the transceivers, the bandwidth of the wavelengths, or both. A message is transmitted over a connection only at the beginning of each round, and a node will not send a new message on the same wavelength until the previous message from this node has been received. To facilitate synchronization, the duration of each round, R, is made equal to the sum of the message transmission time and the maximum propagation delay, i.e. $R = \frac{M}{B} + t_p$. Accordingly, if the schedule requires r rounds, then the schedule length L_s is the sum of $r \cdot R$ and any tuning time.

Although message transmissions occur only at the beginning of each round, the tuning of a transmitter (and receiver) may start immediately after a message is transmitted (and received, respectively), and hence can overlap with message propagation. More specifically, given the tuning time Δ, we may define *tuning delay* $\delta = 0$ if $\Delta \leq t_p$ and $\delta = \Delta - t_p$ if otherwise.

3.1 Negligible Tuning Delay

We start with the simple case where $\delta = 0$. To facilitate our presentation, a connection is said to have *stride* s if a message is to be transmitted to a node s hops away. Two connections with common end nodes (e.g. one from i to j and the other from j to i) have complementary strides, s and $N - s$, and thus form a *circle* and are scheduled in the same round to fully utilize the bandwidth of one wavelength.

Multiple circles are scheduled in the same round to fully utilize the bandwidth of multiple wavelengths. The maximum number of circles that can be scheduled in a round, however, is limited by both the number of wavelengths, K, and the number of transceivers at each node, T. Accordingly, the *theoretical lower bound* (TLB) on the schedule length depends on both K and T. Specifically, the TLB imposed by T is $TLB_T = \lceil \frac{N-1}{T} \rceil \cdot R$, and the TLB imposed by K is $TLB_K = \lceil \frac{\frac{N(N-1)}{2}}{K} \rceil \cdot R$ [15]. The overall TLB is thus $TLB(K, T, \delta = 0) = max\{TLB_K, TLB_T\}$.

Single transceiver per node

We first consider the case where $T = 1$. Assuming N is odd, one method to enumerate all the connections is shown in Figure 2, in which a straight line with an arrow from node i to node j is used to represent the circle involving nodes i and j. Circles are grouped into *virtual topologies (VTs)* such that each VT contains only parallel lines as shown in the figure. This way, there are $\frac{N-1}{2}$ circles in each VT, and every node except one is involved in one circle in a VT. There are N VTs in total, and they have similar patterns but contain different connections. In general, rotating the connections in VT_i by $j \geq 1$ nodes clockwise results in $VT_{mod(i+j,N)}$ (where $mod()$ is the modular function, which will be omitted hereafter in the subscript of all VT's).

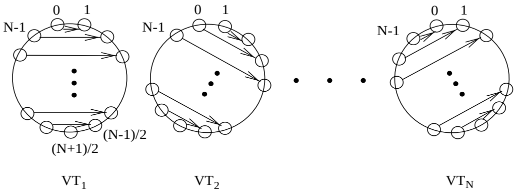


Fig. 2. Virtual topologies representing all the connections when N is odd.

Since $T = 1$, two circles can be scheduled in the same round only if they do not involve common nodes. Given that each node is involved in at most one circle in any VT, all the circles in one VT can be scheduled in one round as long as K is large enough. If K is less than the number of circles in each VT, the circles in one VT have to be scheduled in multiple rounds. Specifically, any K of the $\frac{N-1}{2}$ circles in a VT can be scheduled in one round. Let $\frac{N-1}{2} = mK + y$. After $m = \lfloor \frac{\frac{N-1}{2}}{K} \rfloor$ rounds (and a total of $m \cdot N$ rounds for the N VTs), there will be $y = mod(\frac{N-1}{2}, K) < K$ circles left in each VT. When $y > 0$, in order to fully utilize the bandwidth, the circles left in different VTs can also be scheduled in the same round if they do not involve common nodes.

Let x be the number of circles in each VT, $y = mod(x, K)$, and z be the number of VTs. An upper bound on the number of rounds needed has been obtained as [15],

$$f(x,z,K) = \begin{cases} \frac{x}{K} \cdot z & x \geq K,\ y = 0 \\ \lfloor \frac{x}{K} \rfloor z + \lceil \frac{\lceil \frac{z}{y} \rceil y}{K} \rceil mod(z,y) + \lceil \frac{\lfloor \frac{z}{y} \rfloor y}{K} \rceil [y - mod(z,y)] & x \geq K,\ y > 0 \\ z & x < K \end{cases} \quad (1)$$

Accordingly, the upper bound on the schedule length when $T = 1$ for an odd N is $UB(K, T = 1, \delta = 0) = f(\frac{N-1}{2}, N, K) \cdot R$. The case where N is even is similar except that we have to consider connections having even strides and odd strides (or circles formed by such connections) separately.

