A Comparison between Time-slot Scheduling Approaches for All-Photonic Networks

Xiao liu, Nahid Saberi, Mark J. Coates and Lorne G. Mason
Department of Electrical Engineering
McGill University
3480 University St, Montreal, QC, Canada H3A 2A7

***Abstract*— The internal switches in all-photonic networks do not perform data conversion into the electronic domain. Although this removal of O-E-O conversion eliminates a potential capacity bottleneck, it also introduces scheduling challenges; photonic switches cannot perform queuing operations, so traffic arrivals at these switches must be carefully scheduled. The (overlaid) star topology is an excellent match for an all-photonic network because it simplifies the scheduling problem. In such a network architecture, optical time division multiplexing (OTDM) approaches for scheduling the state of the central switch in the star are attractive. In this paper, we describe two OTDM algorithms that we have recently developed, one that performs scheduling on a slot-by-slot basis and another that schedules frames of multiple slots. We report and analyse the results of OPNET simulations that compare the performance of these scheduling algorithms.**

***Keywords*—OTDM, scheduling, time slot, frame, all-photonic networks**

I. Introduction

Electronic switches in high-speed networks are increasingly proving to be a capacity bottleneck. Replacement with all-photonic switches is attractive, particularly as photonic devices with sub-microsecond switching capability become available. The inability of the photonic switches to perform queuing introduces network design challenges. Control functionality is required to reduce or eliminate the potential of contention for egress ports. Burst switching and just-in-time reservation approaches [1], [2], and routing and wavelength assignment techniques [3], are some of the many approaches that have been used to address the challenge in general mesh topologies. An alternative approach is to focus on a simpler architecture that reduces the complexity of the control challenge.

In this paper, we focus on the overlaid star topology, as specified in the design for the agile all-photonic network architecture of [4], [5]. This architecture (see Figure 1) consists of edge nodes, where the optical electronic conversion takes place, connected via selector/multiplexor devices to photonic core crossbar switches. The overlaid star topology facilitates the introduction of various approaches to time-sharing link capacity and dramatically reduces the complexity of the control problem. The core switches act independently, so the control problem becomes one of scheduling the switch configurations to achieve a good match with the traffic arrival pattern at the edge nodes.

The star topology also makes the introduction of accurate network-wide synchronization much more feasible, and this enables the application of a range of Optical Time Division Multiplexing (OTDM) techniques for sharing link and switch capacity. These techniques involve the introduction of transmission time slots into the network. A source edge-node must be aware of when it has ownership of a given time-slot and is allowed to transmit to a specific destination edge node. By suitably allowing for the differing propagation delays between various edge nodes and the core, time slots arrive at the core crossbar switch at the same time and can be switched to their appropriate destinations without output port collisions.

The schedule of slot allocation can be fixed and deterministic, for example a round-robin assignment of each output port to the competing source edge nodes. Alternatively, the schedule can adapt to the traffic arrivals through signalling between the edge nodes and the core switch. In this paper we compare the effects of two scheduling algorithms on the performance (utilization and delay behaviour) of a star-topology all-photonic network.

The first of these algorithms is statistical slot-by-slot scheduling. In this case the time slots at the core switch output ports are explicitly reserved on a slot-by-slot basis according to signaling requests from the edge switches, which are driven by traffic arrivals. We evaluate the *Adapted PIM (parallel iterative matching)* algorithm that we proposed recently in [6].

When the propagation delay between the edge nodes and the core is substantial (thousands of time slots), the slot-by-slot scheduling procedure can induce substantial delays because of the need to wait for the granting of a reservation request. In these circumstances, it can be preferable to consider *frames* of multiple slots and make requests based on a prediction of how many slots will be required to service future arrivals. The second algorithm that we investigate is *Minimum Cost Search Frame Scheduling*, which we proposed in [7].

The paper is structured as follows. In Section II, we describe the network architecture under study. In Section III we provide an overview of the statistical slot-by-slot OTDM scheduling algorithm that uses the PIM algorithm. Section IV details the Minimum Cost Search frame-based scheduling approach. Section V describes the simulation experiments we have performed to compare the scheduling approaches and analyses the results. Finally, Section VI draws conclusions and indicates intended extensions.

