Resource Sharing for QoS in Agile All Photonic Networks

Anton Vinokurov, Xiao Liu, Lorne G. Mason

Department of Electrical and Computer Engineering,

McGill University, Montreal, Canada, H3A 2A7

E-mail: mason@ece.mcgill.ca

Abstract

This paper presents our simulation study of distributed scheduling methods for Agile All Photonic Networks using the OPNET Modeler. Both Optical Burst Switching (OBS) and Optical Time Division Multiplexing (OTDM) are studied.

We first describe the OPNET implementation of OBS and OTDM in the context of an Agile All Photonic Network. The technique of discrete event simulation in the OPNET Modeler allows for coordination between ingress/egress edge nodes and core node, where every device can transmit and receive simultaneously.

Based on the OTDM resource allocation schemes, we investigated the quality of service (QoS) using two DiffServ traffic classes namely, Expedited Forwarding (EF) and Best Effort (BE). Two classes of scheduling methods are proposed which are called statistical Slot by Slot and frame based deterministic allocation. Performances of these scheduling schemes are measured by their dropping rate, bandwidth utilization and queuing delay. It has been shown that with the statistical Slot by Slot scheme, the network resources are efficiently utilized and adaptively allocated to the two traffic classes while providing the requested quality of service. Simulation results are presented in various scenarios to illustrate the effects of different network sizes and traffic patterns.

Introduction

The Agile All-Photonic Network (AAPN) project [1] aims at the development of a fast optical switched network to efficiently transport bursty Internet traffic. The AAPN research project addresses specific hardware issues as well as new resource sharing algorithm development and analysis outlined in this paper. 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. No buffering is allowed at the core optical switches as fiber delay lines are commercially unattractive. Technology, architectural option analysis, traffic sharing considerations, topology and cost optimizations were addressed in [2].

To effectively utilize network resources (time, wavelength) while satisfying loss/delays constraints specific link sharing techniques based in Optical Burst Switching (OBS) and Optical Time Division Multiplexing (OTDM) schemes were analyzed with the help of OPNET Modeler 10.5 software. 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. See Figures 1 and 2 below for the network topology and the edge and core node functionality. The set of virtual output queues (VOQs) for the different destination Edge nodes is created at each ingress Edge node. 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. Performance analysis of this algorithm, its tuning parameters and comparison with the traditional OBS scheduler was reported previously in [3].

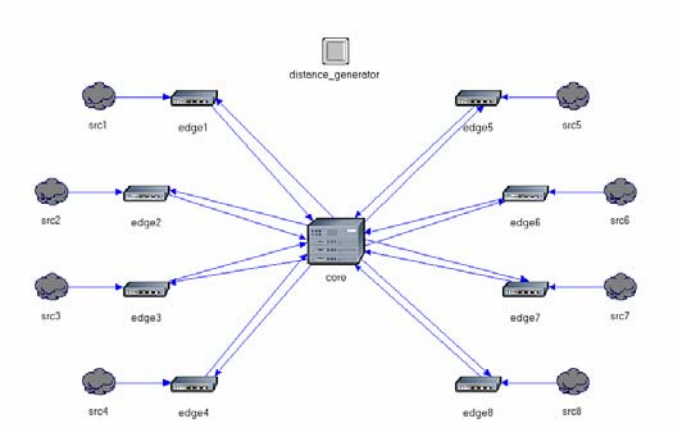

Figure 1. Network structure (8 edge nodes)

Here we are also studying the scheduler performance for real-time and best-effort QoS traffic where data are classified in Edge ingress point, requiring two VOQ's per destination, and different class based scheduling priorities are applied.

Simulation

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 the 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. Two sets of propagation delays were used, one for an average metro network and national network. Both sets were created with Matlab and loaded by init state process of special "distance generator" process. Packet streams delays are then set accordingly to Edge node ID and speed of light.

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

Since we are interested in payload protocol-independent algorithm performance, we decided to simplify our model by implementing all building blocks in the form of a single Modeler "node" entity, constructing edge and core node models as "process models" within that node (fig. 2). That approach exempts us from the necessity of programming the link models, transmitters, packet encapsulation and routing.

