# Time-Slotted Scheduling for Agile All-Photonics Networks: Performance and Complexity

Hana Bilbeisi
School of Electrical and
Computer Engineering
McGill University
Email: hana.bilbeisi@mail.mcgill.ca

Lorne Mason
School of Electrical and
Computer Engineering
McGill University
Email: lorne.mason@mcgill.ca

**Keywords**: OTDM, Input Queued Switches, Distributed Scheduling

***Abstract*—Schedulers in optical switches are still electronic, the performance of these units has a significant impact on the performance of the network and could form a bottleneck in high speed networks, such as AAPN. Four time-slotted scheduling algorithms are investigated in this study, PIM, iSlip, PHM and Adapted-SRA. The study addresses the performance and hardware complexity of each of the algorithms in AAPN. Performance measures were collected using an OPNET model, designed to emulate AAPN. Moreover, the hardware complexity and rate of convergence were evaluated through a hardware implementation of iSlip, and analysis for the rest of the algorithms. The study revealed the superiority of iSlip and PHM over PIM and Adapted-SRA.**

## I. Introduction

Optical Transport Networks (OTNs) are widely seen as the medium of choice for supporting the burgeoning demand for communication services in metro and wide area applications due to their enormous potential capacity. Currently, first generation all-optical networks are being deployed which employ optical cross connects and optical add drop multiplexers in addition to Wave Division Multiplexing (WDM) transmission systems. Such cross-connects provide slow light path switching between end points where optical to electronic transformation takes place. While the agility and transparency inherent in these networks is a definite advance over the existing SONET/SDH transport networks, where the switching function is performed in the electronic domain, the bandwidth granularity is a full wavelength and the switching time and duration of a given switch configuration is quite long, several milliseconds at minimum. This large bandwidth granularity and switching time limits the reach of the optical segment into the loop plant as well as the ability to efficiently multiplex bursty traffic. Optical burst switching (OBS) and Optical Packet Switching (OPS) have been proposed as a solution to the granularity problem. However, unsynchronized OBS suffers from a high switch output port collision rate, unless expensive wavelength conversion is present, while OPS remains a distant technology vision.

The Agile All-Photonic Network (AAPN), is an alternative to OBS, in that it employs Optical Time Division Multiplexing (OTDM) requiring network wide synchronization and distributed scheduling, whereby the output port collisions inherent in unsynchronized OBS are avoided. The AAPN topology is that of overlaid stars, as shown in figure (1), or more generally overlaid trees with Edge nodes (E/O) at the leaves and a fast optical non-blocking switch (sub microsecond switching time) at the root of each star (tree). One can view the AAPN as an *input queued* [20] non-blocking switch where queuing takes place at the edge nodes of the network before E/O conversion is performed. The schedule is computed by a processor co-located with the core fast optical switch. The distinguishing feature of AAPN compared to other input queued switches, is the large and heterogeneous propagation delays between the *VOQs* [19] and the optical non-blocking core switch matrix. This effectively rules out *speed up* for cost reasons as it would involve increasing the capacity of the transmission links between the edge and core switching node. The optical core switch also rules out internal and *output queued* switch architectures as optical buffers are not practical.

High speed network applications require stringent performance measures. Taking Voice over IP as an example, the service is required to support a delay less than 70ms and a packet loss in the range of 0.1%-1%. AAPN is designed to support high speed applications and is required to perform accordingly. The design accommodates for switching time, dedicating a guard band 1us within each time slot for switching. This paper reports on the performance and complexity of slot-by-slot scheduling algorithms for AAPN applications.

Research was first conducted to nominate a number of schedulers for the study. The application of each of the nominated schedulers was further simulated using an OPNET [10] model that emulates an AAPN system [3], to evaluate the effect on the network's performance. Finally the rate of convergence of one of the algorithms was simulated through a hardware model design. Timing assessment of the rest of the algorithms was done by associating results from the hardware implementation model with those reported in the literature review.