Note that the circles left can be packed *recursively* using a procedure which we call "recursive packing procedure"[15]. The schedule length achieved using the recursive packing procedure will be less than the upper bound given above.

Multiple transceivers per node

When $T > 1$ transceivers are used at each node, more circles may be scheduled in each round. A straightforward extension of the scheduling method proposed for the case of $T = 1$ is to combine T VTs to form a *super*-VT such that the I/O capacity provided by the T transceivers per node can be fully utilized by scheduling the circles in the super-VT in the same round when K is large enough. Note that N may not be divided evenly by T, in which case, the remaining $mod(N,T)$ VTs can be considered separately. As an example, when N is odd (the case when N is even is similar), there are $\frac{N-1}{2} \cdot T$ circles within each super-VT, and there are $\lfloor \frac{N}{T} \rfloor$ such super-VTs. Therefore,

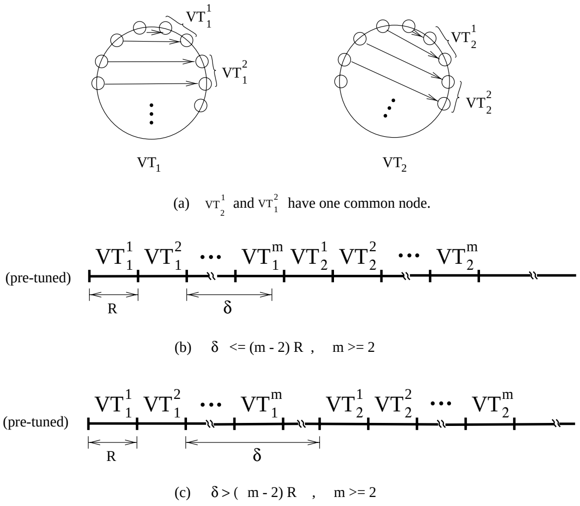
$$UB(K, T, \delta = 0) = [f(\frac{N-1}{2} \cdot T, \lfloor \frac{N}{T} \rfloor, K) + \lceil \frac{\frac{N-1}{2} \cdot mod(N,T)}{K} \rceil] \cdot R \quad (2)$$

3.2 Non-negligible Tuning Delay

In this section, we consider the effects of non-negligible tuning delay (i.e. $\delta > 0$) on the schedule length for the simplest case where N is odd and $T = 1$. We derive an upper bound under the most conservative assumption that any two consecutive transmissions by a node are always scheduled on different wavelengths. In other words, we assume that a node always has to wait for δ seconds before any transmission (including the first one).

Recall that after m rounds, there are y circles left in each VT, and the only possible value of y that is larger than $\frac{N}{4}$ is $y = \frac{N-1}{2}$. In this case, each VT has to be scheduled separately in one round. Since the I/O capacity is fully utilized in each round by each VT, there has to be a delay of δ before each round (except the first round with pre-tuned transceivers). Accordingly, the upper bound on the schedule length is $UB(K, T = 1, \delta) = N \cdot R + (N - 1) \cdot \delta$.

When $y = 0$, there are exactly $m \cdot K$ circles in each VT, and m rounds are needed to schedule them. Let VT_i^h, where $1 \leq h \leq m$, represent the circles scheduled in round h for VT_i (see Figure 3(a) for an illustration of VT_1^1, VT_1^2, VT_2^1 and VT_2^2). Note that $VT_1^1, VT_1^2, \cdots, VT_1^m$ can be scheduled with no tuning delay. However, since VT_2^1 is obtained by rotating VT_1^1 by one node clockwise, it shares one common node with VT_1^2 (and some common nodes with VT_1^1, but no common node with VT_1^{1+k} if $k \geq 2$). This means that VT_2^1 cannot be scheduled until δ seconds after VT_1^2. If $\delta \leq (m-2)R$, the tuning delay can be hidden by overlapping it with the rounds in which VT_1^3 through VT_1^m are scheduled (see Figure 3(b)). However, if $\delta > (m-2)R$, the round for VT_2^1 has to be delayed for $\delta - (m-2)R$ seconds after the round for VT_1^m (see Figure 3(c)).


