

On Slotted WDM Switching in Bufferless All-Optical Networks

Soung Y. Liew and H. Jonathan Chao
Polytechnic University, Brooklyn, NY 11201
syliew@queens.poly.edu, chao@poly.edu

Abstract

The current λ -switching technology requires each lightpath to occupy the full bandwidth of a wavelength throughout the bufferless all-optical domain. With such a constraint, the granularity of bandwidth is so coarse that the utilization of optical fibers is considerably low. To resolve the granularity problem, a simple way is to incorporate time division multiplexing (TDM) into wavelength division multiplexing (WDM) so as to divide the entire λ -bandwidth into smaller base bandwidths. This approach is referred to as the slotted WDM (sWDM). In this paper, we define the slot labeling, mapping and assignment problems, and derive the necessary and sufficient condition that validates sWDM. We also provide simulation results of call blocking rate of sWDM in comparison with λ -switching.

1. Introduction

In traditional packet-routed networks, traffic streams are switched electrically in the network routers in a packet-switching fashion. This switching technology is mature and highly flexible. Electrical packet routers, however, do not adapt well to optical networks because of the high cost of opto-electro-optical (OEO) conversion.

Optical packet switching (OPS) has been proposed to eliminate the necessity for OEO conversion in optical packet-routed networks [1]. In OPS, packets arrive at each switching node in an uncoordinated fashion. Optical buffering is thus necessary because packets may contend for common resources at the same time. The implementation of optical buffering is nevertheless a major problem of OPS.

To get rid of optical buffering¹ and compensate for the low configuration speed of optical switching devices, optical burst switching (OBS) takes advantage of the one-way reservation protocol such as just-enough-time (JET) and just-in-time (JIT) [2]-[4]. In OBS, a request signal is sent in advance to set up a connection by reserving an appropriate amount of bandwidth and configuring the

switches along the path. Without waiting for an acknowledgement for the connection establishment, the request is followed by a data burst. If resources are not reserved successfully at some node due to contention, the burst is dropped. In general, if traffic is heavy, the burst loss rate of OBS could be so high that the bandwidth reservation is ineffective. To resolve part of the contention issue, OBS switches can employ wavelength converters. The problem is then the high cost of wavelength converters because the related technology is not mature yet.

Another all-optical switching paradigm is optical circuit switching (or, λ -switching) [5],[6]. In the all-optical λ -switched network, the connectivity is provided for lightpaths (wavelength channels) in an end-to-end basis, and each lightpath must occupy the full λ -bandwidth of the same wavelength along all links it traverses. This is called the wavelength continuity constraint. The λ -switching scheme is transparent, bufferless and lossless. The main drawback, however, is the inefficient use of network resources, which is a direct consequence of the coarse granularity of bandwidth.

One way of resolving the granularity problem of all-optical networks is to incorporate time division multiplexing (TDM) into wavelength division multiplexing (WDM) so as to divide the entire λ -bandwidth into smaller base bandwidths [7],[8]. This approach is referred to as the slotted WDM (sWDM). However, without optical buffering, the slot alignment is the major challenging issue to sWDM. Table 1 gives a comparison between the four all-optical switching paradigms.

In this paper, we address the implementation issues of sWDM, such as guard-time overhead, slot labeling and mapping, clock signaling, slot assignment. We also derive the necessary and sufficient condition that, in theory, validates the bufferless sWDM scheme.

The remainder of this paper is organized as follows. We describe the research background and sWDM in Section 2. In Section 3, we define the slot assignment problem. We show in Section 4 that the round-trip-integer network topology is the necessary and sufficient condition that validates sWDM. We provide simulation results in Section 5. Section 6 presents the conclusion.

¹ Limited buffer may be used to improve the performance.