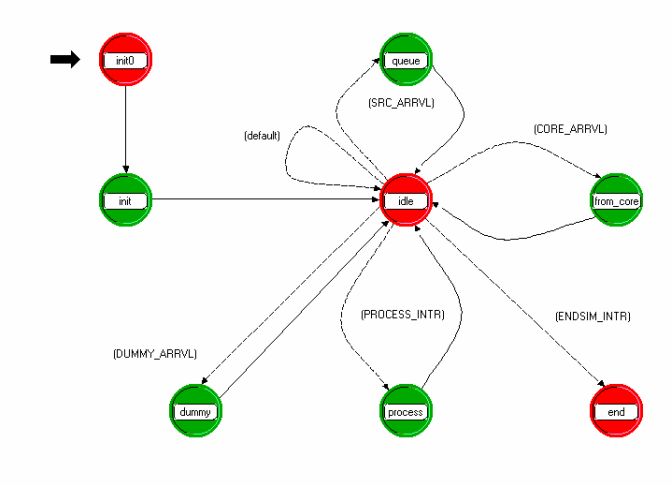

Figure 2. Edge node process model

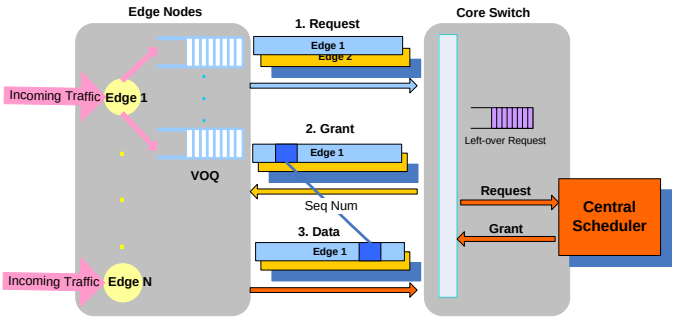

Figure 3. OTDM operation

An edge node has a number of Virtual Output Queues (VOQ, fig. 3) for each destination edge node and QoS service class. (In the studied examples the total number of edges ranges from 2 to 32 with uniform traffic distribution over the destination nodes). The VOQ size is fixed at 400 packets. In case of queue overflow due to service starvation, the newly arriving packets from the 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. For QoS calculations we have two queues for each destination: one (50 packets) for the real time traffic, another (350 packets) is for best-effort data.

OBS [4,5] and OTDM models were implemented in a common way and share the same set of simulation parameters. We collected statistics to determine the performance characteristics, in particular the *packet loss* and *utilization* of bandwidth, and *end to end delay*. Since we are interested in performance counters related to specific values of offered load and not in dynamics during simulation run, there are no

op_stat_write (or similar) API calls. Instead, every Edge node outputs it own counters at the end of simulation while in ENDSIM interrupt. For the originating Edge node there is no direct way to know if the data slot/burst was correctly received by the destination, or what the cumulative delay was. To circumvent this limitation, the destination Edge node forcibly "injects" received packets back onto a special "dummy" stream of source Edge node, allowing it to accurately keep track of and report values mentioned above. Custom Matlab software is then used to extract and display various 2D and 3D dependencies (i.e. *packet loss* vs *offered load* vs *queue length*). Following are the default parameters used in the simulations:

Time slot (τ)	*10 μsec=10^{-5} sec.*
Request threshold	*50~60 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	*Bandwidth of link x duration one time-slot: 10^{-5} x 10^{10}=10^{5} bits.*
Topology	*Metropolitan Area Network (30 km), Wide Area Network (1500 km).*

Table 1: Simulation Parameters

For each value of offered load (amount of traffic generated by the source offered to the network) a discrete-event simulation is performed. To collect reliable statistics more than 1 million packets are forwarded by each Edge. Simulation time is selected in such a way as to allow at least 20 Edge-to-Core round-trips for request/grant frames to achieve stationary state.

Experimental Results for OTDM and OBS

A simple OBS scheduler is implemented as described: 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. There is no collision notification or retransmission of a blocked burst.

