Performance Comparison of OTDM and OBS Scheduling for Agile All-Photonic Network

Xiao Liu, Anton Vinokurov, Lorne G. Mason

Department of Electrical and Computer Engineering,
McGill University,
Montreal, Quebec, Canada
{xiaol, avinok, mason}@tsp.ece.mcgill.ca

ABSTRACT: *The Agile All Photonic Network (AAPN) is a high speed transparent agile optical transport network. A key issue in these all photonic networks is the distributed scheduling method employed at the edge nodes and the manner of co-ordination with the core switching state. In this paper we compare several classes of resource sharing methods. Optical Burst Switching (OBS) and two classes of Optical Time Division Multiplexing (OTDM) referred to as statistical Slot by Slot and deterministic frame based OTDM are investigated. Performance results obtained by simulation show that the OTDM schemes compare favourably with OBS in terms of packet loss and bandwidth utilization while keeping packet delay sufficiently low to meet real time QoS requirements. Slot by Slot scheduling is suitable for MANs while frame based TDM schemes with signalling is appropriate for either WAN or MAN applications.*

KEYWORDS: *Optical Burst Switching, OTDM, Scheduling*

1. Introduction

The anticipated availability of sub micro second photonic switching devices will enable rapid, on demand dynamic bandwidth allocation. To efficiently transport bursty traffic such as found in the Internet, fast optical switching is required to time share light paths. In addition the reduced granularity of the data volume carried in a time slot, packet or burst increases the potential reach of all optical networks to smaller aggregation points closer to the traffic demand sources. Fast photonic switching will also improve the networks' robustness to unpredicted variations in traffic demand. The Agile All-Photonic Network (AAPN) project (Mason *et al.*, 2005; Bochmann *et al.*, 2004) addresses the issue of designing networks to exploit the capability of sub micro second photonic switching devices.

The AAPN can be viewed as a distributed switch comprised of edge nodes, where the optical electronic conversion takes place, connected in an overlaid star topology to photonic core crossbar switches employing sub microsecond photonic switching. The overlaid star topology proposed for the AAPN, facilitates the introduction of various approaches to time sharing link capacity, including Optical Burst Switching (OBS) as well as several Optical Time Division Multiplexing (OTDM) techniques. These alternatives differ in regard to the degree of co-ordination in resource allocation between the edge nodes and the core crossbar photonic switches. Techniques that have been proposed in the literature include just-in-time signalling architectures for WDM burst-switched networks – JumpStart (Zaim *et al.*, 2003; Baldine *et al.*, 2003), optical burst switching (Xu *et al.*, 2001), slot-by-slot routing (Maach *et al.*, 2002; Zang *et al.*, 1999), and concepts introduced in this paper are emerging as candidate approaches.

Taking a broad perspective, a spectrum of alternatives can be envisaged as follows. Asynchronous *Optical Burst Switching* minimizes the co-ordination requirements as it requires neither synchronization nor signaling for time slot reservation. Synchronous slotted OBS version without signaling for slot reservation provides another example where improved traffic handling performance can be anticipated through global synchronization, analogous to that gained in slotted Aloha over pure Aloha systems.

Synchronization which is feasible for overlaid star topologies also enables the application of various OTDM techniques, ranging form *frame based deterministic scheduling* where edge node transmissions are appropriately clocked to the core switch configuration by means of a round robin schedule for example. By suitably allowing for the propagation delay, slots from different end points arrive at the core crossbar switch and are switched to their appropriate destinations without output port collisions. Several variations of such deterministic frame based schemes are feasible by adding additional signaling between the edge and core switches, to make the schedule dependent on the traffic demand.

A third class of schedulers which we study here are dubbed *statistical Slot by Slot scheduling*. In this case the time slots at the core switch output ports are explicitly reserved based on signaling requests from the edge switches driven by the traffic present in the edge node. From the definition of AAPN, no buffering can be used at the core optical switch as we seek a transparent network and delay lines are unattractive. An important issue is how to make fast scheduling decisions that will efficiently use the optical core. Several proposals have been reported on fast scheduling in the literature (McKeown, 1999; Shreedhar *et al.*, 1996; Smiljanic, 2002; Bianco *et al.*, 2003; Kar *et al.*, 2003), however to our knowledge, only one paper has addressed the problem in the presence of propagation delay, in the context of high speed optical router design where line cards corresponding to different input ports may be in different equipment racks. In (Minkenberg, 2003), a stateful protocol is proposed in which the scheduler maintains the state of Virtual Output Queues.

We consider employing a similar input queuing approach in AAPN MAN and WAN applications where propagation delays are significant and heterogeneous as the input queues are co located with edge switches while the switching occurs within the optical core switch which may be a considerable distance away. We propose and evaluate a simple variation of Probabilistic Iterative Matching (PIM) which we call the “adapted PIM algorithm”, which avoids repeated request reservations from the edge node while guaranteeing timeslot delivery.

