MINIMUM REJECTION SCHEDULING IN ALL-PHOTONIC NETWORKS

Nahid Saberi and Mark J. Coates

Department of Electrical and Computer Engineering
McGill University
Montreal, QC, Canada
E-mail: nahid.saberi@mail.mcgill.ca, coates@ece.mcgill.ca

ABSTRACT

Internal switches in all-photonic networks do not perform data conversion into the electronic domain, thereby eliminating a potential capacity bottleneck, but the inability to perform efficient optical buffering introduces network scheduling challenges. In this paper we focus on the problem of scheduling fixed-length frames in all-photonic star-topology networks with the goal of minimizing rejected demand. We formulate the task as an optimization problem and characterize its complexity. We describe the Minimum Rejection Algorithm (MRA), which minimizes total rejection, and demonstrate that the Fair Matching Algorithm (FMA) minimizes the maximum percentage rejection of any connection. We analyze through OPNET simulation the rejection and delay performance.

1. 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, just-in-time reservation, and routing and wavelength assignment are some of the many approaches that have been used in general mesh topologies [1,2]. 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 (AAPN) architecture of [3, 4]. 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 core switches act independently, so the control problem is reduced to the task of scheduling the switch configurations to achieve a good match with the traffic arrival pattern at the edge nodes.

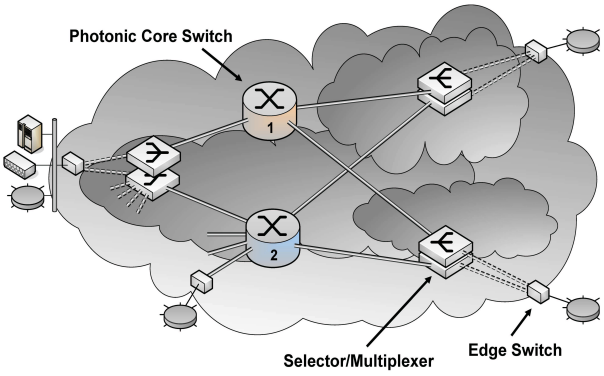

Fig. 1. Architecture of the Agile All-Photonic Network described in [3, 4]. Edge nodes perform electronic-to-optical conversion and transmit scheduling requests to the core photonic node(s). Selectors/multiplexor devices are used to merge traffic from multiple sources onto single fibres and to extract traffic targetted to a specific destination.

The star topology facilitates the introduction of accurate network-wide synchronization [5], and this enables the application of a range of Optical Time Division Multiplexing (OTDM) techniques for sharing link and switch capacity. 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. The slot allocation can be fixed and deterministic, or it can adapt to the traffic arrivals through signalling between the edge nodes and the core switch. In the lat-

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) and industrial and government partners, through the Agile All-Photonic Networks (AAPN) Research Network.

ter case, adaptation can be performed on a per frame basis (a block of slots) or per time-slot basis. Frame-based scheduling is more appropriate for wide-area networks since the impact of propagation delay is reduced (bandwidth is reserved for *predicted* traffic demand in advance of the traffic arrivals) [6]. We focus on fixed-length frames, because this simplifies protocol design and implementation of control functions.

The general objectives of bandwidth sharing are to minimize rejected requests and end-to-end delay, whilst maximizing throughput and maintaining fairness in the network. In this paper, we consider that minimization of the number of rejected requests has highest priority. A secondary objective is to minimize the number of switching operations in a frame in order to reduce the power consumed by the core switch.

Related Work: Scheduling in star-topology networks has been investigated in depth for the past thirty years, so there is naturally much work that is related to what we present here. The majority of the frame-scheduling algorithms proposed for star topologies in optical and satellite networks have focussed on *variable-length* frames (for example, [7–11] and the references therein). We discuss the relationship between our task and the variable-length problem in Section 3. The authors of [12–15] have considered the problem of scheduling a frame of fixed length for star-coupled networks with tunable transmitters/receivers, but do not address the allocation of unused time slots or rejection of inadmissible demand.