Table 1: A comparison between four all-optical switching paradigms

Optical Switching (paradigm)	Bandwidth reservation	Loss rate	Call blocking rate	Optical buffer	Wavelength conversion	Challenging issue
OPS	nil	moderate	nil	required	necessary	optical memory
OBS	link-by-link	high	nil	not required	necessary	high loss rate
λ -switching	end-to-end	nil	high	not required	not necessary	coarse bandwidth
sWDM	end-to-end	nil	low	not required	not necessary	slot alignment

2. Research background

With reference to Figure 1, there are two types of network nodes, edge routers and optical cross connects (OXC) in the all-optical λ -switched network. The edge routers act as an interface between the electrical domain and the optical domain, while the OXC act merely as wavelength-circuit switches without buffering capability. Under the wavelength continuity constraint, a dedicated wavelength λ must be assigned to the lightpath from ingress router s to egress router d throughout the links it traverses as shown in Figure 1.

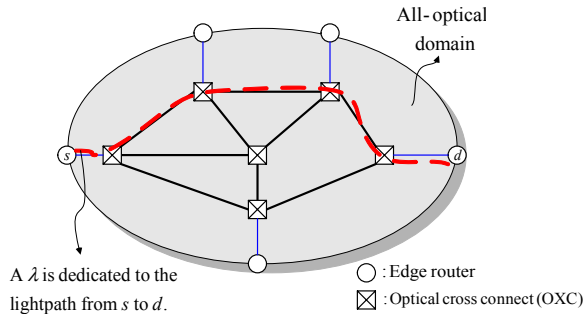


Figure 1. An all-optical λ -switched network.

In sWDM, TDM is incorporated into WDM as follows. Time is divided into frames, each containing F time slots in such a way that F connections can take turns in a round-robin fashion to transmit on a particular wavelength. Each admitted connection is assigned a wavelength and a time slot dedicated to it in each frame. This is illustrated in Figure 2. Note that the bandwidth granularity in this network is $1/F$ λ -bandwidth. For example, if $F=1000$ for 10Gbps λ -bandwidth, then the bandwidth granule is 10Mbps.

To perform sWDM switching at an OXC without buffering, all data slots must arrive just at the moment the OXC is to start a new slot period. Moreover, the routing pattern of the OXC in each time slot must also be set up prior to data arrivals. Therefore, the control issue at an OXC can be decomposed into two sub-problems. (i) How

do we align the arriving slots? (ii) How do we determine the switching state?

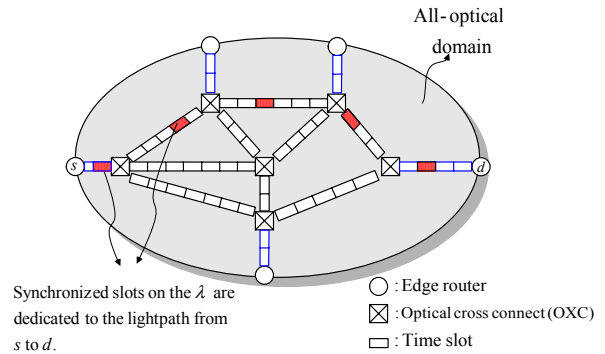


Figure 2. A slotted WDM network ($F=4$).

For question (i), a simple solution is to align all OXC's clocks so that they can advance from one slot to another at the same moment, and then cut fibers carefully so that the transmission time between any two OXC is an integer multiple of the slot time. This is called the one-way-integer scenario. We show later that the one-way-integer scenario can be relaxed with the round-trip-integer network topology. Some small amount of slot misalignment is inevitable nonetheless. This is due to the error caused by such things as inaccurate fiber cutting and imperfect clock boundaries synchronization. However, the minor slot misalignments (advance or lagged) can be offset by the offset periods (1 or 2) associated with each time slot as shown in Figure 3. As a quick peek, a fiber cutting error of 20m requires the offset period needed to be $20 / (2 \times 10^8) \text{ s} = 0.1 \mu\text{s}$, or 1% of overhead if the slot time is set to be 10 μs .

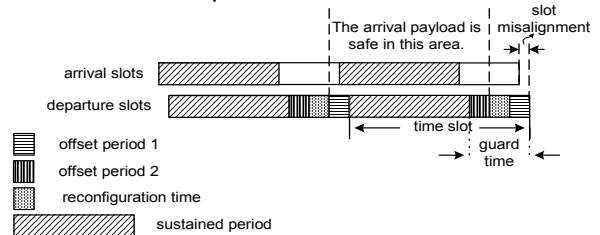


Figure 3. Guard time and sustained period.