The remainder of this paper is organized as follows. Section 2 provides a description of the system we model for studying the scheduling problems. Section 3 provides a detailed description of the scheduling algorithms considered, including Optical Burst Scheduler, and two classes of Optical Time Division Multiplexing Schedulers, corresponding to the deterministic framed Round-Robin scheme, and the statistical slot by slot scheduling using the adapted PIM matching algorithm. In Section 4 we show the effect of different traffic demand matrices and network topologies for these alternative scheduling methods using the OPNET discrete event simulator, and discuss the results. We conclude and discuss ongoing research in Section 5.

2. Configuration studied

Stars and composite stars (Figure 1) are robust to various traffic distributions. The topology considered has a significant influence on the implementation complexity of the bandwidth management mechanisms. For example Optical Time Division Multiplexing (OTDM) requires network synchronization while Optical Burst Switching (OBS) does not. OBS can operate in a general class of topologies such as meshes, trees, rings, stars etc. However, network synchronization can be more easily realized for tree network topologies. By tree topology we mean the links connecting to a particular core switch and the edge node Virtual Output Queues (VOQ) buffers served by that core switch form a tree. Star and star-star topologies are special cases of

the tree topologies. In order to eliminate the Head of Line (HOL) blocking at the edge node input queue, we employ VOQs with each VOQ storing the packets destined for the same egress edge node.

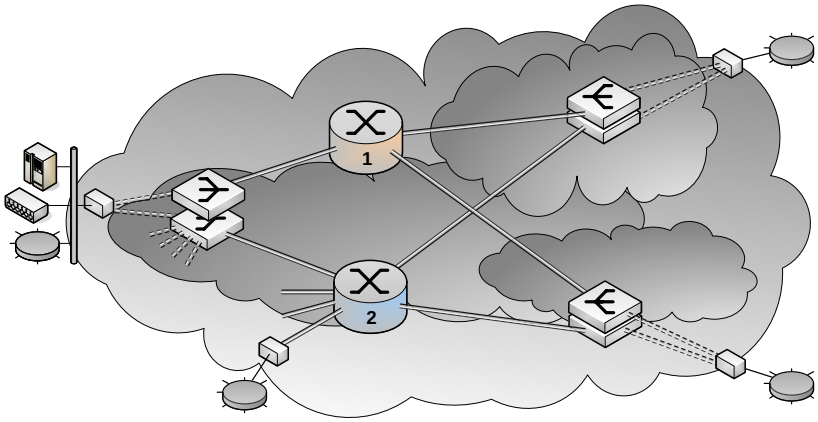


Figure 1. *Composite star network.*

Synchronization of overlaid stars (which is a superposition of physically independent star networks) can be realized by partitioning the virtual output queues (VOQ) in the origination edge nodes according to the core switches which serves them. This effectively decouples the synchronization of different core switches and their subtending edge node buffers making the synchronization manageable. The set of VOQs for the different destination edge nodes, located at each ingress edge node are partitioned into J groups, according to the core switch through which the traffic is routed. For the case of deterministic shortest path routing, where only a single path is used for a given origin-destination (O-D) pair, a single VOQ for a specific destination edge node is needed at the origin edge node or ingress. For the more general case of load sharing where traffic is carried on P disjoint paths linking a given O-D pair, then P copies of VOQ buffers are required per edge node for each such O-D pair.

Due to the decomposition discussed above, without loss of generality, we consider an optical star network comprised of a non-blocking cross-bar core switch and several edge nodes, which are able to receive and transmit simultaneously. In addition, a source node is attached to each ingress edge node to generate traffic flows for that ingress edge node. Figure 2 illustrates the network structure.

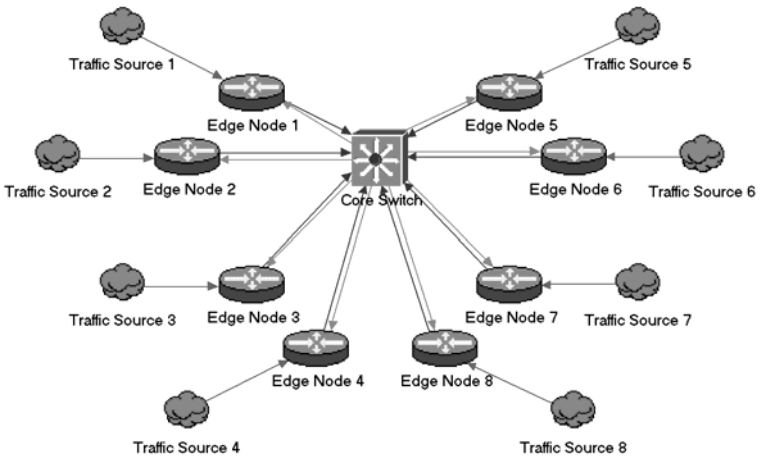


Figure 2. *Network structure (8 edge nodes)*

3. Scheduling algorithms

3.1. Optical Burst Switching Algorithm

Source packets are dispatched onto Virtual Output Queues according to a uniform destination distribution. Queues are server in a round robin fashion; there is a minimum queue threshold as well as finite queue size defined. If a queue is selected to be served, a burst is formed (variable length, but smaller than maximum defined size). A configuration request is sent to the Core Node, followed by the data burst itself (after an offset delay which is *scheduling decision + switching time*). The Core node processes the incoming request and switches incoming ports to its output ports on “first come, first served” basis. If an output port collision occurs, the later burst is lost.