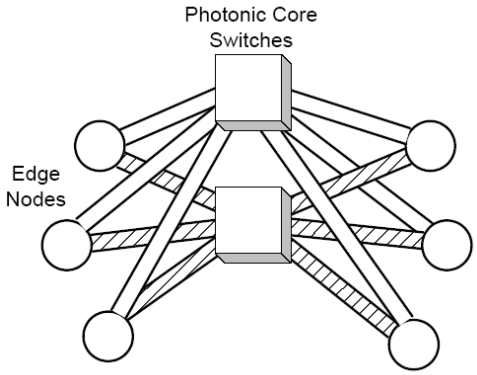



Fig. 1. Overlaid Star architecture: [18]

## II. Slot-by-Slot scheduling schemes

Crossbars are configured by controllers that run scheduling algorithms. An algorithm examines the set of service requests submitted by $N^2$ VOQs, where N is the number of nodes in the network, and forms a matching map between input and output ports, refer to figure (2). The concept is best described by the bipartite matching. A matching behavior could achieve one of the following:

- Maximum matching: Matches the maximum number of edges in an event, achieving high link utilization. The algorithms are highly complex resulting in speed deterioration, causing high service delay and starvation of ports.
- Maximal matching: Iterative matching scheme that forms a subset of maximum matching schemes, where the performance depends on the number of iteration runs.

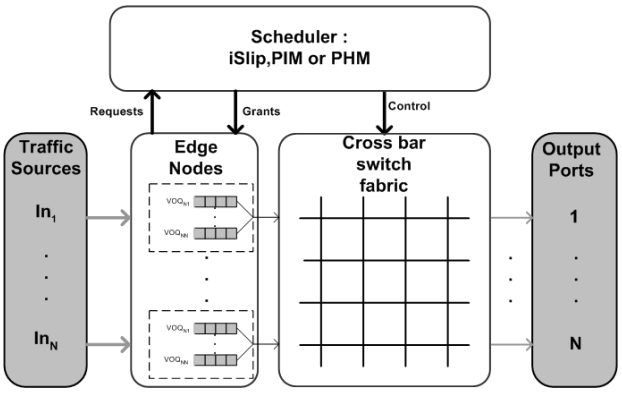



Fig. 2. VOQ cross-bar switch in AAPN

Different scheduling schemes exhibit different characteristics, ranking their aptness for a certain design. The following characteristics were adopted for evaluating each of the scheduling schemes investigated in the research.

1) Performance measures
   a) Link utilization and throughput
   b) Delay
   c) Loss rate
2) Scalability
3) Starvation of nodes
4) Fairness of matching
5) Computational complexity
6) Hardware requirements
   a) Simplicity of implementation
   b) Area required on chip
   c) Memory requirements
   d) Speed requirements
   e) Power consumption
   f) Pipelining amendments for better processing utilization

The application of four scheduling algorithms to an AAPN system was studied and is present in this paper. The algorithms are PIM, as a continuation of a former study [3], iSlip, PHM and Adapted-SRA. The following subsections provide an overview of each one of these algorithms, please refer to [2] for more details about protocols and concepts.

### A. PIM: Parallel Iterative Matching

PIM is a maximal matching algorithm. It randomly chooses edge nodes in the matching process. The algorithm employs independent arbiters that select ports in a probabilistic fashion. Thereby, the performance effect on the network is dependent on the choice of the probabilistic function and the number of iteration runs. Simulations done in [3], [4], and this study illustrate that this algorithm yields link utilization in the range of 85% to 100%. The algorithm is generally starvation free, depending on the quality of the probabilistic function. On average, PIM converges to a maximal match in $O(\log_2 N)$ iterations [4]. Random matching algorithms have some drawbacks. The first is that of the hardware complexity of the algorithm. Each arbiter runs a random number generating function which is highly expensive in terms of processing time and hardware requirements [5]. Such hardware complexities limit the scalability of the network. Finally, PIM is an unfair scheduling algorithm. Nodes will have different admission probabilities, leading to unfairness, especially in situations where the nodes are oversubscribed. Traffic monitoring in [3] resolves the matter by limiting the submission of a service request to a certain number of packets, that however adds to the hardware complexity.

### B. iSlip

iSlip is an iterative matching algorithm derived from the functionality of PIM, following a different node admission procedure. iSlip employs rotating priority (Round Robin) arbitration instead of randomness in matching the nodes. The scheduler is fair and starvation free [6]. However its speed of convergence depends on several factors, mainly the offered load. At high offered load, the algorithm is expected to converge in a single iteration [6]. Nevertheless, analytical studies showed that the algorithm converges in at most N iterations under regular load [5] [6], in an NxN network.