Contribution: The combination of a fixed-length frame and a side constraint (the number of switching operations) conspires to introduce some unique aspects to the bandwidth allocation problem. This paper offers the following important contributions: (i) we formulate the AAPN bandwidth allocation task as a scheduling problem, identifying the cost function that must be optimized; (ii) we assess the complexity of this problem, determining the conditions under which the problem is NP-hard and to what extent it can be approximated by a polynomial algorithm, and we outline the parallels with other scheduling problems; (iii) we prove that the Fair Matching Algorithm (FMA), proposed in [16], minimizes the maximum percentage rejection experienced by any connection; and (iv) we propose a novel scheduling algorithm that minimizes total rejection. We analyse the performance of the algorithms through OPNET simulations.

Structure of the paper: Section 2 provides a statement of the scheduling problem that we address. Section 3 identifies the parallels with variable-length frame scheduling problems for star topologies, and Section 4 presents results concerning problem complexity. Section 5 details our proposed frame-based scheduling algorithms and examines their properties. Section 6 describes the simulation experiments we have executed to assess the performance of the scheduling approach and discusses the results. Finally, Section 7 draws conclusions and indicates intended extensions of our work. The Appendix (Section 8) contains proofs.

2. PROBLEM STATEMENT

The AAPN architecture is an overlaid star-topology of N edge nodes that operates over multiple wavelengths [4]. It permits each node to transmit to one destination node and receive from one source node simultaneously *on each wavelength*. We consider that (flow-based) load balancing has been conducted to divide incoming traffic amongst the various stars. The remaining task is to schedule the traffic for each star. We are presented with a demand matrix D, where D_{ij} is the number of slots requested by source node i for destination j during the next fixed-length frame. We consider a frame of length F time slots with W available wavelengths, such that there are $L = FW$ slots for each destination node available for allocation. Herein we focus on the case where $W = 1$ for clarity, but the algorithms and results are easily extended.

Our aim is to devise a schedule S such that the element S_{jk} identifies the source node allocated to the k-th time slot associated with destination j in the frame. The schedule should minimize the number of rejections $REJ(S, D, L)$ whilst also attempting to minimize the number of times that the switch must reconfigure, $N_s(S)$. A switch reconfiguration occurs between two consecutive time slots k and $k+1$ if the allocated source node to any destination j is altered; $N_s(S)$ counts the number of switch reconfigurations in the entire schedule, not merely those within the frame.

The number of rejections is defined as:

$$REJ(S, D, L) = \sum_i \sum_j \max(0, D_{ij} - \sum_{k=1}^{L} \mathbb{I}[S_{jk} = i]), \quad (1)$$

where $\mathbb{I}$ is the indicator function. We can define an objective function (the cost of transmission) as:

$$C(S, D, L) = REJ(S, D, L) + g \,.\, N_s(S)\,, \quad (2)$$

where g is a constant that determines the relative importance of reducing the number of switch reconfigurations.

PROBLEM 1: Solve the following optimization problem for a frame of fixed length L with $C(S, D, L)$ defined by (2) to identify a frame schedule.

$$S_1^* = \arg\min_S C(S, D, L) \tag{3}$$

2.1. Terminology and Definitions

We now define some terminology that will be used throughout the paper and recall some definitions. We denote the line sum of line ℓ of the demand matrix D by LS_ℓ. Note that line ℓ consists of a set of source-destination demands (connections). Each of these connections belongs to two lines (a row and a column). The i-th row represents a link from source i to the optical switch at the core, and the j-th column represents the link from the core to destination node j. The *row-sum*, $r_i(D) = \sum_{j=1}^{N} D_{ij}$, is the total demand at source i, and the *column-sum*, $c_j(D) = \sum_{i=1}^{N} D_{ij}$, is the total demand for destination j.

Definition 1. *A demand matrix D is* admissible *if*

$$\max\{\max_i\{r_i(D)\}, \max_j\{c_j(D)\}\} \leq L, \tag{4}$$

where L is the frame-length, and $r_i(D)$ and $c_j(D)$ are the i-th row-sum and j-th column-sum of the demand matrix, respectively.