For each value of offered load (amount of traffic generated by the source to the network) a discrete-event simulation is performed. To collect reliable statistics more than 1 million packets were forwarded by each edge. Edge node and core node blocking was tracked as well as end-to-end packet delay (due to queue starvation and signal propagation delay). The number of edge nodes in the simulations varied from 2 (pure full-duplex communication) to 32. No significant difference in packet loss was found compared to 16-edge network.

3.2. Optical TDM scheduling

When designing TDM schedule algorithm, we adopted a *slot scheduling* technique, which is first described in (Liew *et al.*, 2003; Ramamirtham *et al.*, 2003).They actually did slot mapping and allocation without using any buffering in the TDM network. In our system, we propose a scheduling scheme namely *slot-by-slot* scheduling, in which slots from distinct sources are scheduled in a coordinated manner. The configuration of the core switch is computed once for each time-slot. We assume that the matching algorithm is fast enough to find the match within one timeslot. When there are conflicts in the requests, for example, two ingress edge nodes have traffic going to the same egress edge node; one of them is selected for transmission while the others are blocked at the edge nodes until they are granted access by the core.

Because transmission at each edge node is carried out on a time slot basis, a major challenge for global synchronization arises. Since no delay line can be applied in AAPN architecture, we manage the timing by delaying transmission at the edge node side to accommodate the propagation delay between it and the core switch. In addition, to account for the switch reconfiguration time, we apply a narrow guard time that separates slot boundaries. Following is a description of the slot-by-slot scheduling method. At each time-slot:

a) Every edge node sends a request to the core switch; the request contains information on whether a specific VOQ has traffic to send or not.

b) A central electronic controller co-located with optical core node sends grants to edge nodes. It will run the matching algorithm to determine which VOQs are to be served in a given future time slot, and configures the cross-bar optical switching matrix at the appropriate future time to transparently switch the arriving traffic slots. After the schedule is computed at the core, *grants* are sent to those edge nodes that are allocated the future time-slot in question. The grant indicates which VOQ(s) in an edge node will be served and the *sending time* for edge node to transmit data.

c) The Edge nodes transmit data to core switch via the optical data channel.

In the following part, we present the scheduling algorithms we developed for our system based on the slot by slot scheme. First, we introduce the round robin allocation, which is a simple form of frame-by-frame scheduling, there is no signalling phase, and the switch configuration is made in the central scheduler in a deterministic round robin manner. After that, we describe the PIM algorithm, and our proposed modification called the adapted PIM algorithm.

3.3. Round Robin Allocation

Round robin allocation is a very simple and obvious allocation method. Consider for example an 8-edge network: every ingress edge node will have traffic bound to one of the other 7 egress edge nodes periodically. Each VOQ buffer will be served by one slot in every 7 time-slot frame. Fixed Round robin allocation is very simple to implement, as no explicit signalling is required. The transmissions are clocked to arrive at the core matrix when it is set to route those simultaneous incoming arrivals to the correct output port.

A short coming for this algorithm is that the allocation is fixed and deterministic which might lead to a high inefficiency under unbalanced or bursty or variable traffic conditions. Several approaches to improve the performance of the frame based deterministic round robin scheme are possible such as those employed in conventional TDM switching where slots are allocated to connections at call set up via signalling and call processing. Alternatively one could allow the allocation to be data dependent in that the deterministic schedule would be updated to be best matched to measured or predicted traffic demand. We have at this point implemented the fixed deterministic scheme in the OPNET simulator and report here on its performance. The other signalling enhanced frame by frame methods alluded to are currently being implemented and will be reported in the future.

3.4 PIM: Parallel Iterative Matching

PIM is a basic matching algorithm which is derived from Round-Robin Matching, both the selection on the input and output sides are made randomly, and more than one iteration can be performed. Each iteration of the algorithm consists of the following these steps:

1) Request: Each unmatched input sends a request to every output for which it has queued slots.

2) Grant: If an unmatched output receives any requests, it grants to one of them by randomly selecting a request uniformly over all requests.

3) Accept: If an input receives a grant, it accepts one by selecting an output randomly among those that granted to that input.

NOTE. — During each iteration, more matchings may be added to the previous one.

3.5 Adapted PIM

Adapted PIM scheduling is a realization of PIM in our system. It combines the matching procedures described in PIM, and also accounts for propagation delay on

the transmission links from the edge node to the core switch. In the PIM algorithm, both input and output ports are selected randomly, without regard to traffic load. This may lead to unfair resource allocation because a VOQ with a light traffic load may get access and block another VOQ with heavy traffic load. To deal with this, we design a fair request mechanism where each request is made for the same amount of traffic.