### C. Hardware Requirements of PIM and iSlip

PIM and iSlip follow the same matching protocol, three-step iterative matching. The Steps are outlined below:

1) Request: Each unmatched input submits a service request to each output for which it holds queued cells.
2) Grant: Each unmatched output selects one input request among all the requests it receives (if any), and grants service to it. The choice is either random, as in PIM, or based on a certain criteria.
3) Accept: Each input selects one output grant, if it receives any. Again the choice is based on a certain criteria.

The protocol requires the exchange of approximately $\{(N^2 + 2\,N)\log_2 N\}$ messages. $N^2$ request messages from each VOQ, N grant messages and N accept messages. Furthermore, the total number of messages is multiplied by the number of iterations, by which the algorithm is expected to converge, assuming it is $\log_2 N$ on average. Each of the messages contain $\log_2 N$ bits, which accounts for the hardware requirements of the scheduler, mainly the number of the I/O pins, memory, on-chip area, and power consumption.

### D. PHM: Parallel Hierarchical Matching

PHM algorithms form a different class of schedulers. The algorithms operate by dividing the VOQs into N maximum throughput groups. Each of the groups is assigned to a unit hierarchy in the system. Matching is done with respect to the hierarchical level of each VOQ, acting as a priority measure [7] [8]. A hierarchy matrix H in PHM is used to divide the VOQs into different levels. The matrix is then associated with all the service requests (arranged in a request matrix) to form the matching. The behavior of PHM is totally dependent on the routine by which the hierarchical matrix is updated. Several routines were suggested in the literature [7]. Updating routines must take in consideration the nature of traffic, and the application of the scheduler.

### E. SRA: Single Round-Robin Arbitration

SRA is a maximum-size matching algorithm. The algorithm employs a single round-robin arbiter for each output port. The original SRA algorithm as described in [9], is non-iterative and finds up to N matches in a single time slot. The algorithm could match more than one VOQ within an input, allowing the input to send to more than one output in a single time slot. However, a single port in an optical switch can not be involved in more than one matching in a time-slot, on a single wavelength. Thus the original SRA algorithm was modified to fit an AAPN design. Adapted-SRA is a modified version of SRA, and is utilized in this study. Matching in adapted-SRA is done between single input/output ports where extra monitoring messages are employed. The algorithm is fair [9] and starvation free. Adapted-SRA has a complexity of $O(N^2)$ and is very slow in comparison to other matching algorithms explored in the study. In terms of hardware, the functionality of SRA [9] and adapted-SRA [2] require a controller with larger memory than those needed for other algorithms. Moreover, SRA requires less hardware, since only one set of arbiters are involved in the matching operation.

## III. Simulation Model

This section presents the study of the proposed schedulers in AAPN. The study branches to simulate the performance of the schedulers on one hand, and the hardware implementation, and timing measures on the other hand. An OPNET model was utilized to test the performance of each of the schedulers in the environment of an AAPN system. Furthermore, the hardware implementation of iSlip was evaluated by implementing its functionality on an FPGA chip.

### A. Performance Simulation Model

In previous work [3], one layer of the AAPN star architecture was modeled in OPNET. The scheduling algorithms discussed in section II were coded in that model, and simulated under the effect of different traffic patterns.

*1) Performance Measures:*

Below is an outline of the performance measures collected by the simulation model:

1) End-to-End Delay:
   a) Propagation Delay: depends on the coverage area of the network, and has a great effect on the scheduling and switching performance.
   b) Transmission Delay: independent of switching and scheduling techniques, but affects their performance.
   c) Queue Latency: totally dependent on the scheduling algorithm and the controller's performance.
2) Loss Rate:
   Cross-bar switches are non-blocking, which is crucial in meeting the performance requirements of the network. Consequently, the packet loss rate in the network is mainly due to the overflow of the queues. Therefore the rate by which packets are dropped in the network depends on the size of the VOQs, and the performance of the scheduling algorithm. Studying the effect of a scheduling algorithm on the rate of packet loss requires setting the size of the VOQs to a constant value. The VOQ size could cause a performance bottleneck in the network, so the effect of the size must form a fair tradeoff between the delay and the packet loss [2].
3) Utilization and Throughput:
   The throughput of a network expresses the amount of data delivered from the source to the destination per unit time, and is measured by bits/s.
   Equation (1) illustrates the formulation of the link utilization measure in the design, BP in the equation stands for Blocking Probability.