II. Network Architecture

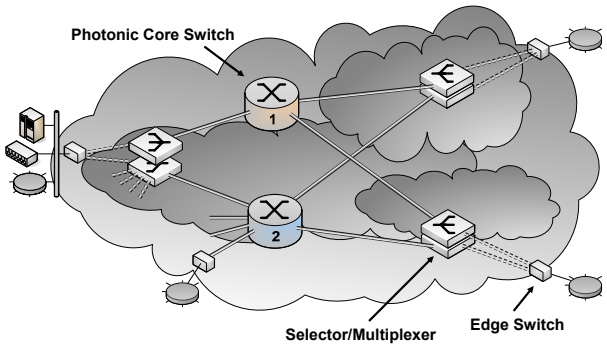

Fig. 1. Architecture of the Agile All-Photonic Network described in [4], [5]. Edge nodes perform electronic-to-optical conversion and transmit scheduling requests to the core photonic node(s). Selectors/multiplexor devices are used to merge traffic from multiple sources onto single fibres and to extract traffic targetted to a specific destination. The structure forms an overlaid star topology (see Figure 2).

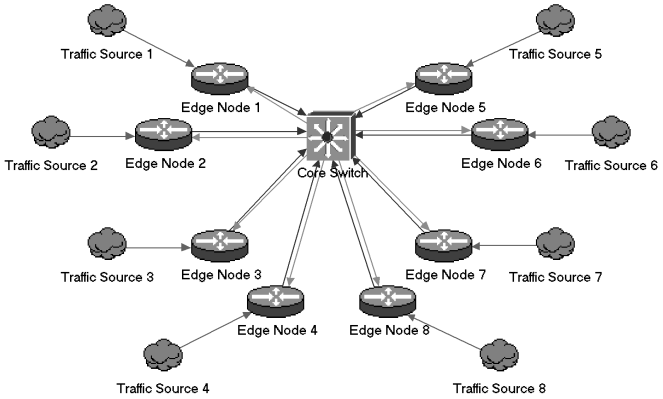

Fig. 2. The star topology induced by the agile all-photonic network architecture.

The design of efficient scheduling methods for all photonic switched networks is challenging because no effective optical buffer devices exist. Once an optical signal is launched into the network, the arrival time at junction points or switches is determined exclusively by the length of the fiber link and the signal's propagation speed. For synchronous time slot switching, the slots arriving on the input ports of the optical space switch must be phase aligned and separated by a guard time sufficient for switch reconfiguration, in order for the slots to traverse the switch without corruption. Phase alignment can be accomplished for star network topologies as well as more general tree network topologies by buffering the inbound traffic in electronic buffers at the edge nodes, and launching the signals at the appropriate offset time in order that all slots arrive in phase at the photonic switch. The underlying assumption is that the core switch and all edge nodes of the star are synchronized relative to a single clock. In general slot phase alignment is much more difficult and often impossible to achieve in general network topologies such as mesh networks. Keslassy et al. [8] have employed delay graph models of networks to examine the class of topologies which admit efficient scheduling methods. While star and tree network topologies have the desired delay graph properties to allow efficient link utilization in all cases, general mesh networks do not.

Star networks with a non blocking core switch are globally non blocking, i.e., there is no internal network blocking as any idle input can be connected to any idle output by appropriately setting the core switch and launching the traffic at the appropriate times. This feature implies that the network is robust to variations in the traffic distribution, a very desirable feature as the precise traffic distribution is difficult to forecast. The drawback of star and more generally tree networks is that they are subject to single point failures. To overcome this difficulty we employ a set of overlaid tree (star-star) topologies, as shown in Figure 1, where a core switch is placed at the root of each distinct tree, and the leaves of each of these trees correspond to distinct groups of Virtual Output Queue (VOQ) buffers in the edge nodes. As the synchronization of distinct trees may be independent from one another, the resulting network topology can be readily synchronized and it is robust to errors in forecasted traffic distribution and resilient to link and node failures provided that there is sufficient spare capacity and adequate restoration procedures exist.

III. Statistical Slot-by-slot Scheduling

In this section we outline a slot-by-slot scheduling algorithm, first described in [6], in which the configuration of the core switch is computed once for each time slot, according to reservation requests from the edge nodes. The proposed algorithm is an adaptation of slot scheduling methods discussed in [9], [10]. The modifications specifically address the challenges of significant and varying propagation delays between the edge-node buffers and the core switch. The configuration of the switch is performed by applying a matching algorithm that identifies ingress-egress node pairs based on the incoming requests.

Each ingress edge node maintains a set of virtual output queues (VOQs), one associated with each egress node. At each time slot, every edge node sends a request to the core switch, specifying whether a specific VOQ has traffic to send and hence requires a slot. A central electronic controller located at the optical switch applies a matching algorithm to determine a schedule based on the arriving requests, and sends grants back to the edge nodes, indicating which VOQs may transmit during specific time slots.