VT$_1^1$
VT$_1^2$
VT$_1$
VT$_2^1$
VT$_2^2$
VT$_2$
(pre-tuned)
VT$_1^1$ VT$_1^2$ ••• VT$_1^m$ VT$_2^1$ VT$_2^2$ ••• VT$_2^m$
R
δ
(pre-tuned)
VT$_1^1$ VT$_1^2$ ••• VT$_1^m$ VT$_2^1$ VT$_2^2$ ••• VT$_2^m$
R
δ

(a) VT$_2^1$ and VT$_1^2$ have one common node.

(b) δ <= (m - 2) R , m >= 2

(c) δ > (m - 2) R , m >= 2

Fig. 3. Non-pipelined scheduling with non-negligible tuning delay.

Summarizing the above discussion, an upper bound on the schedule length when $C = mK$, where $m \geq 2$, is:

$$UB(K, T=1, \delta) =$$

$$\begin{cases} N \cdot m \cdot R & \delta \leq (m-2)R,\ m \geq 2 \\ N \cdot m \cdot R + (N-1) \cdot [\delta - (m-2)R] & \delta > (m-2)R,\ m \geq 2 \end{cases} \qquad (3)$$

When $0 < y \leq \frac{N}{4}$, the first $m \cdot K$ circles in each VT can be scheduled in the same way as described above. Afterwards, a similar method can be applied to schedule the $N \cdot y$ circles left. By overlapping tuning of transceivers with as many rounds as possible, the effects of tuning delay can be reduced or even eliminated.

4 Pipelined Scheduling

Pipelined transmission allows a batch of N packets, one from each node, to be transmitted on the same wavelength at the same time. These N packets are sent over connections with the same stride s (e.g. $s = 4$ in Figure 1(b)) such that they will be received simultaneously. When multiple batches are to be transmitted on the same wavelength, the transmission of the second or later batch can start as soon as the previous batch is being received. Accordingly, when calculating the time needed to send and receive multiple batches of packets on the same wavelength, one may include the transmission time of the first batch ($\frac{p}{B}$), and omit that of all subsequent batches. Hereafter, in order to simplify our presentation, we will focus on the so-called ***sub-schedule length*** and the corresponding ***sub-schedules***. The former, denoted by L'_s, is the time for each node to send to (and receive from) every one of the other $N-1$ nodes ***a single packet*** (instead of q packets), excluding the time to transmit the first packet. Based on the above discussion, the schedule length, L_s, is equal to $\frac{p}{B} + q \cdot L'_s$ (may be shorter if several sub-schedules can be overlapped partially as to be discussed later).

In pipelined transmission, the behavior of every node is the same at any given time. Accordingly, we will only need to consider how to schedule transmissions at one node. In addition, even though a node cannot transmit (or receive) another packet on the same wavelength in $s \cdot d$ seconds, the transmitter (or receiver) is free to tune to a different wavelength to transmit (or receive) another packet. However, when the tuning time Δ is relatively large, tuning a transceiver once every packet time will result in a high overhead.

In the following discussion of pipelined transmission, we use the notion of a time slot, whose duration is d seconds, instead of the notion of a round used before. We will study pipelined transmission under the assumption that $\Delta \le d$. In particular, we will describe the following two approaches. One uses mixed tuning and transmission slots by choosing p such that $\frac{p}{B} + \Delta = d$. The other is to use a separate slot for tuning followed by a slot for transmitting a packet of $d \cdot B$ bits. Intuitively, the former approach is suitable when Δ is small compared to d. When Δ is a significant portion of d, the latter may be more suitable.

4.1 Mixed Tuning and Transmission Slots

In this subsection, we study the first approach whereby tuning and packet transmission can take place in one slot. When using this approach, the actual value of Δ will have no effect on the sub-schedule length (although the larger the Δ, the smaller the p and the larger the q, and accordingly, the longer the schedule length and the lower the throughput). We will first study the case where $T = K$, then extend the discussion to two other cases where $2 \le T < K$ and $T = 1$, respectively.

The case where $T = K$

When $T = K$, each transceiver can be fixed at a distinct wavelength, and thus no tuning is needed except for initialization. Since the total time needed for a node to send one packet to every other node is $\frac{N(N-1)}{2}$ slots on a single wavelength, the TLB on the sub-schedule length is $TLB_p(T = K) = \lceil \frac{\frac{N(N-1)}{2}}{K} \rceil \cdot d$, which may be achieved if all packet transmissions can be uniformly distributed among K wavelengths.