$$\text{Utilization} = {}^{\text{Carried traffic}}/_{\text{Capacity}}$$

$$\text{Carried traffic} = \text{Offered load}\,(1 - \text{BP}) \quad (1)$$

$$\text{BP} = {}^{\text{Dropped Packets}}/_{\text{offered packets}}$$

*2) Traffic Patterns and Distributions:* The performance of a scheduling algorithm is highly affected by the nature of the input traffic and its distribution among network hosts. Traffic patterns are modeled by their arrival events. The study employs the following arrival events:

- Independent events, modeled by Poisson arrival processes with exponential holding times.
- Events exhibiting long-range dependence (LRD), called Self Similar. These are modeled by generating packets with sizes drawn from a heavy tailed distribution, reference [2] proves the validity of such an approach.

As for the distribution of traffic among network hosts, two options were implemented in the simulated design:

- Uniform traffic: Generated traffic is uniformly distributed among destinations.
- Non-uniform traffic: Traffic is distributed in a weighted fashion, where more traffic is sent to particular destinations than others [3].

*3) Design Parameters:* The following parameters where set in the simulation design to emulate an actual AAPN system.

a) Link Capacity= 10Gbps
b) Slot-Time= 10us
c) Slot-size= $10^5$ bits
d) Switching time= 1us (guard band)

### *B. Hardware Simulation of iSlip*

The study of the hardware implementation of iSlip did not get as much attention as the study of its performance. Serpanos et al discussed the basic design of the request-grant-accept protocol in hardware in [11] [12]. Another study by Gupta and McKeown [13] addressed the delay imposed by the arbiters in iSlip. The study in [13] discussed several arbiters' implementations, presenting the tradeoff between the hardware requirements and complexity of each design. The simulation model used in this study adopts a simplified model that improves the design in [11] and utilizes the optimal arbiter design proposed in [13]. The design was implemented in Quartus II 6.1 [15], a hardware CAD tool.
Figure (3) demonstrates a block diagram of the hardware model utilized in the study, the functionality of each one of the blocks is outlined below:

- Grant Blocks:
  The blocks utilize arbiters that generate grants for service requests.

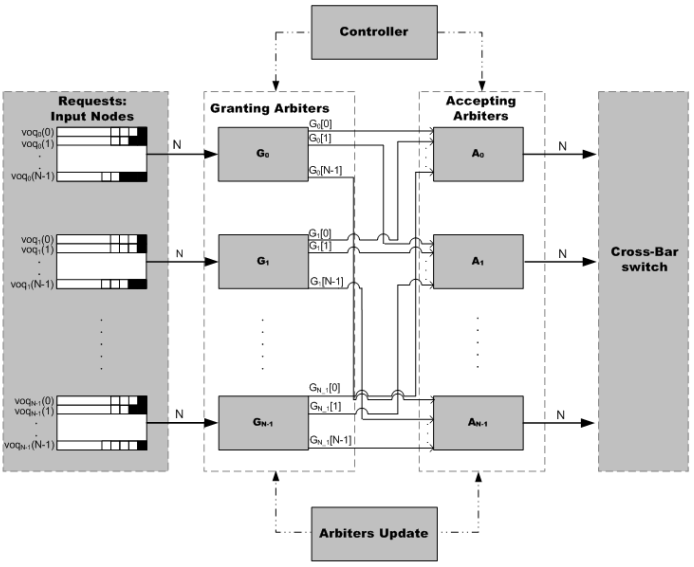



Fig. 3. iSlip Hardware simulation model

- Accept Blocks:
  The blocks utilize arbiters that generate grants for output nodes. Additionally, these blocks follow the following protocol:

  **IF** the scheduler is running the final iteration,
  **THEN** output control signals of this block configure the switch for that time slot.
  **ELSE** the signals are used to block the matched inputs/outputs from being considered in the following matching iteration.