To calculate the schedule for each time-slot, we use a matching algorithm that is an adaptation of PIM (Parallel Iterative Matching) [11], an iterative matching algorithm that randomly identifies input-output pairs. Each iteration of PIM consists of three 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 one of them, selecting at random.

3) *Accept*: If an input receives grants, it accepts one (selecting at random if multiple grants are received).

We have adapted PIM to make it applicable to the AAPN architecture [6]. An edge node may send multiple requests before it has received a single grant. However, an edge node does not send a request immediately upon the arrival of a packet in a VOQ. The number of packets in the VOQ for which a request has not been issued must exceed a specified threshold before a new request is sent. Many packets fit in a time slot, so if this policy is not in place, a lightly-loaded edge node may request more slots than it needs and be granted a disproportionate number of slots. This can lead to poor utilization within slots and blocking of heavily-loaded edge nodes.

Once a request has been issued, the packets associated with that request are "marked" and no second request is issued for them. This avoids the problem of receiving multiple grants for the same set of packets. We must however ensure that every request is eventually granted, although there may be some time delay in the process. To achieve this, the central controller maintains a list of ungranted requests. These ungranted requests have higher priority than requests that have just arrived, and the priority is highest for those requests that have waited longest. The controller applies the PIM algorithm, but instead of each output randomly selecting an input in stage one, it selects the input with highest priority request. If multiple requests have the same priority, one of them is selected at random. As a practical matter, unmatched output ports are randomly assigned to a VOQ and a grant is sent despite the absence of a request.

IV. Minimum Cost Search Frame Scheduling

This section describes an alternative approach, briefly described in [7], for switch configuration based on the periodic scheduling of a *frame*, a block of contiguous time slots. In this paper, we consider fixed-length frames comprised of L slots. Instead of sending information (and potentially a request) every time-slot, edge node i sends a request once per frame, indicating how many slots they will need τ frames into the future, where τ depends on the propagation delay between the edge node and the core. The request is a prediction based on the past traffic arrivals. Here we employ a naive predictor, where the request d_{ij}, the number of slots required from source edge node i to destination edge node j, is equal to the number of slots of traffic that have arrived during the current frame interval.

The set of requests form a traffic demand matrix, $\mathbf{D} = \{d_{ij}\}$, which the central controller uses to form the schedule for the future frame. The frame scheduling algorithm assigns time slots within the frame to source-destination pairs. The aim is to minimize the number of rejected time slot requests in each frame. In order to reduce signalling overhead and to reduce scheduling complexity, we require the algorithm to satisfy the *transparency* property [12]. This requires that the scheduling is only modified for new requests or tear-downs (if d_{ij} decreases or increases).

The minimum cost search algorithm we propose does not achieve optimal utilization, because it does not consider the global allocation problem; instead it allocates requests sequentially on a single time slot basis. The algorithm operates by repeatedly visiting the (i, j) entries in the traffic demand matrix $\mathbf{D}$ in a round-robin fashion; at each visit, if the requested number of slots has not yet been assigned, the algorithm attempts to allocate a single time slot to the (i, j) request. The round-robin allocation results in an approximately fair assignment of slots to each pair.

In order to determine which slot to allocate to the request, we define a *cost* for the allocation of a (i, j) source-destination pair to a time slot pair t_k for k in $1, \ldots L$. This cost is determined entirely by the extant, partial frame schedule. The cost function is:

$$C_{ij}(t_k) = N_{fs}(t_k) + \lambda K_{ij}(t_k), \tag{1}$$

where $N_{fs}(t_k)$ is the number of free sources at this time slot, i.e., the number of sources not transmitting to any other destinations, λ is a small positive constant, and $K_{ij}(t_k) = \{0, 1, 2\}$ is the number of additional switching operations that the core switch must perform to accommodate the allocation. The motivation behind this cost function is simple. The first term represents the current flexibility of that time slot (the number of free sources for future allocation) and reflects the desirability of retaining flexibility by allocating demands to the most constrained slots where possible. The second term reflects the desirability of minimizing the power consumption of the optical switch, which is partially determined by the number of switching operations that it must perform each frame.

The scheduling of a single (i, j) time slot request is performed by first identifying the (i, j)-*eligible* slots in the frame, which are defined as the free time slots during which i is not transmitting to any other destination and j is not receiving from another source. The cost $C_{ij}(t_k)$ of each of these eligible time slots is evaluated, and the demand is assigned to the slot incurring minimum cost. In the case of ties, the demand is assigned to the earliest slot. Deallocation is implemented by a reverse procedure, in which we seek and release the most costly currently-allocated time slot.