In each VOQ, packet monitoring is applied. We use the *request bound* to determine the threshold for sending a request. If the queued packets reach the threshold, a request is made for this VOQ. Owing to the request pipelining, more requests may have been issued than the traffic in this VOQ, so it might happen that a grant is scheduled for an empty VOQ. To avoid this problem in our design, when a request is made for some packets, these packets are marked, and no more requests will be sent for them in the future.

However, under this design, another problem arises. If one request is not granted in central controller, then the packets that made this request will not be served in the future. Our solution to this problem is to keep a list of ungranted requests in the central controller in order to deal with them in later time-slots. When computing schedules, those ungranted requests are treated with higher priority over the newly arriving requests, and the matching procedure follows the 3-steps described in the PIM algorithm. At the end of the matching, the ungranted requests are placed in a list, and the longer the request stays in the list, the higher the priority it has to be treated. In this way, each request is guaranteed to be granted in the future. Experimentally we observe from the matching results, that a grant is usually delayed by 0~7 time-slots after the initial request is read by the central controller. Hence, the total delay of a grant to a request is:

[propagation delay (request)] + [grant delay] + [propagation delay (grant)].

Assuming a 50-time-slots propagation, the packets in the VOQ are served after 107 time-slots.

Following is a description of the algorithm: Consider an *NxN* central crossbar switch, where each edge node $i,j \in \{1,\ldots,N\}$, has *N-1* Virtual Output Queues, corresponding to the each of the other egress edge nodes. In each time slot, a packet can be transmitted from any chosen VOQ. The input of the algorithm is the status of all VOQs (empty/ nonempty, central controller read the status from request shown in Figure 3). The output of algorithm is a schedule, which can be interpreted as a set

Match = { (i,j) | packet will be sent from edge node i to edge node j }

In any time slot, an ingress edge node can only transmit one packet, and an egress edge node can receive only one packet. A list for left-over requests from previous time slots or a set *unMatch = { (i,j) | packet want be sent from edge node i to edge node j }* is kept in the central scheduler and a set for new arriving requests *R={(i, j)| packet want to be sent from edge node i to edge node j}.* The schedule for one time slot in the future is determined as follows:

Iteration 1:

Step 1: Update R(i , j); i=0; j=0;i=o={1,…,N}; Match{i,j}=0

Step 2: Ingress edge node i check unMatch(i, j), if any, choose egress edge node j with highest entry, put (i, j) into S(i, j). If it has all the same entries, choose j randomly; if there is no entry, choose j from R(i, j) randomly.

Step 3: i=i+1; if i<N, go to step 3.

Step 4: Egress edge node j choose an ingress edge node from S(i,j). Fix the matching pair (i,j), remove (i, j) from unMatch(i, j) or R(i,j), put (i,j) into Match(i,j).

Step 5: j=j+1, if j<N, go to step 4.

Step 6: Put unmatched R(i, j) into unMatch(i, j). If it already exists in the list, then increase the entry by 1.

Iterations 2~N:

Step 1: Check if ingress edge node i has already been matched from previous iteration, if not, choose an egress edge node from unMatch(i, j), put into S(i, j).

Step 2: i=i+1; if i<N, go to step 1.

Step 3: Check if egress edge node j has already been matched from previous iteration, if not, choose an ingress edge node from S(i, j), put into Match(i, j), remove (i, j) from unMatch(i, j).

Step 4: j=j+1, if j<N, go to step 3.

Here is a summary of this adapted PIM scheduling:

— Request bound: is packet number bound for sending request;

— VOQ monitor: packets that have request sent will be marked, and no more requests will be sent for them. This completely eliminates the problem of wasted grant.

— Schedule computation: non-granted requests are stored and be matched in later timeslots. Accumulated requests from the same VOQ will be assigned higher priority in the bipartite matching. In this way a guaranteed grant will be made to a request, but the grant might be delayed.

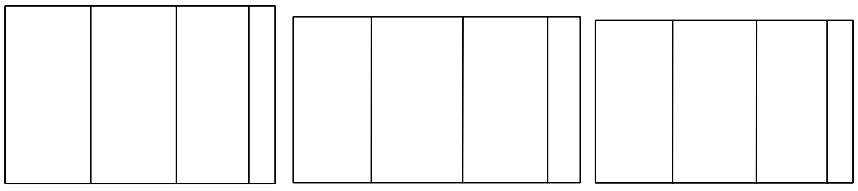


Table 1. *Demonstration of request and schedule*

Instead of changing the algorithm, we modified the protocol between ingress edge node and central scheduler, which provide us with precise traffic demand monitor and efficient slot allocation. However, a grant to request might be delayed at the central scheduler. In our experiment with 8-edge node network, a grant is usually delayed for 0~7 timeslots which is still small compared to propagation delay.

It has been shown that the PIM algorithm finds a *maximal* matching after $log_2N+3/4$ iterations on average (Anderson *et al.*, 1993). The PIM algorithm is designed to find a maximal match, so that its link utilization can not be as good as *maximum* match. In the worst case scenario, the number of pairings in a maximal match can be as small as 50% of the number of pairings in a maximum match (McKeown, 1999). However, PIM algorithm yields a logarithmic time $O(logN)$ computation, which is much better than $O(N^2)$ time in maximum matching. The PIM algorithm also avoids the potential starvation of VOQ service inherent in the maximum match algorithm.