In addition to the offset periods, a reconfiguration time is also needed for optical switching devices to reconfigure the switching state from slot to slot. The offset periods and the reconfiguration time constitute the guard time in a time slot as shown in Figure 3, where the remaining part is the sustained period that carries the normal payload.

One way of excluding reconfiguration time from guard time is to use the parallel-plane architecture in each OXC. For example, suppose that there are two parallel switching planes in each OXC such that when one plane is performing switching, the other can be reconfiguring its switching state by means of the whole slot time. In this case, the slot time can be reduced to be equal to the reconfiguration time. By the same token, with K parallel switching planes it is possible to push the slot time down to $1/(K-1)$ of the reconfiguration time.

In response to question (ii), each OXC must set up a predetermined periodic routing schedule such that the switching state of the OXC changes from slot to slot based solely on the schedule. To do this, a routing, wavelength and slot assignment (RWSA) algorithm is needed to assign a unique combination of resources, which includes a route, a wavelength and a slot, to each connection request during the call admission phase. Once the connection is admitted, all OXCs involved in the assigned route must update their switching schedules accordingly.

3. Slot assignment

Consider an sWDM core network with N OXCs, M links, W wavelengths on each link, and F time slots per frame per wavelength. An example is given in Figure 4, in which $N=6$, $M=8$, $W=4$ and $F=4$.

We consider the unit of time to be one time slot in the remainder of this paper. To label slots, we must define a clock system for each node. For $0 \leq i \leq N-1$, let $local_i(t)$ denote the local time in node i at real time t such that,

$$local_i(t) = t + TR_i - \left\lfloor \frac{t + TR_i}{F} \right\rfloor \times F \quad (1)$$

where TR_i is the time reference at node i , and $\lfloor a \rfloor$ is the greatest integer that smaller than or equal to a . The local time is a piecewise linear (with slope equal to one) periodic function. An analogy of this is the 24-hour day time system, which is periodic and may have different time references with respect to Greenwich time in different locations. Note that, TR_i can be any real value, but for any time reference $TR_i = c \cdot F + \tau$, where c is an integer and τ is a real number, it results in the same local time system as $TR_i = \tau$ does. For example, $TR_i = 1$ and $TR_i = -3$ result in the same local time system at node i , considering $F=4$.

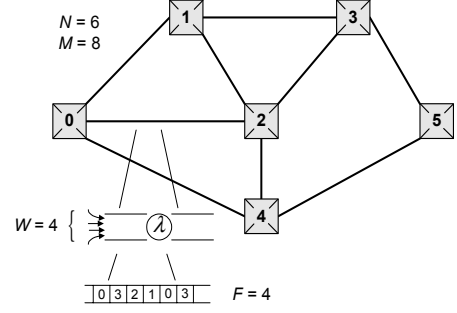


Figure 4. An sWDM core network.

At node i , the switching state of the OXC is allowed to change only when $local_i(t)$ is an integer (o'clock time), and the state will last at least for a time slot. We label the slots departing from node i on any outgoing links at time t as (i, x) , where $x = \lfloor local_i(t) \rfloor$. This is illustrated in Figure 5, considering again $F=4$. Note that we use $t = \tau^+$ to denote the time immediately after $t = \tau$. Consider $TR_0=1$. At $t = 25^+$, the slots departing from node 0 is labeled by $(0, 2)$ because $local_0(25^+) = local_0(6 \times F + 1^+) = 1^+ + TR_0 = 2^+$. It is worth noting that such a labeling scheme is also periodic from frame to frame, analogous to the o'clock times, which are periodic from day to day.