- Arbiters Updating Block:
  The block controls the location pointed at by the arbiter after a certain stage, corresponding to the output of that stage. Only one arbiter updating block is utilized in this model. The block updates the grant arbiters while the system is in the accept phase, and updates the accept arbiters while the system is in the grant phase. The appropriate functionality of the block is triggered by a Finite State Machine (FSM) in the conrtoller, while the values to be-updated are fed by the outputs of the grant and accept blocks.
- Controller unit:
  The controller synchronizes the operation of the blocks to accomplish an ordered execution within a single iteration run. The controller utilizes a three-state FSM, which generates control signals that enable/disable each of the blocks. It also sets the mode of operation of the arbiters updating block to Grant or Accept.

## IV. Results and Analysis

The first part of this section presents results collected from simulating an 8-edge node sample of the perfor-

mance model. Whereas the second part presents results obtained from the hardware implementation model.

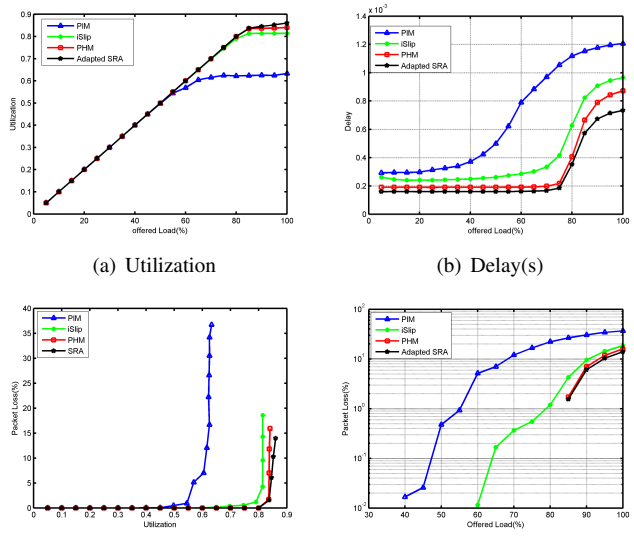
### A. Performance Results

*1) Performance of a single iteration under idealistic traffic model:*
Figure (4) presents graphs generated by the system when uniformly distributed traffic is generated by a Poisson process. The graphs demonstrate the following facts:

a) The performance of one iteration of PIM does not fit the requirements of an AAPN system.

b) The performance results of PHM and the Adapted-SRA are very close, but it was proven that SRA has a higher complexity.

The slight improvement in the performance of SRA was traded for complexity in this study. Therefore Adapted-SRA is replaced by PHM and will not be reported in the rest of the performance results.

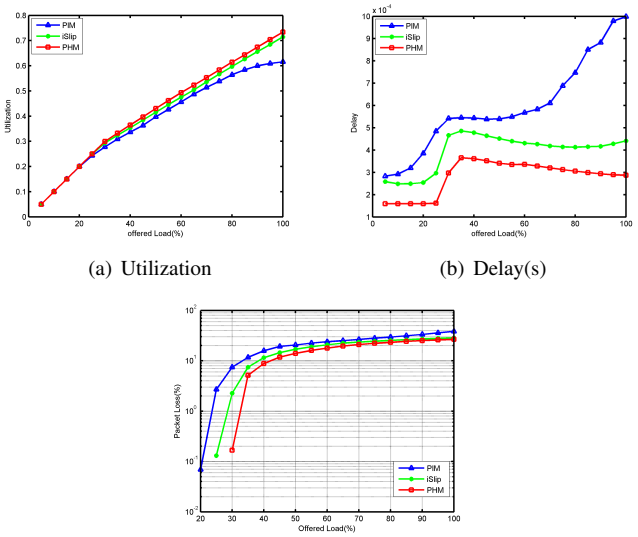



(a) Utilization (b) Delay(s)

(c) Packet Loss, linear scale (d) Packet Loss, logarithmic scale

Fig. 4. 1-iteration run of each of the scheduling algorithms, MAN topology with VOQ size=1000 packets

*2) Poisson arrival process, with traffic non-uniformly distributed among network hosts in a MAN topology:*
Figure (5a) shows that the throughput of the network drops when the traffic is non-uniformly distributed, in comparison to the idealistic case in figure (4a), it also enforces the limitation of the random arbitration. Furthermore, the delay performance graph in figure (5b) demonstrates the superiority of iSlip and PHM over PIM. One would notice a peculiar behavior in the figure around 30% load, where the graphs reach a maximal point and then ramp down again, with the exception of PIM. The peak is due to the non-uniform distribution of the traffic. Some VOQs get blocked and start losing packets while keeping a constant delay, whereas other VOQs queue more packets, these nodes will experience larger delays. Averaging the delay produces higher values than the case when packet loss becomes more pronounced (around 30% load).