4. Comparison and analysis

4.1 Modeling

We compare the performances of these scheduling algorithms by means of network simulation. The AAPN scheduling framework was implemented in the OPNET Modeler 10.5 discrete-event simulator software. The network model consists of following objects:

— Traffic source, which generate data packets with specified inter-arrival time (Poisson process) and variable size (uniform distribution). Sources are representing the legacy part of network sending data to the AAPN core at rates up to 10 Gbit/s.

— Edge node, which takes incoming data stream from source and sends packets to network. Destination edge node is chosen by a random process (uniform and weighted uniform distribution). Edge node implements the client part of scheduling process.

— Packet streams providing packet delivery from edge node to core node and back, with a given propagation delay.

— Core node, which implements server portion of scheduling process as well as switched incoming packets according to computed schedule.

An edge node has a number of Virtual Output Queues (VOQ) by number of destination edge nodes. The VOQ size is fixed, in case of queue overflow due to service starvation, newly arrived packets from source are dropped and the queue loss counter is increased.

In case of output port collision in the core node (burst mode), data is also lost, and the corresponding counters are also updated.

The number of edge nodes considered ranges from 2 to 32. Propagation delays were generated randomly for both the Metro Network (average distance is 10 km) and the Wide Area Network (average distance is 1000 km.). Optical Burst Switching model and various Optical Time Division Multiplexing models (discussed below) were implemented in a common way and share the same set of simulation parameters. We collect statistics to determine the performance characteristics, in particular *packet loss* and *utilization* of bandwidth, and *end to end delay*. Following are the default parameters used in the simulations:

Time slot (τ): 10 μsec=10^{-9} sec.

Request bound: 70~80 packets.

Bandwidth of link (capacity): 10 Gbps.

Arrival rate of uniform Poisson traffic: λ=0~10.000.000 packets/sec.

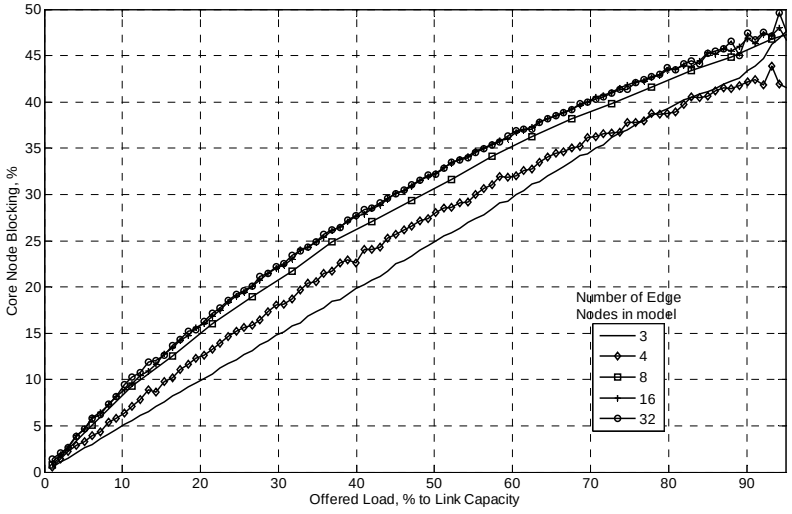
Mean packet size: 1000 bits.

Slot size: we define the slot size by multiplying the bandwidth of link and the duration one time-slot, which is: 10^{-5} x 10^{10}=10^{5} bits.

Topology: Metropolitan Area Network (MAN); Wide Area Network (WAN).