The OTDM slot-by-slot scheduler implements data transmission from distinct sources in a coordinated manner. 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 more than one edge node has traffic going to the same egress edge node, one of them is randomly selected for transmission while the others are blocked at the edge nodes until they are granted access by the core. We introduce a packet scheduling protocol so called Adapted PIM scheduling. It combines the procedures in PIM (Parallel Iterative Matching [6]) algorithm ($\log_2 N$ complexity), and also accounts for propagation delay on the transmission links from the edge node to the core switch. At the core switch, the central controller maintains a list of ungranted requests, to ensure that every request gets granted eventually. The longer a request waits in the list, the higher priority it has in the matching. The central controller performs PIM while counting in the priority of each request. For further improvement, we assign random matching to un-matched VOQs in the end of matching procedure despite the absence of requests from it. We call the matching at this stage *leftover matching*.

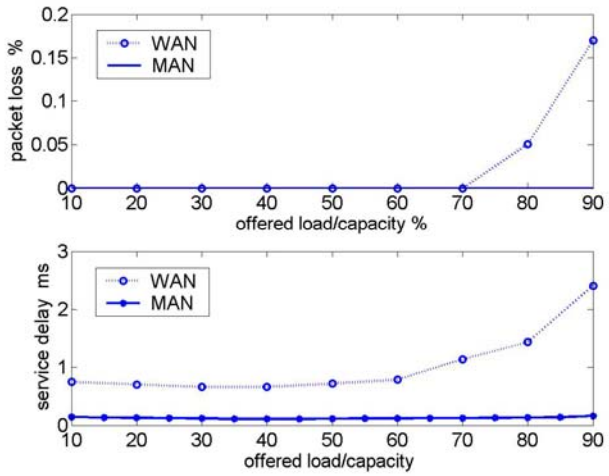

Figure 4. Simulation results for OTDM

No packet loss occurred in the simulation of the MAN topology over the range of traffic loads shown, while packets loss occurs when offered load is around 80% in a WAN topology (fig. 4). There is no additional service delay in the OBS case due to absence of request/grant signalling and associated round-trip time. However, in order to retransmit lost bursts one should implement some kind of back-signalling, retransmission and destination buffering to compensate

out-of-order arrival, which makes OBS very unattractive. In OTDM scheduling, apart from the propagation delay from ingress to egress edge node, at least a round trip delay from ingress edge node to core switch is required for reservation signalling. However, the service delay shown is much smaller than the round trip time (approximately 10 ms). This improvement is due to the use of a request threshold and leftover matching featured in our scheduling algorithm. The request is made if arriving packets to VOQ exceeds the threshold. This threshold is set to be 50 packets, while a slot can in fact carry around 100 packets. Accordingly, the full request and grant delay does not apply to the packets arriving after the threshold is reached. Moreover, the leftover matching can further decrease the grant delay because a VOQ can be served even before any request is issued for it. With VOQs served more often, fewer requests are made to the central scheduler - this decreases processing delay for each grant (fig. 5).

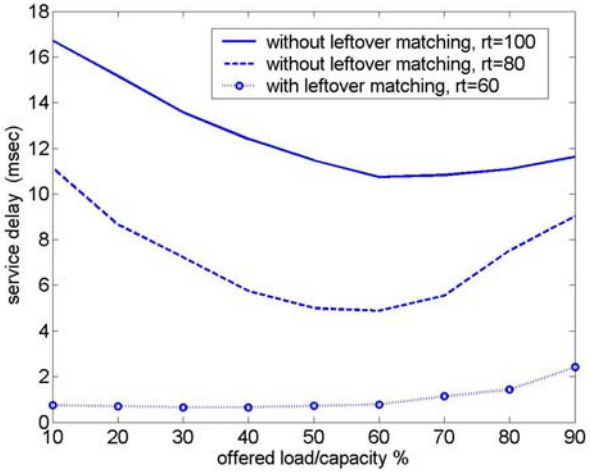

Figure 5. Effect of leftover matching

Scheduling Algorithm for Different QoS Requirements

Based on the OTDM scheduling algorithm, we implemented a scheduler for this two-class transport service scheme. The APIM algorithm is extended with the Strict Priority (SP) discipline, in order to allocate time slots to both EF and BE traffics.