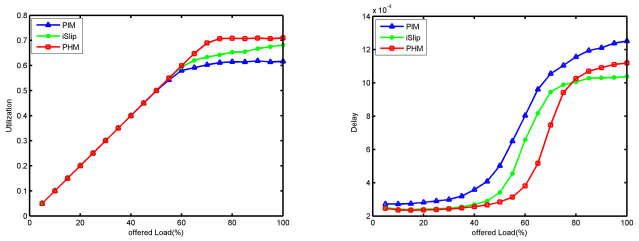



(a) Utilization (b) Delay(s)

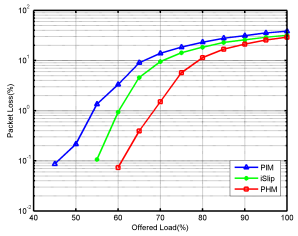
(c) Packet Loss

Fig. 5. Network performance under Poisson, non-uniformly distributed traffic, MAN topology, VOQ size=1000 packets, utilizing 3-iteration runs

*3) Self Similar traffic, uniformly distributed among network hosts in a MAN topology:*
The behavior of the graphs in figure (6) indicates that self similar traffic is supported in a MAN AAPN system. The network throughput drops under self similar traffic, as compared to Poisson arrivals in figure (4a). PIM achieves very poor utilization, figure (6a). One would note that the delay of the network under PHM is greater than that of iSlip for traffic loads greater than 70%, which is not the case for Poisson traffic. In conclusion, PHM would meet the stringent performance requirements of a MAN network under self similar traffic, as long as the load is kept below 75%.
*4) Poisson arrival process, with traffic non-uniformly distributed among network hosts in a WAN topology:*
The graphs in figure (7) illustrate a vigorous increase in the delay under traffic loads of 20%-25%, which is caused by the non-uniform behavior of the traffic. For higher loads, the percentage of packet loss increases, more nodes start blocking packets, and so the delay increases as expected. Comparing these results with those illustrated by figure (5), one would note the drop in performance in both delay and packet loss due to the large distances traversed by the traffic.
*5) Self Similar traffic, uniformly distributed among network hosts in a WAN topology:*

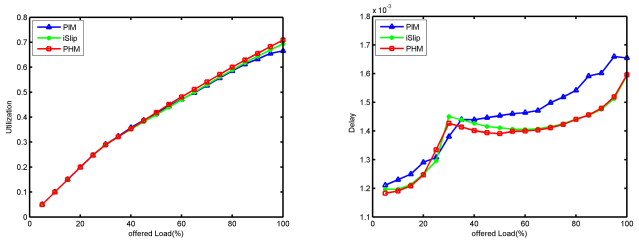



(a) Utilization (b) Delay(s)

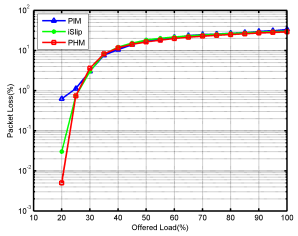



(c) Packet Loss

Fig. 6. Network performance under Self Similar, uniformly distributed traffic, MAN topology, VOQ size=1000 packets, utilizing 3-iteration runs




(a) Utilization (b) Delay(s)



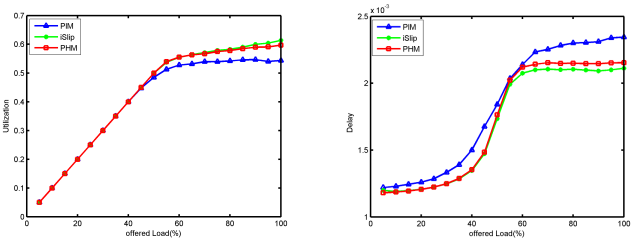

(c) Packet Loss

Fig. 7. Network performance under Poisson, non-uniformly distributed traffic, WAN topology, VOQ size=1000 packets, utilizing 3-iteration runs