4.2. Effect of network size

4.2.1. Burst Switching Scheduling



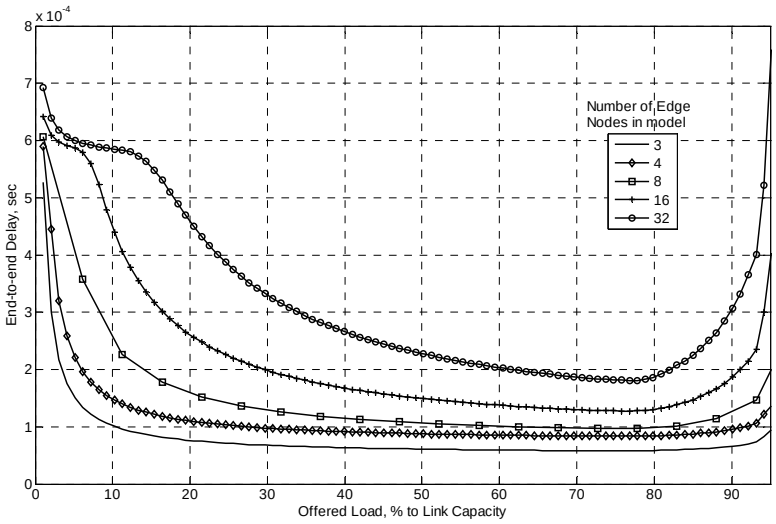


Figure 3. *Burst switching scheduling*

It is clear that for pure OBS scheduler the blocking probability is unacceptably high even for low utilization values (Figure 3). While scheduler performance can be slightly improved by introducing a queue service threshold, minimum and maximum burst sizes, this effect is negligible. Some research has been reported to extend the OBS model with some kind of "reservation" or retransmission schemes, but this requires a signalling channel to exist between core node back to edge node. This, in turn, increases latency and jitter (at least doubles!), causes out-or-order burst arrival at destination (may be corrected by buffering for the price of additional delay), and makes the scheduler more complex. Other options are fiber delay lines, deflection routing or even wavelength conversion for collided bursts, all of which did not conform to the AAPN design requirements. One possible extension to the OBS algorithm is to replace the "unpredictable" sending of data bursts with some coordinated process. This process should depend on a centralized decision engine, running at network core node and aware of all traffic demands for all network endpoints.

4.2.2. Metropolitan Area Network (MAN)

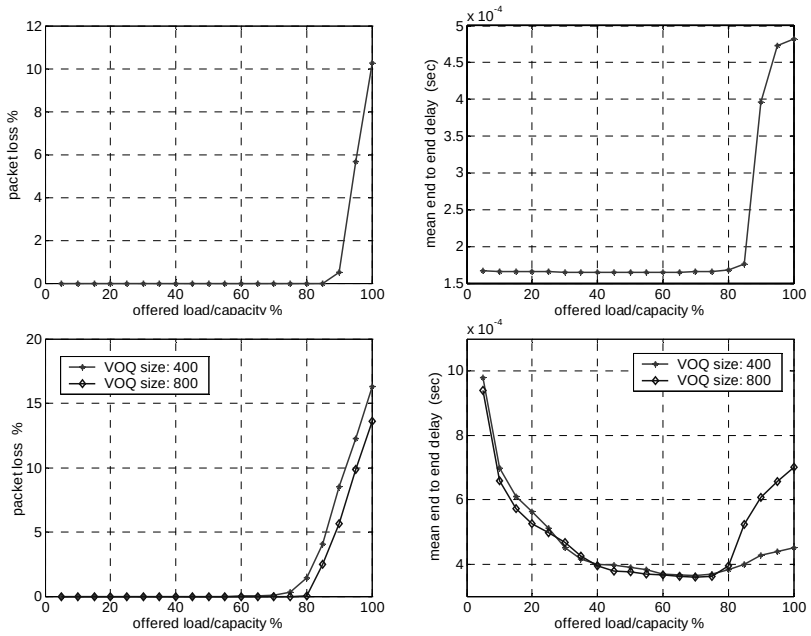


Figure 4. *Round robin allocation (top) and adapted PIM with MAN topology (8 edge nodes)*

In the round robin allocation the packet loss is very low when the offered load is lower than 82% capacity. We notice that, mean end to end delay for adapted PIM is longer than round robin. This occurs because for an 8-edge node network, every VOQ is served on a periodic basis under the round robin allocation, every 7τ where τ is the slot duration.

It is worthwhile mentioning that in the adapted PIM simulations, the performance of packet loss varies with the *"request bound"* design parameter. We currently set it to 80 packets, which is close to the slot size.

In the figure of end to end delay for adapted PIM, we notice that the delay is longer when offered load is low, this is because it takes longer time to fill up the "request bound" when packets arrive with lower rate.

4.2.3. Wide Area Network (WAN)

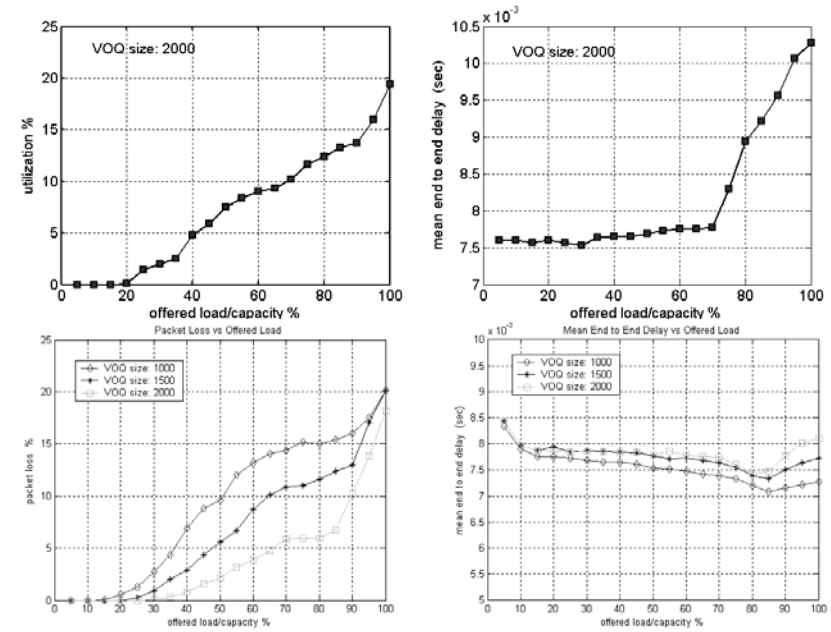


Figure 5. *Round robin (top) and adapted PIM with WAN topology*

Figure 5 shows the algorithms' performance for the WAN topology. We can observe performance degradation in both packet loss and utilization for these larger edge node buffer sizes. In a WAN network structure, more buffering is required at each edge node to obtain corresponding buffer overflow probability than is the case for the MAN network. We performed simulations with the VOQ sizes of 1000, 1500 and 2000 packets, and the results as expected show a significant impact of network propagation delay on packet loss performance as well as the mean end-to-end delay. By increasing the buffer size we can reduce the packet loss however this comes at the expense of greater delay.