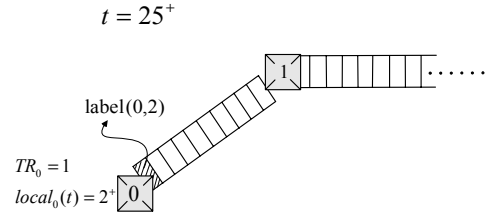


Figure 5. Local time and slot labeling.

At any time, any two nodes could have different local times due to different time references. We define the time difference of node j with respect to node i as

$$diff[i, j] = TR_j - TR_i \quad (2)$$

Suppose that $link(i, j)$ exists and the propagation delay from i to j over the link is denoted by $fly[i, j]$. Such a delay is referred to as the flying time from i to j .

With reference to Figure 6, when a slot from node i arrives at node j , its label is changed. We define the slot lag from i to j as

$$lag[i, j] = fly[i, j] + diff[i, j] \quad (3)$$

in such a way that the slot labeled by (i, x) on $link(i, j)$ is labeled by (j, y) at node j , where $y = (x + lag[i, j]) \bmod F$, as shown in Figure 6. In this example, we further consider $TR_1 = 2$, and hence $diff[0, 1] = TR_1 - TR_0 = 1$. In addition, suppose that $fly[0, 1] = 10$, then $lag[0, 1] = fly[0, 1] +$

$diff[0,1] = 11$. Thus, the slot with label (0, 2) receives a new label (1, 1) at node 1, because $(2 + 11) \bmod 4 = 1$.

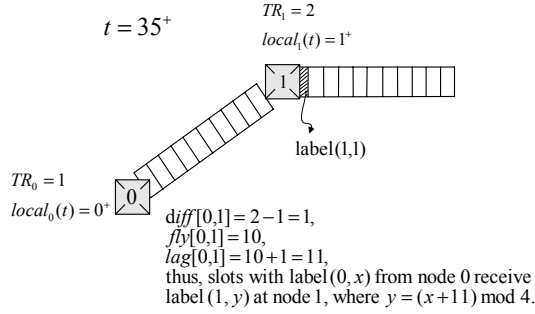


Figure 6. Label mapping.

An interesting analogy of the slot-mapping problem is that, suppose we want to fly from New York to Hong Kong, via Tokyo in transit without any waiting delay. We must first know the time difference and flying time between New York and Tokyo so that we can schedule an immediate flight from Tokyo to Hong Kong, in accordance with the local departure time at Tokyo. The similarity can be applied to the case of more than one via point as follows. Given a path, p_{ab} , from node a to node b , the slot lag over this path can then be defined as

$$lag[a,b] = \sum_{link(i,j) \in p_{ab}} lag[i,j] \quad (4)$$

such that slots with label (a, x) , traversing path p_{ab} from a to b , receive label (b, y) at node b , where $y = \left(x + lag[a,b] \right) \bmod F$.

An example, extended from Figure 6, is shown in Figure 7. In this example, we are further given $TR_3 = 0$ and $fly[1,3] = 11$. To find the lag from node 0 to node 3, along path p_{03} , we have $lag[1,3] = 11 + (TR_3 - TR_1) = 11 + 0 - 2 = 9$, and thus $lag[0,3] = lag[0,1] + lag[1,3] = 11 + 9 = 20$. As a result, the slot with label (0,2) receives a new label (3,2) at node 3 along the path, because $(2 + 20) \bmod 4 = 2$.

To define the slot assignment problem, we introduce an array of Boolean variables, $V[i,j,\lambda,x]$, for each node:

$$V[i,j,\lambda,x] = \begin{cases} 0, & \text{if slot } x \text{ of } \lambda \text{ on link } (i,j) \text{ is available;} \\ 1, & \text{otherwise.} \end{cases} \quad (5)$$

where $i, j \in \{\text{all edge routers}\} \cup \{\text{all OXCs}\}$. The problem of finding an available slot label for a new connection, from ingress edge router s to egress edge router d along a given route p_{sd} , is then equivalent to finding a set of λ and x in such a way that

$$\sum_{\substack{link(i,j) \in p_{sd} \\ p_{st} \subset p_{sd}}} V[i,j,\lambda,y_i] = 0, \quad (6)$$

where $y_i = (x + lag[s,i]) \bmod F$.