V. Simulation Experiments and Results

In this section we report the results of simulations of the scheduling approaches performed using OPNET Modeler [13]. We performed simulations on a 16 edge-node star topology network. The links in the network have capacity 10 Gbps. A time slot is of length 10 μs, and a frame has a fixed length of 1 ms (or 100 slots). The virtual output queues in the simulations have fixed buffer size. Whenever the buffer is full, packets arriving at the edge node are dropped.

In the simulations, traffic sources inject traffic at rates up to 10 Gbps into the edge nodes. The arrival distribution of the data packets is Poisson and the size distribution is exponential with mean size of 1000 bits. We investigated two cases of destination distributions: (i) a uniform case, where sources

send equal amounts of traffic to each destination, and (ii) a non-uniform case, where all destinations receive an equal amount of traffic on average, but each source sends 5 times as much traffic to one destination.

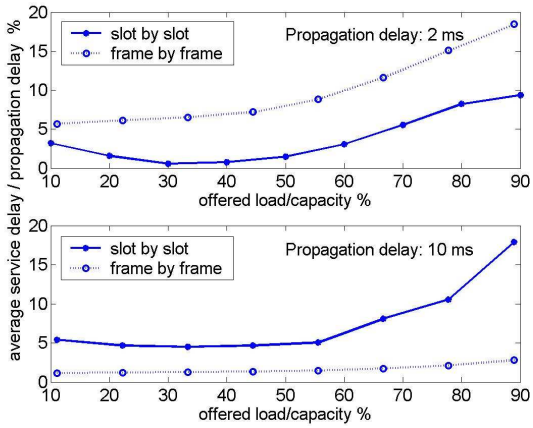

Fig. 3. Average service delay over propagation delay as a function of offered load in uniform traffic scenario. Top panel: 2 ms propagation delay. Bottom panel: 10 ms propagation delay. Here service delay is total end to end delay less propagation delay, and the propagation delay is from ingress edge node to egress edge node.

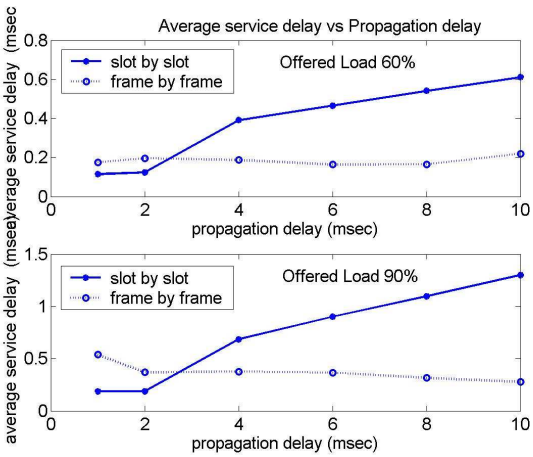

Fig. 4. Average service delay as a function of propagation delay in uniform traffic scenario. Top panel: Offered load of 60%. Bottom panel: Offered load of 90%.

Figure 3 shows the average service delay over a wide range of offered load, from 10% to 90% link capacity. For the slot-by-slot scheme, higher delay is observed for very light offered load (around 10%) than loads in the range of 20% - 50%, because it takes longer time to reach the threshold for issuing a request. Figure 4 compares average service delay as a function of propagation delay for the frame-by-frame and slot-by-slot scheduling methods. The delay components are propagation delay, transmission delay, and queuing delay. For simplicity, we call the latter two components service delay.

The frame-by-frame scheduling method is less sensitive to propagation delay because the round trip time required by the

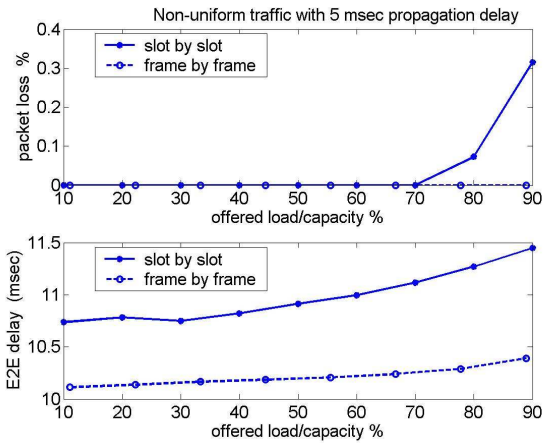

Fig. 5. Performance of the scheduling algorithms with non-uniform traffic as a function of offered load with a propagation delay of 5ms. Top panel: Packet loss ratio. Bottom panel: End-to-end delay.

slot-by-slot scheme for the request-grant-transmit process is avoided. In the frame-by-frame scheme the edge nodes send requests for the predicted traffic demand in advance of the traffic arrival, thereby reducing the delay associated with the grant and request processes. On the other hand the frame-by-frame method may reserve a slot which is unused or under utilized if the actual traffic arriving is less than that forecast. One would anticipate that the accuracy of traffic prediction and the resulting efficiency of the frame-by-frame scheme will depend upon the stability of the traffic demand. The frame-by-frame method on the other hand incurs a delay associated with transmitting a frame. On average the traffic must be buffered for at least half of a frame-duration.