For an inadmissible demand matrix, we denote the set of overflowing rows of the demand matrix (rows with $r_i(D) > L$) as O_r, and the set of overflowing columns ($c_j(D) > L$) as O_c. The set of overflowing lines, $O_\ell = \{\ell : \ LS_\ell > L\}$ is the union of O_r and O_c. We define a *critical connection*, or critical demand element, as any demand entry D_{hp} such that $h \in O_r$ and $p \in O_c$. The remaining entries constitute *non-critical* connections/demands.

We now recall the definitions of *feasibility* of rate allocation and *weighted max-min fairness* [17, 18].

Definition 2. Feasibility*: Consider an arbitrary network as a set of links $\mathcal{L}$ where each link $\ell \in \mathcal{L}$ has a capacity $C_\ell > 0$. Let $\{1, \cdots, \zeta\}$ be the set of connections in the network, and H_ℓ the set of all connections passing through link ℓ. Let D_u be the demand (request) of connection u and υ_u be its assigned rate. We call a rate allocation $\{\upsilon_1, \upsilon_2, \cdots, \upsilon_\zeta\}$* feasible*, when for every link ℓ we have:*

$$\sum_{u \in H_\ell} \upsilon_u \leq C_\ell \quad \forall \ell \in \pounds. \tag{5}$$

Definition 3. Weighted max-min fairness*: Let $\omega_u(\upsilon_u)$ be an increasing function representing the weights assigned to connection u at rate υ_u. An allocation $\{\upsilon_1, \upsilon_2, \cdots, \upsilon_\zeta\}$ is* weighted max-min fair *if for each connection u any increase in υ_u would cause a decrease in transmission rate of connection z satisfying $\omega_z(\upsilon_z) \leq \omega_u(\upsilon_u)$. The special case of max-min fairness is obtained by $\omega_u(\upsilon_u) = \upsilon_u$.*

3. RELATIONSHIP TO VARIABLE-LENGTH FRAME SCHEDULING

The most closely related work to the optimization embodied in *PROBLEM 1* is the problem of finding an optimum schedule for a variable-length frame, which has been extensively studied in WDM and satellite systems [7–11]. The goal is to minimize the overall transmission time T:

$$T(S) = \ T_x(S) \ + \ \tau \ . \ N_s(S), \tag{6}$$

where N_s is the number of switch reconfigurations, τ is the switching time, and T_x is the time spent transmitting the traffic [8, 10]. All times are measured in slots.

PROBLEM 2: Solve the following optimization problem for a frame of variable length with total transmission time $T(S)$ defined by (6), observing the constraint that $S \in \mathcal{S}$, the set of schedules that satisfy the demand matrix, i.e., $REJ(S, D, T_x(S)) = 0$.

$$S_2^* = \arg\min_{S \in \mathcal{S}} T(S) \tag{7}$$

PROBLEM 2 is NP-hard for non-negligible values of τ [10, 19]. Crescenzi et al. demonstrate that it cannot be approximated by a polynomial algorithm within a factor less than $\frac{7}{6}$ [19]. For small values of τ the problem can be closely approximated by the minimization of T_x, which is solvable in polynomial time [20–22]. The minimum traffic transmission time is [23]:

$$T_x^* = \max\{\max_i\{r_i\}, \max_j\{c_j\}\}.$$

We can then establish:

Claim 1. *A schedule S_x that minimizes the traffic transmission time, i.e., $T_x(S_x) = T_x^*$, solves PROBLEM 2 to within an approximation factor of $1 + \tau$.*

Proof. The number of switch reconfigurations $N_s(S) < T_x(S)$ and $T(S_2^*) = T_x(S_2) + \tau N_s(S_2) > T_x^*$. Hence if S_x minimizes the traffic transmission time, it satisfies $T(S_x) < T_x^*(1 + \tau) < T(S_2^*)(1 + \tau)$. □

For the special case of small τ, approximate algorithms that attempt to minimize N_s subject to the constraint that T_x is minimum have been proposed in [7, 8, 11, 19]. The algorithms achieve minimum traffic transmission time, T_x^*, but do not guarantee minimum *total* transmission time, $T(S_2^*)$, unless the switching overhead is completely neglected. The *EXACT* algorithm, presented in [11, 19], achieves a minimum traffic transmission time, T_x^* and the derived schedule has at most $N_s = N^2 - 2N + 2$ switch configurations [11]. In the case of an admissible demand matrix, the *EXACT* algorithm generates a schedule S that has length less than L and therefore zero rejection. The *EXACT* algorithm is an iterative procedure that repeatedly performs maximum cardinality bipartite matching (MCBM) to obtain the schedule. It lies at the heart of the algorithms we present in this paper for the case of fixed-length frames.