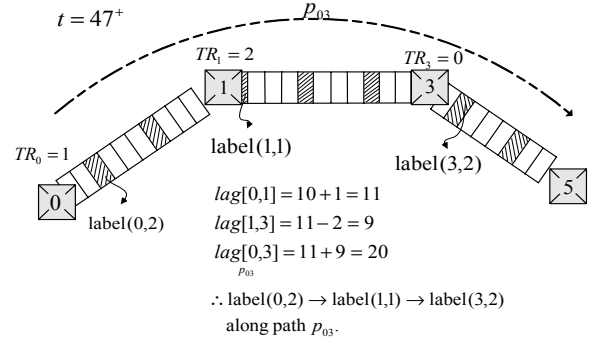


Figure 7. Slot labeling along a path.

If such a set of λ and x does not exist, then we simply try another route (i.e., another path) or block the call according to the RWSA algorithm. Note that the RWSA algorithm can be done locally at the ingress edge router. After the set of resources has been selected for the new connection, the ingress edge router will then send a reservation signal to all OXCs involved such that they can update their switching schedules accordingly. This reservation procedure is necessary not only for updating the switching schedules, but also as a contention resolution protocol in case some other edge routers request some common resources at the same time. Once the ingress edge router receives all confirmation acknowledgments, the route establishment is done and the connection can start sending data along this route.

4. Round-trip integer networks

So far, we have assumed that both $fly[i,j]$ and $diff[i,j]$ are integers for all link (i,j) in the network. However, it is not necessarily true as long as $lag[i,j]$ is an integer.

An example is given in Figure 8, which shows that with a careful assignment of the time references at nodes i and j , the non-integer flying time can be offset such that the slot lag is an integer. In this example, the flying time from i to j is 1.8 time slots, and the time difference of j with respect to i is $0.8 - 0.6 = 0.2$. Thus, the time lag from i to j is $1.8 + 0.2 = 2$, an integer. Figure 8 also shows that a slot that departs from i to j can be just "in time" to be switched at j , recalling that the switching state of a node can only be changed when the local time is an integer.

Given this set of time references, we must, at the same time, ensure that the time lag for the reverse path, i.e. $lag[j,i]$, is also an integer. Obviously, it can be made possible if the backward flying time is not necessarily the

same as the forward flying time along this link. An example is given in Figure 9.

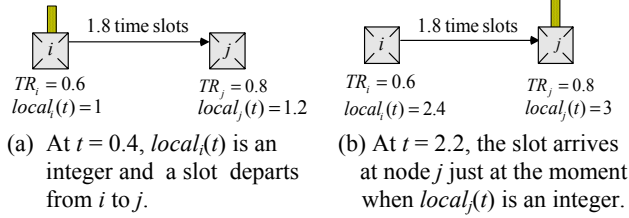


Figure 8. Non-integer flying time and integer lag.

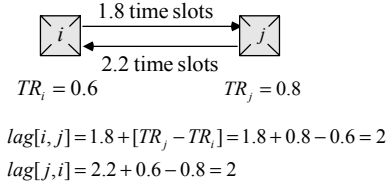


Figure 9. Different forward/backward flying times

We define that a network is a round-trip-integer network if and only if, starting from any node, traversing any loop in the network back to the node, the total flying time is an integer. Figure 10 gives an example of the round-trip-integer loop.

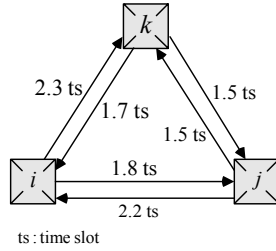


Figure 10. A round-trip-integer loop

Theorem: Given a graph $G=[V,E]$, representing a connected network, there exists a time-reference assignment for all nodes in V such that all lags in G are integers if and only if G is a round-trip-integer network.