Accordingly, one would anticipate a "break-even" distance where the two methods achieve equal mean delay performance. Below this critical distance the slot-by-slot scheme yields lower delays and would appear suitable for MAN and perhaps regional networks, while the frame-by-frame scheme is more attractive for networks with a large diameter such as in WANs. Figures 3 and 4 indicate that this critical network diameter is around 600km.

For the uniform traffic demand scenarios no buffer overflow occurred during the simulation time. For the non-uniform traffic scenario, as shown in Figure 5, buffer overflow or blocking arises at high traffic loads. Accordingly, by appropriately provisioning link capacity and buffer capacity, high utilization is possible with acceptably low loss and end-to-end mean delay and delay variation or jitter. It is important to note that both scheduling methods adapt to the non-uniform traffic demand with only marginal loss in traffic handling efficiency.

VI. Conclusion and Future Work

Two viable scheduling schemes were specified, implemented and evaluated by simulation for application in WANs and MANs. For Poisson traffic, high utilization is achieved, on the order of 90%, for a single high quality, best effort transport service class. A critical distance exists where the two

schemes break even in terms of service delay performance. For distances larger than this break-even value (approximately 600 km), frame-by-frame scheduling produces marginally smaller end-to-end delay than slot-by-slot scheduling. Thus frame-by-frame is suitable for WANs. The reverse is true for smaller distances typical of MANs where the slot-by-slot protocol yields smaller delay values. Ongoing research is investigating an iterative scheduling mechanism that exploits the fact that propagation distances are heterogeneous.

REFERENCES

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

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

[3] R. Ramaswami and K.N. Sivarajan, "Routing and wavelength assignment in all-optical networks," *IEEE/ACM Trans. Networking*, vol. 3, no. 5, pp. 489–500, Oct. 1995.

[4] G.V. Bochmann, M.J. Coates, T. Hall, L.G. Mason, R. Vickers, and O. Yang, "The agile all-photonic network: An architectural outline," in *Proc. Queens' Biennial Symp. Comms.*, Kingston, Canada, June 2004.

[5] L.G. Mason, A. Vinokurov, N. Zhao, and D. Plant, "Topological design and dimensioning of agile all photonic networks," to appear, *Computer Networks*, 2005.

[6] X. Liu, A. Vinokurov, and L.G. Mason, "Performance comparison of OTDM and OBS scheduling for agile all-photonic network," in *Proc. IFIP Metropolitan Area Network Conference*, Ho Chi Minh City, Vietnam, Apr. 2005.

[7] N. Saberi and M.J. Coates, "Bandwidth reservation in optical WDM/TDM star networks," in *Proc. Queens' Biennial Symp. Comms.*, Kingston, Canada, June 2004.

[8] I. Keslassy, M. Kodialam, T.V. Lakshman, and D. Stiliadis, "Scheduling schemes for delay graphs with applications to optical packet networks," in *Proc. IEEE Workshop High Performance Switch and Routing*, Phoenix, AZ, Apr. 2003.

[9] S.Y. Liew and H.J. Chao, "On slotted WDM switching in bufferless all-optical networks," in *Proc. IEEE Symp. High Performance Interconnects*, Palo Alto, CA, Aug. 2003.

[10] J. Ramamirtham and J.S. Turner, "Time-sliced optical burst switching," in *Proc. IEEE Infocom*, San Francisco, CA, Mar. 2003.

[11] T.E. Anderson, S.S. Owicki, J.B. Saxe, and C.P. Thacker, "High-speed switch scheduling for local-area networks," *ACM Trans. Computer Systems*, vol. 11, no. 4, pp. 319–352, Nov. 1993.

[12] M.A. Marsan, A. Bianco, E. Leonardi, F. Neri, and A. Nucci, "Simple on-line scheduling algorithms for all-optical broadcast-and select networks," *IEEE European Trans. Telecommunications*, vol. 11, no. 1, pp. 109–116, Jan. 2000.

[13] "OPNET modeler 10.5," http://www.opnet.com.