When τ is very large (on the order of maximum transmission time), the problem is reduced to minimizing T_D subject to the constraint that N_s is minimum. Approximate algorithms for this special case have been proposed in [8,10]. The intermediate scenario, when it is desirable to obtain near minimum solutions for both the number of switchings and the traffic transmission time, has been addressed in [11].

4. COMPLEXITY RESULTS

We establish two results concerning the complexity of *PROBLEM 1*:

Claim 2. *If the demand matrix D is admissible and contains no zero entries (for an $N \times N$ switch and frame of length L) then the EXACT algorithm provides a solution S_E to PROBLEM 1 such that $C(S_E) < C(S_1^*) + g(N^2 - 3N + 2)$.*

Proof. Since the demand matrix is admissible, $T_x^* < L$. Hence the schedule devised by *EXACT* results in zero rejections, $REJ(S, D, L) = 0$. *EXACT* ensures that the number of switch reconfigurations in this solution is less than $N^2 - 2N + 2$. The minimum number of switch reconfigurations for any schedule under the constraint of no zero-entries in the demand matrix is N [24]. Hence the maximum discrepancy is $N^2 - 3N + 2$. □

Theorem 1. *For large g, such that $g > \max(||D||_1 - L, 0)$, where $||D||_1 = \sum_i \sum_j D_{ij}$, PROBLEM 1 is reduced to the problem of minimizing $REJ(S, D, L)$ subject to the constraint that $N_s(S)$ is minimized. For this range of g, PROBLEM 1 is NP-hard.*

See the Appendix (Section 8.1) for a proof.

5. AAPN SCHEDULING ALGORITHMS

In a practical scenario, although it is desirable to reduce power expenditure by minimizing the number of switchings, minimizing the number of rejections is far more important. Hence we address the scheduling problem (*PROBLEM 1*) when g is small. In this case, we can rewrite the problem as:

MINREJ(D,L): For a frame of fixed length L with demand matrix D identify a frame schedule S_1^* that satisfies:

$$S_1^* = \arg\min_S REJ(S, D, L) \tag{8}$$

In this section, we describe two algorithms for bandwidth reservation in the AAPN architecture that address fixed-length frame scheduling. The Fair Matching Algorithm minimizes the maximum percentage rejection experienced by any demand, while the Minimum Rejection Algorithm minimizes the total rejection (that is, it provides a solution to *MINREJ(D,L)*).

5.1. Fair Matching Algorithm (FMA)

The $EXACT$ algorithm can be applied directly to the case of fixed length frames (Claim 2 states that it provides a solution for *PROBLEM 1* when g is small and demand admissible). When the demand matrix is inadmissible, the schedule determined by the EXACT algorithm must be truncated after L time slots. This can lead to starvation of some source-destination traffic, and result in unfairness (such as substantially different average service times for traffic arriving at different nodes). The Fair Matching Algorithm (FMA) was described in [16], but therein the emphasis was on achieving fair allocation of extra capacity. Here we concentrate on how FMA behaves in the case of inadmissible demands, and specifically how it treats the connections competing for capacity on overloaded lines.

FMA processes lines one at a time. It identifies the most overloaded line and reduces the demands on that line such that they sum to capacity (L). The nature of this reduction is important: FMA reduces demand proportional to the original demand, i.e. each adjusted demand experiences the same *percentage reduction*. In subsequent iterations, FMA identifies the next most constrained line, taking into account the effect of any previous adjustments, and clamps its demand to capacity. It repeats the process until no lines exceed capacity.