We skip the proof here due to the constraint of the paper length. An interesting and useful property of round-trip-integer networks is observed as follows. We can arbitrarily select one of the nodes in a round-trip-integer network as the center and find a spanning tree for the center to broadcast clock signals. In this case, time references of all other nodes will be automatically defined by the first clock signal. This is illustrated in Figure 11, based on the network given in Figure 10. Suppose that node i is selected to be the center and the clock spanning tree is $i \rightarrow j \rightarrow k$. In the beginning, we set $TR_i = 0$. At $t = 0$, node i sends the first time slot, which also carries the clock signal, to node j . At $t = 1.8$, node j receives the

clock signal so it starts its first time slot. In this case, $local_j(1.8) = 0 = 1.8 + TR_j$ and hence $TR_j = -1.8$. Then, the clock signal is propagated from j to k and it reaches k at $t = 3.3$. Node k starts immediately its first time slot and hence $TR_k = -3.3$. This scheme is referred to as the clock signal broadcasting scheme. It avoids independently-defined difficultly-aligned local clock systems since all nodes only need to advance slots at the moments they receive clock signals.

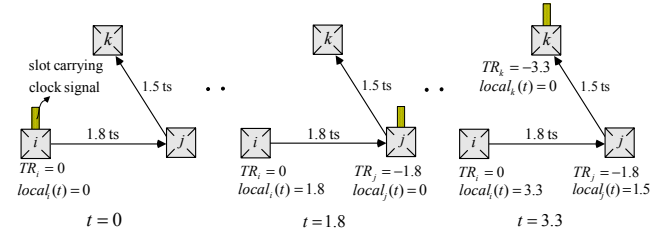


Figure 11. Clock signal determines time references

5. Performance study

In this section, we study through simulation the performance of call blocking rate of the following schemes: (i) and (ii) WDM with or without wavelength conversion; (iii), (iv), (v) and (vi) sWDM with or without wavelength conversion with zero or different lags.

The simulation is based on a mesh network topology as shown in Figure 12. We assume calls arrive at each edge router randomly and are equally likely to be destined for any other edge router. After being admitted, a call stays in the network for a finite amount of time before termination. In addition, we employ fixed-alternate routing and the first-fit wavelength-slot assignment algorithm [6] in the simulation. The input parameters to the simulation are: ρ , the load to each edge router in the network, which is equal to *arrival rate* \times *mean service time*; W , the number of wavelengths supported by a link; F , the number of calls that can be carried on a wavelength. Note that in sWDM, F is also the number of time slots.

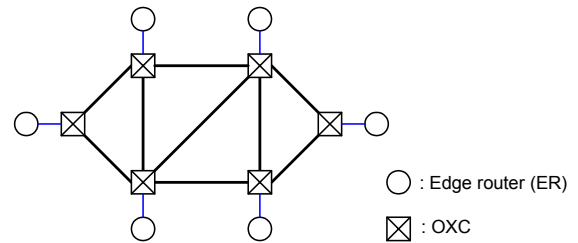


Figure 12. Network topology for simulation

Figure 13 gives the simulation results. As can be seen, the blocking rates of sWDM schemes are always comparable with the Erlang blocking probability. In other words, with fine granularity of bandwidth in sWDM,

even a simple RWSA algorithm can achieve a performance close to the optimum.

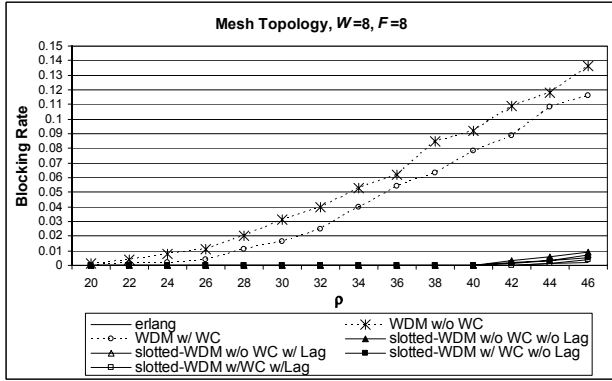


Figure 13. Blocking rate versus offered load

Link failure may occur from time to time. To restore the network functionality as quickly as possible, one way is to assign each connection request a primary path and a backup path for admission. In this case, the admitted connection transmits data with the primary path in the beginning and with the backup path only when any link on the primary path fails. However, if the network resources are assigned to primary paths and backup paths in a 1:1 ratio, the network utilization has a bound of 50%. To achieve better utilization, there are three constraints for the path-protected RWSA algorithm.