4.2.4. Analysis

We studied the effect of network propagation delay on different algorithms, and the simulation results show that burst switching scheduling is the least affected algorithm as expected because the there is now reservation delay. PIM has the most performance sensitive to when the propagation delay as a minimum of a round trip time is required for the reservation and granting process. Consider the WAN topology for example, if an edge node is 600τ away from core node, the stored packets in VOQ should at least wait for 1200τ until it is sent. Consequently with the PIM scheduling approach, very large VOQs are required with WAN topology. Hence, slot-by-slot design is not suitable for WAN topologies.

4.3. Effect of traffic pattern

4.3.1. Traffic Model

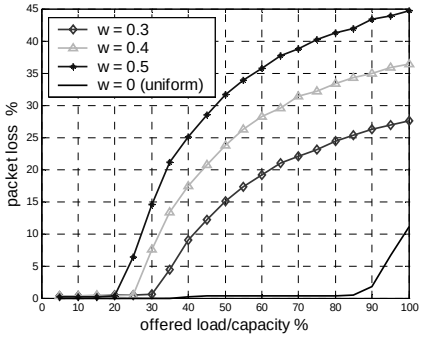
We evaluated the performance of the scheduling algorithms under a uniform Poisson traffic demand matrix and non-uniform Poisson traffic demand matrix in this section. By uniform we mean that the destination of arriving traffic to each ingress edge node is uniformly distributed over the other destination nodes. For the non-uniform case, we adopt a traffic model, which is characterized as follow:

$$\lambda_{ij} = \begin{Bmatrix} 0, & if\ i = j; \\ \lambda\left(w + \dfrac{1-w}{N-1}\right), & if\ j = (i+1) \bmod N; \\ \lambda\dfrac{1-w}{N-1} & otherwise \end{Bmatrix}$$

where λij represents the traffic intensity from input i to output j; w is the non-uniform coefficient, whose value changes from 0~1; N is the number of edge nodes. Note that the normalized offered load for each ingress edge node and egress edge equals to unity. Note that the offered load to every ingress edge node and egress edge node is:

$$\lambda_i = \sum_{j=0}^{N-1} \lambda_{ij} = \lambda\left[w + (N-1)\frac{1-w}{N-1}\right] = \lambda = \sum_{i=0}^{N-1} \lambda_{ij} = \lambda_j$$

4.3.2. Simulation Results



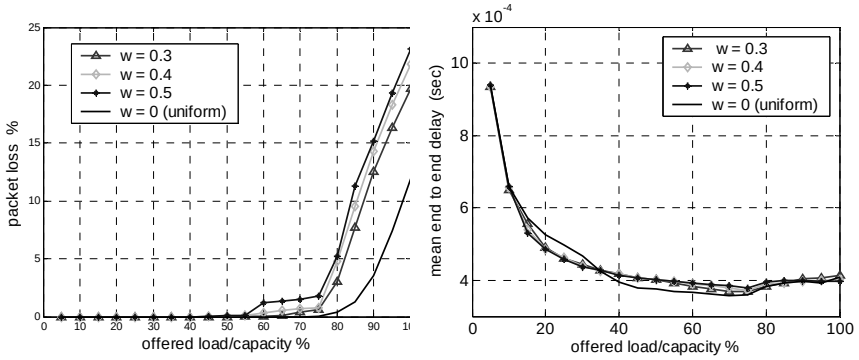


Figure 6. *Packet loss vs. offered load for round robin(top) , packet loss vs. offered load for adapted PIM (left), and end to end delay for adapted PIM (right)*

In Figure 6, we vary the value of w from 0.3 to 0.5 to get un-balanced traffic. It is clear that round robin allocation becomes less efficient as the traffic imbalance factor, w, grows. On the other hand, the negative performance effect of non-uniform traffic is not so pronounced in the adapted PIM scheduling method and packet loss remains as low as 2% even when the offered load reaches 75% of link capacity.

4.3.3. Analysis

As expected the adapted PIM algorithm is robust to changes in the traffic demand while the fixed round robin scheme is not. This is because the former calculates the schedule according the traffic demand at every edge node, while the latter allocates timeslots in a fixed deterministic manner without regard to the actual instantaneous traffic demand.