Our idea for approaching the TLB_p is to group the transmissions into *phases* with a fixed number of slots, L_p. More specifically, on each wavelength, up to two transmissions with strides s and $L_p - s$ can be scheduled in each phase. Figure 4 shows an example schedule with $L_p = N = 17$ slots, where each row corresponds to a wavelength, and each arrowed line segment corresponds to a connection with a specific stride. In this example, $TLB_p(T = K = 4) = 34 \cdot d$, which is achieved in two phases since the sub-schedule length is $L'_s = 2 \cdot L_p \cdot d = 34 \cdot d$.

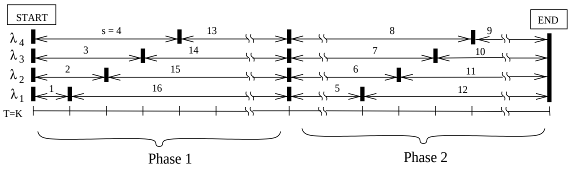


Fig. 4. An example of pipelined scheduling $(N = 17, T = K = 4)$.

Note that in some cases, combining connections with strides s and $N - s$ to form phases with $L_p = N$ may not be appropriate. In general, the appropriate value of L_p depends on N and K as to be discussed next. In order to find an efficient sub-schedule, we first determine how big K needs to be.

Theorem 1. *At most $\lceil \frac{N}{2} \rceil$ wavelengths are needed for scheduling pipelined transmissions when $T = K$.*

Proof. Since the largest s is $N - 1$, the sub-schedule length cannot be shorter than $N - 1$ slots. Without loss of generality, assume that the connection with stride $N - 1$ is scheduled on wavelength λ_1. Let (s_1, s_2, λ_i) denote that two connections with strides s_1 and s_2 are scheduled on wavelength λ_i, where $2 \leq i \leq K$. A straightforward way to schedule the remaining $N - 2$ connections is $(1, N-2, \lambda_2), (2, N-3, \lambda_3), \cdots$, and so on. This sub-schedule requires $\lceil \frac{N}{2} \rceil$ wavelengths in total, and the sub-schedule length achieved is $(N - 1) \cdot d$, the shortest schedule possible. □

Note that the proof above also implies an optimal sub-schedule for the case where $K = \lceil \frac{N}{2} \rceil$, which is accomplished in one phase with a length of $L_p = N - 1$ slots. When $K < \lceil \frac{N}{2} \rceil$, let $N - 1 = 2mK + \mu$, where $m = \lfloor \frac{N-1}{2K} \rfloor$ and $\mu = mod(N - 1, 2K)$. It turns out that the sub-schedule length L'_s will be minimized if each phase has a length of $L_p = N + \mu$ slots[14]. The resulting sub-schedule is illustrated in Figure 5, where each number inside the squares denotes the stride of a connection, and an arrow indicates the direction at which the strides increase.

From Figure 5, it is clear that when $T = K$, an upper bound on the sub-schedule length is,

$$UB_p(T = K) = [m(N + \mu) + \mu] \cdot d \tag{4}$$

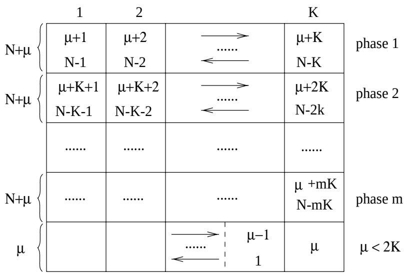


Fig. 5. One heuristic sub-schedule in pipelined transmission.

and the upper bound on the schedule length is thus simply $\frac{p}{B} + q \cdot UB_p(T = K)$.

In the next two subsections, we will only consider $K < \lceil \frac{N}{2} \rceil$ because when $K = \lceil \frac{N}{2} \rceil$, $m = 0$ and the $N - 1$ connections can always be scheduled in the same way as the last μ connections, as to be described next.

The case where $2 \le T < K$

When $2 \le T < K$, in order to use all the wavelengths available, a transmitter has to transmit on different wavelengths in an *interleaved* way. Specifically, it has to transmit one packet on one wavelength at the beginning of one slot, and then another packet on another wavelength at the beginning of another slot, and so on. Since each transmitter interleaves among $\lceil \frac{K}{T} \rceil$ wavelengths, $\lceil \frac{K}{T} \rceil - 1$ more slots are needed to send a packet on each of the K wavelengths comparing with the case where $T = K$. Intuitively,

$$TLB_p(2 \le T < K) = TLB_p(T = K) + (\lceil \frac{K}{T} \rceil - 1) \cdot d \qquad (5)$$