- i A primary path must have a disjoint backup path.
- ii A set of λ (wavelength label) and x (slot label) on a link can be assigned to a primary path in an unshared manner. Alternatively, it can be allocated in a shared manner to multiple backup paths.
- iii All backup paths that share the same set of λ and x on each link must have disjoint primary paths.

Figure 14 shows the simulation results for the mesh network given in Figure 12 with path-protection scenario. Again, the sWDM outperforms λ -switching.

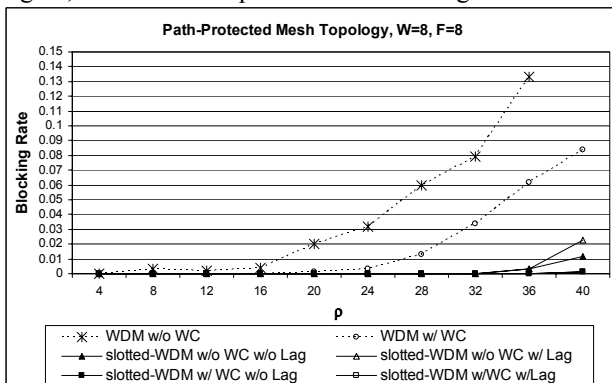


Figure 14. Blocking rate versus offered load with path protection

6. Conclusion

With the objective of breaking λ -bandwidth into smaller base bandwidth, we have studied the implementation and theoretical issues of the all-optical bufferless sWDM scheme. Slots in sWDM can be labeled by local times, and assigned for connection requests according to the slot lag on each link. Clock signaling, slot alignment and guard-time overhead are other challenging issues to sWDM. We have proposed a new clock signal broadcasting scheme and provided feasible solutions to slot misalignment and guard-time overhead problems. Through simulation, we have shown that given a blocking rate requirement, sWDM can achieve much higher network utilization in comparison with λ -switching.

Acknowledgement

The authors like to thank Mr. Jianfeng Xu for providing the simulation results in this paper.

7. Reference

- [1] T. S. El-Bawab and J. D. Shin, "Optical packet switching in core networks: between vision and reality," *IEEE Communications Magazine*, vol. 40, no. 9, pp. 60-65, September 2002.
- [2] J. Turner, "Terabit burst switching," *J. High Speed Networks*, 1999.
- [3] C. Qiao and M. Yoo, "Optical burst switching (OBS) – A new paradigm for an optical Internet," *J. High Speed Networks*, vol. 8, no. 1, pp. 69-84, 1999.
- [4] I. Baldine, G. N. Rouskas, H. G. Perros, and D. Stevenson, "Jumpstart: a just-in-time signaling architecture for WDM burst-switched networks," *IEEE Communications Magazine*, vol. 40, no. 2, pp. 82-89, February 2002.
- [5] B. Mukherjee, "WDM optical communication networks: progress and challenges," *IEEE J. Select. Areas Commun.*, vol. 18, no. 10, pp 1810-1824, October 2000.
- [6] H. Zang, J. P. Jue, and B Mukherjee, "A review of routing and wavelength assignment approaches for wavelength optical WDM networks," *Optical Networks Magazine*, vol. 1, no. 1, pp. 111-124, January 2000.
- [7] N. Huang, G. Liaw, and C. Wang, "A novel all-optical transport network with time-shared wavelength channels," *IEEE J. Select. Areas Commun.*, vol. 18, no. 10, pp. 1863-1875, October 2000.
- [8] B. Wen, K. M. Sivalingam, "Routing, wavelength and time-slot assignment in time division multiplexed wavelength-routed optical WDM networks," in *Proc. IEEE INFOCOM 2002*, New York, April 2002.