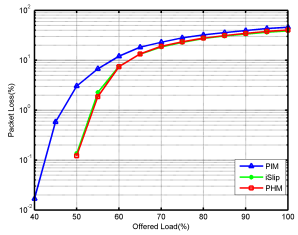
The graphs in figure (8) demonstrate higher delays and packet loss percentages than those illustrated in figure (6). In conclusion, the network performance would support applications with stringent requirements, only if the traffic load is kept below 60%, which would achieve a throughput less than 0.57 in the best case scenario, when PHM is employed.

### B. Hardware Complexity Results

This section presents the results collected from the hardware implementation of iSlip on a Cyclone II, EP2C70F89618 device [15]. The results agree with the




(a) Utilization (b) Delay(s)




(c) Packet Loss

Fig. 8. Network performance under Self Similar, uniformly distributed traffic, WAN topology, VOQ size=1000 packets, utilizing 3-iteration runs

research findings reported in the literature, considering the lack of protocol-steps pipelining in our design. The main objective of the study is to evaluate the worst timing requirements of iSlip, which did not exceed the 1us interval set by AAPN, as shown in table I.

It should be noted that the simulation device has a significant impact on the hardware results. The layout of a device sets a lower limit on the propagation delay of the signals, where signals are sent from one unit to another, that contributes to the difference between the results obtained in our design and other designs. The results demonstrated in tables I and II were collected from running the design in a 4x4,8x8 and 16x16 network environments, to investigate the scalability of the design.

*1) Timing Requirements:*

The clock frequency of the system was set to a value that would support the delay of the bottleneck unit in the design, the grant block.

One would note that the clock frequency decreases as the network expands, since more nodes require more logic, more memory and higher processing times. Table I confirms the applicability of iSlip in an AAPN system, supporting up to 16 nodes. Moreover, the Tiny Tera project [13] confirmed the applicability of a pipelined version of the algorithm on a network with 32 nodes.

*2) Resource Utilization:*

Table II outlines the hardware requirements of iSlip in the Cyclone device. The requirements obviously change from one device to another, but the results give a rough idea about the general requirements of the implementation. It should be noted that the EP2C70F89618 device would not support a 32x32 implementation of the design, while other industrial devices would.

TABLE I
TIMING RESULTS

| N | Worst-case propagation delay (ns) | Clock Frequency MHz | Total Time iterations | time (ns) |
|---|---|---|---|---|
| 4 | 8 | 100 | 2 | 40 |
| | | | 3 | 80 |
| 8 | 13 | 66 | 3 | 80 |
| 16 | 22 | 45 | 3 | 130 |
| | | | 4 | 180 |

TABLE II
RESOURCE UTILIZATION RESULTS

| N | Total logic elements | Total combinational functions | Dedicated logic registers |
|---|---|---|---|
| 4 | 265 | 249 | 96 |
| 8 | 1495 | 1495 | 374 |
| 16 | 8777 | 8477 | 1432 |

*3) Analytical Comparison of iSlip and PHM:*
The timing constraints of PHM were studied in [8] and [14]. An analytical approach was followed by simulations on an ASIC library in [8]. The results confirmed that PHM has a shorter running time than that in arbitration based algorithms. Reference [14] on the other hand, confirmed the applicability of PHM to such high speed networks by synthesizing the algorithm on several devices. Comparing the results obtained by our design and those reported in reference [14], indicates that generally PHM is faster than iSlip, but requires more logic units.

## V. CONCLUSION

The study of the four time-slotted algorithms revealed the aptness of PHM and iSlip in high speed networks. The study proved the high performance of SRA, but the algorithm was ruled out due to its complexity that failed to meet the requirements of AAPN. On the other hand, PIM failed to meet both the performance and complexity requirements. Finally PHM and iSlip proved to meet the performance requirements under certain network environments, mainly MAN topologies. However, a significant drop in performance was observed in WAN topologies. It is worth mentioning that PHM showed a slightly better performance than iSlip in some scenarios. As for the hardware requirements, both algorithms proved to meet the minimum timing requirements of AAPN for a 16x16 network, whereas resource utilization is highly dependent on the choice of hardware. A tradeoff between timing and resource utilization could be a deciding factor between the two algorithms, since PHM is faster but requires more resources.