We now consider how to modify the sub-schedule in Figure 5 to approach the above TLB_p. First, we divide the K wavelengths into $\lceil \frac{K}{T} \rceil$ groups such that the ith group, where $1 \le i < \lceil \frac{K}{T} \rceil$, contains the following T wavelengths, $\lambda_{K-(i-1)T}$, $\lambda_{K-(i-1)T-1}$, $\cdots$, λ_{K-iT+1}, and the last group (i.e. the $\lceil \frac{K}{T} \rceil$-th group) contains only $mod(K, T)$ wavelengths, namely, $\lambda_{mod(K,T)}$, $\lambda_{mod(K,T)-1}$, $\cdots$, λ_2, λ_1. Let the T transmitters transmit on the first group of wavelengths at the beginning of each phase, and then on the second group, and so on. Note that there will be two transmissions of complementary strides on each wavelength in each phase. The first transmissions in phase 1 on the wavelengths in group i ($1 \le i \le \lceil \frac{K}{T} \rceil$) will be delayed for $i - 1$ slots relative to the case where $T = K$ (see Figure 6(a)). If we denote by $D_T(k)$ the delay of the first transmission on λ_k in phase 1, where $1 \le k \le K$, we have, $D_T(k) = \lceil \frac{K-k+1}{T} \rceil - 1$.

In fact, the first m phases in the sub-schedule in Figure 5 can be simply modified by delaying every transmission on λ_k by $D_T(k)$ slots. This is possible, or in other words, the resulting sub-schedule is valid, because of the following theorem (whose proof is omitted).

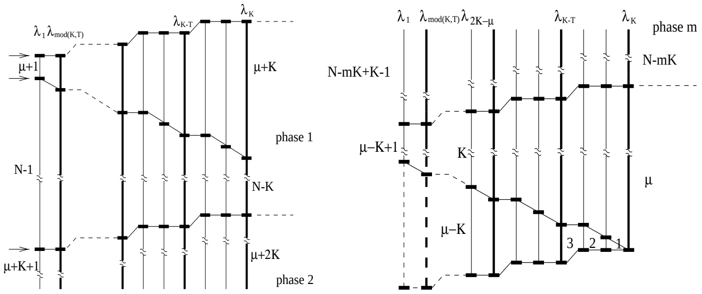


Fig. 6. (a) Phase 1 when $2 \leq T < K$ (left) and (b) Schedule of the remaining μ connections when $2 \leq T < K$ and $K < \mu < 2K$ (right).

Theorem 2. *In the modified sub-schedule, at most T packets are transmitted and received simultaneously by a node in the first m phases.*

We can schedule the remaining μ (where $0 < \mu < 2K$) connections, in the order of decreasing strides, as follows. When $K < \mu < 2K$, we schedule the first K of them on λ_K through λ_1, and the remaining $\mu - K$ of them on $\lambda_{2K-\mu}$ through λ_{K-1} (see Figure 6(b)). In this way, the connection that ends last is the one with stride $\mu - K$, which is scheduled on $\lambda_{2K-\mu}$ with a delay of $D_T(2K-\mu)$ slots. When $0 < \mu \leq K$, we schedule the μ connections on λ_μ through λ_1. Accordingly, the connection with stride μ ends last on λ_μ with a delay of $D_T(\mu)$ slots. The overall delay relative to the case where $T = K$ for different μ is thus:

$$D_T = \begin{cases} D_T(2K-\mu) & K < \mu < 2K \\ D_T(\mu) & 0 < \mu \leq K \\ D_T(1) & \mu = 0 \end{cases} \tag{6}$$

and based on Eq. (4), the modified sub-schedule will have a length of

$$UB_p(2 \leq T < K) = [m(N+\mu) + \mu + D_T] \cdot d \tag{7}$$

Note that in the above sub-schedule, when $\mu = 0$, transmission on the first group of wavelengths (containing λ_K) starts and ends earlier than that on the last group (containing λ_1) by $D_T(1)$. Accordingly, when the sub-schedule needs to be repeated (e.g. in order for each node to send multiple packets to every other node), two consecutive sub-schedules can overlap with each other partially just as the two phases do in Figure 6(a). As a result, the schedule length is only $D_T(1)$ (instead of $q \cdot D_T(1)$) longer than the schedule length when $T = K$. Even when $\mu > 0$, it is possible for two consecutive sub-schedules to overlap partially, and hence, the overall schedule length when $T < K$ will be increased by less than $q \cdot D_T$ when compared to the schedule length when $T = K$.