5. Conclusion

The *OPNET Modeler* discrete-event simulator was employed to investigate the performance characteristics of a simple version of Optical Burst Switching (OBS) and two variations of Optical Time Division Multiplexing (OTDM) scheduling schemes, referred to as statistical Slot by Slot scheduling and deterministic Round Robin scheduling. These three classes of resource sharing methods differ in the manner and degree of co-ordination of resource allocation between the edge nodes and the core switches. We have evaluated the various resource sharing schemes for the case where a single high quality best effort service class supports all offered traffic types. Small buffers are employed at the edge nodes to provide acceptable delay performance for an appropriately small designed level of buffer overflow. Our simulation results show that the OTDM schemes compare favourably with OBS in terms of packet loss and bandwidth utilization while keeping packet delay sufficiently low to meet real time QoS requirements.

The Slot by Slot scheduling approach has been shown to be robust to variations in traffic distribution and can achieve high bandwidth efficiency with acceptably low buffer overflow probability. Accordingly it may be suitable for Metropolitan Area Network applications. The fixed Round Robin scheduling method is less robust to variations in traffic demand distribution as one would expect, however it avoids the need for signalling and reservation at the time slot level and as a result will yield better delay performance than is possible with the slot by slot scheme. The traffic robustness of the round robin scheme can be improved by allowing the slot allocation to vary from frame to frame according to traffic demand. Call by Call or control driven slot allocation similar to conventional TDM switching is one obvious alternative. A data driven approach is also possible where the network updates the frame based slot allocation based on traffic measurements and forecasts. These later alternatives are currently being investigated for both WAN and MAN networks.

The authors would like to thank the Natural Sciences and Engineering Research Council (NSERC) and industrial and government partners, through the Agile All-Photonic Networks (AAPN) Research Network for supporting this work, and OPNET Technologies, Inc., for their Modeler software.

6. References

Anderson T. E., Owicki S. S., Saxe J. B., and Thacker C. P., "High-Speed Switch Scheduling for Local-Area Networks," *ACM Transactions on Computer Systems*, vol. 11, no. 4, pp. 319-352, Nov.1993.

Baldine I., Rouskas G. N., Perros H. G., and Stevenson D., "JumpStart: A just-in-time signaling architecture for WDM burst-switched networks," *IEEE Communications Magazine*, vol. 40, no. 2, pp. 82-89, Feb.2002.

Bianco, G. Galante, E. Leonardi, F. Neri, and A. Nucci, "Scheduling algorithms for multicast traffic in TDM/WDM networks with arbitrary tuning latencies," *Computer Networks*, vol. 41, no. 6, pp. 727-742, Apr.2003.

Bochmann G.V; Hall T.; Yang O.; Coates M.J.; Mason L.G.; Vickers R. The Agile All Photonic Network: An Architectural Outline. *Queen's Biennial Conference on Communications*, Feb.2004

Kar K., Stiliadis D., Lakshman T. V., and Tassiulas L., "Scheduling algorithms for optical packet fabrics," *IEEE Journal on Sel. Areas in Comm.*, vol. 21, no. 7, pp. 1143-1155, 2003.

Liew S. Y. and Chao H. J., *On slotted WDM switching in bufferless all-optical networks*, HOT Interconnects, ed Standford Univ, 2003.

Maach A. and Bochmann G.V, "Segmented Burst Switching: Enhancement of Optical Burst Switching to decrease loss rate and support quality of service", *Proc. Sixth IFIP Working Conference of Optical Network Design and Modelling*, Torino, Italy, Feb.2002

Mason L.G., Vinokurov A., Zhao N., Plant D. Topological Design and Dimensioning of Agile

All Photonic Networks. Submitted to "*Computer Networks*", 2005

McKeown N., "The iSLIP scheduling algorithm for input-queued switches," *IEEE-ACM Transactions on Networking*, vol. 7, no. 2, pp. 188-201, Apr.1999.

Minkenberg C., "Performance of i-SLIP scheduling with large round-trip latency," *High Performance Switching and Routing Workshop*, 2003.

Ramamirtham J. and Turner J., "Time sliced optical burst switching," *22th Annual Joint Conference of the IEEE Computer and Communication Societies*, 3 (30) 2003, pp. 2030-2038.

Shreedhar M. and Varghese G., "Efficient fair quelling using deficit round-robin," *IEEE-ACM Transactions on Networking*, vol. 4, no. 3, pp. 375-385, June1996.

Smiljanic A., "Flexible bandwidth allocation in high-capacity packet switches," *IEEE-ACM Transactions on Networking*, vol. 10, no. 2, pp. 287-293, Apr.2002.

Xu L. S., Perros H. G., and Rouskas G., "Techniques for optical packet switching and optical burst switching," *IEEE Communications Magazine*, vol. 39, no. 1, pp. 136-142, Jan.2001.

Zaim A. H., Baldine I., Cassada M., Rouskas G. N., Perros H. G., and Stevenson D., "Jumpstart just-in-time signaling protocol: a formal description using extended finite state machines," *Optical Engineering*, vol. 42, no. 2, pp. 568-585, Feb.2003.

Zang H., Jue J.P., and Mukherjee J., "Photonic Slot Routing in All-Optical WDM Mesh Networks," *Proc.IEEE Globecom '99*, Rio de Janeiro, Brazil, Dec.1999