## REFERENCES


[1] www.aapn..mcgill.ca
[2] Hana Bilbeisi, "Time-Slotted Scheduling for Agile All-Photonics Networks: Performance and Complexity," Masters thesis, Department of Electrical and Computer Engineering, McGill University, 2007,"to be published".
[3] X. Liu, A. Vinokurov, and L. Mason, "Performance Comparison of OTDM and OBS Scheduling for Agile All-Photonic Network," IFIP 2005 Conference on Metropolitan Area Networks, April 2005. Vietnam.
[4] T. Anderson, S. S. Owicki, J. B.Saxe, and C. P.Thacker, "High-Speed switch scheduling for Local-Area networks," ACM Transactions on Computer Systems, pp. 319352, Nov 1993.
[5] N. McKeown and T. E. Anderson, "A quantitative comparison of iterative scheduling algorithms for input-queued switches," Computer Networks and ISDN Systems.
[6] N. McKeown, "The iSLIP Scheduling Algorithm for Input-Queued Switches," IEEE/ACM Trans. Networking.
[7] F. J. Gonzlez-Castao, R. Asorey-Cacheda, C. Lpez-Bravo, P. S. Rodrguez-Hernndez, and J. M. Pousada-Carballo, "On the behavior of PHM distributed schedulers for input buffered packet switches," IEEE Trans. Commun., vol. 51, pp. 1057 1060, July 2003.
[8] F. J. Gonzlez-Castao, R. Asorey-Cacheda, C. Lpez-Bravo, P. S. Rodrguez-Hernndez, and J. M. Pousada-Carballo, "Analysis of Parallel HierarchicalMatching Schedulers for Input-Queued Switches under Different Traffic Conditions," Eighth IEEE Symposium on Computers and Communications, p. 527, 2003.
[9] F. C. Kevin, E. H.-M. Sha, and S. Q. Zheng, "A Fast Non-iterative Scheduler for Input-Queued Switches with Unbuffered Crossbars," 8th International Symposium on Parallel Architectures,Algorithms and Networks, pp. 230235, 2005.
[10] OPNET, http://www.opnet.com.
[11] D. Serpanos, P. Mountrouidou, and M. Gamvrili, "Evaluation of Hardware and Software Schedulers for Embedded Switches," ACM Transactions on Embedded Computing Systems (TECS), vol. 3, pp. 736 759, 2004.
[12] D. Serpanos, P. Mountrouidou, and M. Gamvrili, "Evaluation of Switch Schedulers for Embedded Systems," Eighth IEEE Symposium on Computers and Communications, pp. 541, 2003.
[13] P. Gupta and N. McKeown, "Design and Implementation of a Fast Crossbar Scheduler," IEEE Micro, vol. 19, pp. 2028, Jan. 1999.
[14] E. Soto, E. Lago, and J. Rodrguez-Andina, "FPGA implementation of highperformance PHM / DPHM schedulers," Field Programmable Logic and Applications,2006. FPL 06.
[15] http://www.altera.com.
[16] L. Mason, A. Vinokurov, N. Zhao, and D. Plant, "Topological Design and Dimensioning of Agile All Photonic Networks," Computer Networks, Special issue on Optical Networking.
[17] R. Vickers and M. Beshai, "PetaWeb Architecture," Networks 2000 Symposium, Canada, 2000.
[18] G. Bochmann, T. Hall, O. Yang, M. Coates, L. Mason, and R. Vickers, "The Agile All Photonic Network: An Architectural Outline," Queens Biennial Conference on Communications, Feb 2004.
[19] Y. Tamir and G. Frazier, "High-performance multi-queue buffers for VLSI communications switches," ACM Special Interest Group on Computer Architecture, pp. 343 354, 1988.
[20] M. Karol, M. Hluchyj, and S. Morgan, Input versus Output Queueing on a Space Division Switch, IEEE Trans. Communications, 1987, pp.1347-1356