The case when $T = 1$

We may first derive the following by extending Eq. (5),

$$TLB_p(T=1) = TLB_p(T=K) + (K-1) \cdot d \tag{8}$$

However, we cannot simply extend the interleaved scheduling method proposed for the case where $2 \leq T < K$, which will cause multiple transmissions (and receptions) to start at the same time when $T = 1$. A possible solution is to reverse the order at which the wavelengths are interleaved. That is, we will let the transmitter transmit on λ_1 first, λ_2 next, and so on (at the beginning of each phase), which results in $D_1(k) = k - 1$ (instead of $K - k$). In short, as long as $\mu > K - 2$, the first m phases (phases 1 through m) outlined in the sub-schedule shown in Figure 5 can be modified by delaying every transmission on λ_k by $D_1(k)$ slots. However, if $\mu \leq K - 2$, only phases 2 through m can be modified this way, and the connections scheduled in phase 1, together with the remaining μ connections, will be scheduled using a heuristic method. One such heuristic method, for example, is to consider the connections in the order of decreasing stride, and try to schedule each connection as soon as possible.

4.2 Using Separate Tuning and Transmission Slots

When $T < K$ and the tuning time Δ is a large portion of d, it makes more sense to use a separate slot for tuning of transceivers. Obviously, the (sub-)schedule length will be longer than that when $\Delta = 0$. However, the (sub-)schedule length is the same for any $0 < \Delta < d$ as long as a dedicated slot is used for tuning.

The heuristic method mentioned above for the case when $T = 1$ can be modified to ensure that each transceiver has a slot for tuning before transmitting/receiving a packet on a different wavelength. Note that, when adopting this approach, it is implied that not every slot for packet transmissions has to be preceded by a tuning slot. In addition, the ratio of time used for transmission over that used for tuning could be larger. Accordingly, a better performance than that obtained by using mixed tuning and transmission slots may be obtained as to be shown next.

5 Numerical Results

In this section, we present some results on the schedule length (L_s) and throughput, which is $\frac{N(N-1)\cdot M}{L_s}$, achieved in non-pipelined and pipelined scheduling. By default, we assume that $B = 10$ Gbps (equivalent to OC-192) and $M = 100$ Mbits (accordingly, the default *message time* is $\frac{M}{B} = 10ms$). In addition, $N = 19$ and $d = 50\mu s$ (accordingly, the distance between two adjacent nodes is 10 km, and the maximum delay t_p is $900\mu s$).

Figure 7(a) shows the schedule lengths achieved, as a function of T and K, with all other parameters set to their default values (note that, the results are applicable to any $\Delta \leq 900\mu s$ since δ will be 0). As can be seen, in most cases, the TLB is achieved, and can be closely approached otherwise. In addition, as long as $K \leq \frac{N}{2}$, there is no need to use more than $T = 1$ transceivers per node since the schedule performance will not be improved.

The throughputs in non-pipelined transmission scheduling when $T = 1$ and $\Delta = 5\mu$s is shown in Figure 7(b) for different values of M and d. Curve (1) shows

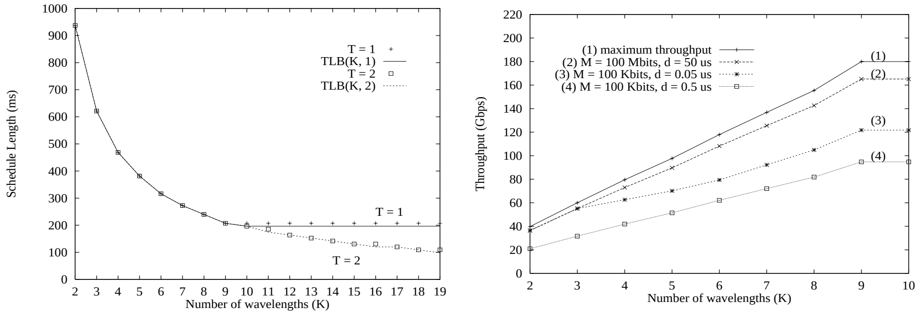


Fig. 7. Non-pipelined scheduling: (a) Schedule lengths achieved using the recursive packing procedure when $\Delta = 0$; (b) Throughput achieved for different values of M and d when $T = 1$ and $\Delta = 5\mu s$.