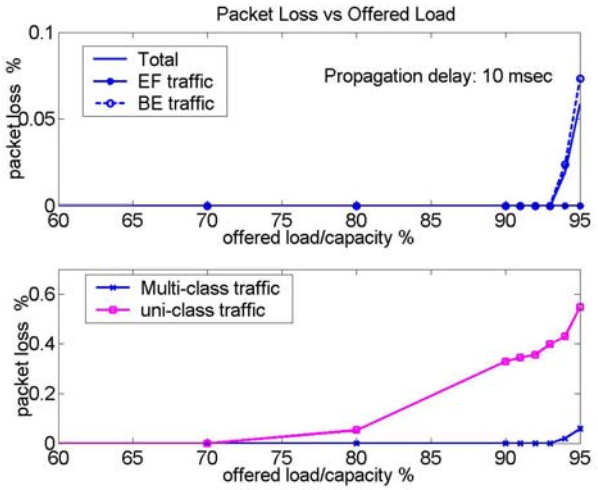

Figure 6. Packet loss vs. offered load

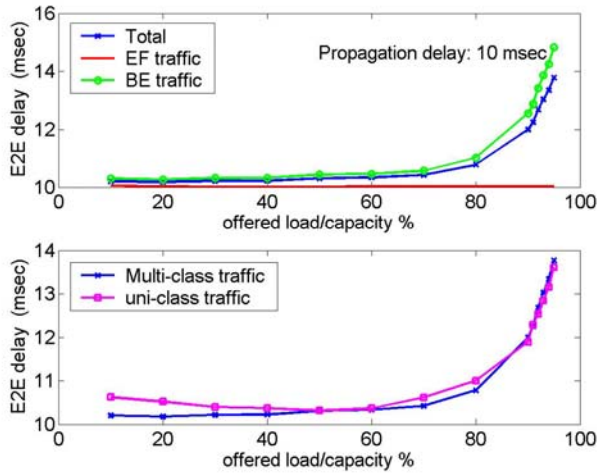

Figure 7. End to end delay vs. offered load

The figures shown above are taken from the simulation scenario with 0.005 sec propagation delay, corresponding to a WAN topology. We find that in a Metropolitan Area Network, a single high quality BE class can provide adequate QoS at high utilization values on the order of 90%. In a WAN topology, the introduction of the EF and BE classes can increase bandwidth utilization by an order of 15% relative to a single BE class to due to the reduced loss rate at high offered load. This behaviour can be explained as follows. When a single high quality BE class is offered, the VOQ buffers must be dimensioned small enough to assure that all packets are served in time to meet the real time needs of the most stringent class. For MANs the buffer delay required for scheduling is small, as it is proportional to the propagation delay which is also small. Thus an aggregate load of 95% link capacity can be

carried without overflowing the small buffer (400 packets capacity). On the other hand for WAN applications with a single BE class, the buffers must be large enough to accommodate the round trip request and grant process. At an offered load of 95% link capacity the uni-class traffic experiences a loss in excess of 0.5%, for the single class QOS buffers dimensioned to meet delay requirements of the real time traffic flows. When separate VOQ buffers are used for the EF and BE classes, one can dimension the buffers for the BE class sufficiently large to avoid overflow as there is no delay guarantee for BE traffic. The EF class is served by a small buffer sufficient for realizing delay requirements. As the EF VOQ buffer is served with non-pre-emptive priority, no buffer overflow occurs until the combined offered load is in excess of 90%. For the uni-class case the no loss maximum utilization is only 70% in WAN applications, implying that significant bandwidth savings can be realized by providing two QoS classes, namely EF and BE instead of over provisioning for a single high quality BE class.

Conclusion and future work

In this paper we described our implementation of the Opnet Modeler framework for resource sharing in a fast-switched optical network, in particular for the AAPN project. This simulation environment allowed us to study various properties of Optical Burst Switching and Optical Time Division Multiplexing schedulers and to choose appropriate tuning parameters to utilize network resources in an efficient way. Our experience allows us to extend an existing network model to support other types of traffic (self-similar, for instance) as well as non-uniform demand distribution.

Acknowledgements

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 the license for Modeler software.

References

[1] 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

[2] Mason L.G., Vinokurov A., Zhao N., Plant D., “Topological Design and Dimensioning of Agile All Photonic Networks”. To be published in special issue of “Computer Networks”, 2005

[3] Liu X., Vinokurov A., Mason L.G., “Performance Comparison of OTDM and OBS Scheduling for Agile All-Photonic Network”, IPIP 2005 MAN conference, Vietnam, April 11-13, 2005

[4] 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.

[5] 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.

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