the maximum throughput that would be achieved if both the propagation delay and tuning delay were ignored. When propagation delay is considered, one needs to have a large ratio of message time over t_p (or $\frac{M}{B} >> t_p$) in order to achieve a high throughput. For example, curve (2) shows that, with the default M and d, and accordingly, the ratio of message time over t_p is $\frac{100}{9}$ and $\delta = 0$ (since $\Delta < t_p$), a high throughput can be achieved. In fact, our results (not shown) indicate that the same throughput can be achieved when both M and d (and thus the message time and t_p, respectively) are reduced proportionally, e.g. by 100 times (i.e. to $M = 1$ Mbits and $d = 0.5\mu$s), as long as δ remains 0. However, having a large ratio of message time over t_p is not sufficient for achieving a high throughput. For example, if we reduce the two by too much, e.g. 1000 times, that is, to $d = 0.05\mu$s (and $M = 100$ kbits), we have $\delta = 4.1\mu s$ and consequently, the throughput will be much lower as shown by curve (3). Note that, if we keep the same M (e.g. to 100 kbits) but increase d to 0.5μs to make $\delta = 0$, the ratio decreases and consequently, an even lower throughput will be achieved (see curve (4)) since the propagation delay is increased.

Figure 8(a) shows the schedule lengths achieved using pipelined scheduling with the same default settings of the parameters as in Figure 7(a). Since $\Delta = 0$, we have set $p = d \cdot B$ and $q = \frac{M}{p}$. Based on the discussion in Section 4, we have limited the number of wavelengths to be less than $\lceil \frac{N}{2} \rceil$. Note that, only $TLB_p(T = K)$ is shown in Figure 8(a) since the TLB on the schedule length for any given T is almost the same when $K \leq \lceil \frac{N}{2} \rceil$ (however, unlike in non-pipelined scheduling, increasing the number of transceivers, e.g. from 1 to 2, will reduce the schedule length in pipelined scheduling). The results show that the TLB is closely approached. In addition, pipelined scheduling results in a shorter schedule length than non-pipelined scheduling.

The throughputs achieved in pipelined scheduling for various Δ (and T and K) with the default settings of all other parameters (and an appropriate p and q) are shown in Figure 8(b). For the purpose of making comparisons, non-pipelined throughput for any T (up to $K = 10$) and any $0 \leq \Delta \leq d$ (or any $\Delta \leq t_p$) is also

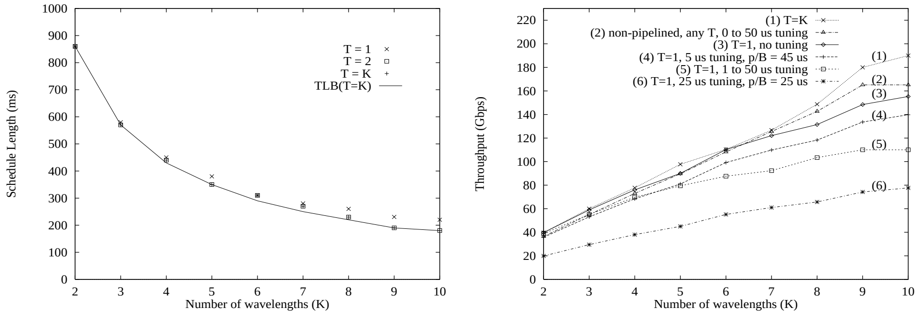


Fig. 8. Pipelined scheduling: (a) Schedule length when $\Delta = 0$; (b) Throughput for various Δ values.

shown as curve (2). Note that this curve is the same as curve (2) of Figure 7(b), even though the latter is for $T = 1$ and $\Delta = 5\mu s$.

As can be seen from curve (1), when $T = K$, (and hence with any tuning time), pipelined scheduling achieves a higher throughput. Note that, based on Figure 8(a) and related discussion, it is also conceivable that with a large T and a small Δ, pipelined scheduling can achieve a higher throughout than non-pipelined scheduling. In fact, curve (3) shows that even when $T = 1$ (as long as $\Delta = 0$), the pipelined throughout can be higher when $K \leq 6$.

On the other hand, curves (4) through (6) show that pipelined throughput can be lower than non-pipelined throughput when T is small (e.g. $T = 1$) and Δ is non-zero. These results also show that in such cases, the approach corresponding to curve (4), which combines tuning ($5\mu s$) and packet transmission ($45\mu s$) in one slot ($50\mu s$), is better when Δ is small, but the other approach which uses a separate slot for tuning is better when Δ is large (e.g. $25\mu s$).

6 Conclusion

In this paper, we have studied the problem of scheduling non-pipelined and pipelined transmissions for uniform traffic in unidirectional WDM rings, taking into consideration of a limited number of wavelengths, K, a limited number of transceivers per node, T, and their tuning time Δ. We have determined the TLBs on the schedule length, and proposed scheduling methods which can achieve optimal or near optimal schedules for various cases. We have also found that in general, pipelined scheduling is more suitable if T is equal (or close) to K or Δ is small relative to the propagation delay between the two adjacent nodes (which limits the size of the packets in pipelined transmission), but otherwise, non-pipelined scheduling is more suitable especially if message size (M) is large relative to t_p, the maximal propagation delay in the ring network. Our results can also be used to determine the amount of resources (e.g. K and T) for a given throughput requirement to achieve cost-effectiveness.

References

1. S. Hinrichs, C. Kosak, D. R. O'Hallaron, T. M. Stricker, and R. Take, "An architecture for optimal all-to-all personalized communication," in *Proc. Sixth Annual ACM Symposium on Parallel Algorithms and Architecture (SPAA)*, pp. 310–319, 1994.
2. S. H. Bokhari, "Multiphase complete exchange: A theoretical analysis," *IEEE Transactions on Computers* **45**(2), pp. 220–229, 1996.
3. C.-T. Ho and M. T. Raghunath, "Efficient communication primitives on hypercubes," in *Proc. Sixth Distributed Memory Concurrent Computers*, pp. 390–397, 1991.
4. D. Scott, "Efficient all-to-all communication patterns in hypercube and mesh topologies," in *Proc. Sixth Distributed Memory Concurrent Computers*, pp. 398–403, 1991.
5. S. Johnsson and C.-T. Ho, "Optimal broadcasting and personalized communication in hypercubes," *IEEE Transactions on Computers* **38(9)**, pp. 1249–1268, 1989.
6. Y.-J. Suh and S. Yalamanchili, "All-to-all personalized exchange in two-dimension and 3-dimension tori," in *10th International Parallel Processing Symposium*, 1996.
7. L. Tassiulas and J. Joung, "Performance measures and scheduling policies in ring networks," *IEEE/ACM Transactions on Networking* **3**(5), pp. 576–584, 1995.
8. E. Varvarigos and D. P. Bertsekas, "Communication algorithm for isotropic tasks in hypercubes and wraparound meshes," *Parallel Computing* **18**, pp. 1233–1257, 1992.
9. G. R. Pieris and G. H. Sasaki, "Scheduling transmissions in WDM broadcast-and-select networks," *IEEE/ACM Transactions on Networking* **2**(2), pp. 105–110, 1994.
10. H. Choi, H.-A. Choi, and M. Azizoglu, "Optimum transmission scheduling in optical broadcast networks," in *Proc. Int'l Conference on Communication*, pp. 266–270, 1995.
11. M. S. Borella and B. Mukherjee, "Efficient scheduling of nonuniform packet traffic in a WDM/TDM local lightwave network with arbitrary transceiver tuning latencies," in *Proceedings of IEEE Infocom*, pp. 129–137, 1995.
12. G. N. Rouskas and V. Sivaraman, "On the design of optimal TWDM schedules for broadcast WDM networks with arbitrary transceivers tuning latencies," in *Proceedings of IEEE Infocom*, pp. 1217–1224, 1996.
13. C. Qiao, X. Zhang, and L. Zhou, "Scheduling all-to-all connections in WDM rings," in *SPIE Proceedings, All Optical Communication Systems: Architecture, Control and Network Issues*, pp. 218–229, November 1996.
14. X. Zhang and C. Qiao, "Pipelined transmission scheduling in all-optical TDM/WDM rings," in *IC3N'97*, pp. 144–149, 1997.
15. X. Zhang and C. Qiao, "Scheduling in unidirectional WDM rings and its extensions," in *SPIE Proceedings, All Optical Communication Systems: Architecture, Control and Network Issues III*, pp. 208–219, 1997.
16. A. Elrefaie, "Multiwavelength survivable ring network architectures," in *Proc. Int'l Conference on Communication*, pp. 1245–1251, 1993.
17. J.-C. Bermond, L. Gargano, S. Perennes, A. A. Rescigno, and U. Vaccaro, "Efficient collective communication in optical networks," in *International Colloquium on Automata, Languages, and Programming*, pp. 574–585, 1996.
18. C. Qiao and R. G. Melhem, "Time-division optical communications in multiprocessor arrays," *IEEE Transactions on Computers* **42**(5), pp. 577–590, 1993.
19. R. G. Melhem, D. Chiarulli, and S. P. Levitan, "Space multiplexing of waveguides in optically interconnected multiprocessor systems," *The Computer Journal* **32**(4), pp. 362–369, 1989.