Here we describe how FMA treats demands belonging to the overloaded lines in the set O_ℓ (FMA also adjusts small loads to form a complete schedule [16]). We define $\mathcal{A}_D \subseteq O_\ell$ as the set of unmodified overloaded

lines and $\mathcal{B}_D \subseteq O_\ell$ as the set of modified overloaded lines. Initially $\mathcal{A}_D$ contains all lines in O_ℓ and $\mathcal{B}_D$ is empty. Similarly, we define b_ℓ as the set of modified connection (i.e. the connections whose demands are modified) in line ℓ and a_ℓ as the set of unmodified connections. Initially, a_ℓ contains all the connections passing line ℓ and b_ℓ is empty. Define $S_{a_\ell} \triangleq \sum_{(i,j)\in a_\ell} D_{ij}$ and $S_{b_\ell} \triangleq \sum_{(i,j)\in b_\ell} D'_{ij}$, where D'_{ij} is the modified demand of connection (i,j). Define for each of line in $\mathcal{A}_D$ the value $G_\ell \triangleq \frac{L-LS_\ell}{S_{a_\ell}}$.

FMA performs the following line adjustment when it reduces demand:

$$D'_{ij} = D_{ij} \times \frac{L - S_{b_\ell}}{S_{a_\ell}} \quad \forall\, (i,j) \in a_\ell \tag{9}$$

Algorithm 1 FMA for overloaded lines

while $LS_\ell > L$ for some $\ell \in \mathcal{A}_D$ **do**
  Identify the line $\ell^* = \arg\min_{\ell\in\mathcal{A}_D} G_\ell$.
  Apply (9) to line ℓ^*.
  Transfer ℓ^* from $\mathcal{A}_D$ to $\mathcal{B}_D$.
  Update a_ℓ and b_ℓ for all lines $\ell \in \mathcal{A}_D$.
  Re-evaluate LS_ℓ for all lines in $\mathcal{A}_D$.
  Remove any lines from $\mathcal{A}_D$ that have $LS_\ell \leq L$.
end while
Apply $EXACT$ to $\lfloor D' \rfloor$ to generate S.

The following theorem states that prior to rounding, FMA achieves weighted max-min fair allocation of capacity for the set of connections on the overloaded links. For the proof of the theorem, see the Appendix (Section 8.3).

Theorem 2. *FMA generates a demand matrix D' with weighted max-min fair allocation, where the weight is $\omega(D'_{ij}) = \frac{D'_{ij}}{D_{ij}}$.*

Define the *percentage rejection* as $1 - \frac{D'_{ij}}{D_{ij}}$. Consider the set of demands that experience the highest percentage rejection. Since the weight ω is a monotonically increasing function of allocated rate D'_{ij}, weighted max-min fairness implies that it is impossible to increase the rate allocated to these demands (or decrease the maximum percentage rejection) without violating feasibility. We thus have the following corollary:

Corollary 1. *Subject to the capacity constraints, FMA generates a schedule that minimizes the maximum percentage rejection experienced by any connection.*

5.2. Minimum Rejection Algorithm

In this section we describe an algorithm that generates a schedule that minimizes total rejection. We first develop a theorem that helps to identify a procedure for solving *MINREJ(D,L)*. We commence by defining a max-flow linear programming problem called *MAXFLOW(D,X,L)*.

Problem Y = MAXFLOW(D,X,L): D is a demand matrix, X is a non-negative matrix that specifies capacity bounds, and L is the frame-length (available capacity on each row/column). Matrices D, X and Y are all of size $N \times N$. Identify a nonnegative matrix Y such that $\sum_{h\in O_r}\sum_{p\in O_c} Y_{hp}$ is maximized, subject to the following constraints:

$$\begin{aligned}
Y_{hp} &= 0 \quad \text{if } h \notin O_r \text{ or } p \notin O_c \\
Y_{hp} &\leq X_{hp} \quad \forall\, (h,p) \text{ s.t. } h \in O_r \text{ and } p \in O_c \\
\sum_{p\in O_c} Y_{hp} &\leq r_h(D) - L \quad \forall\, h \in O_r \\
\sum_{h\in O_r} Y_{hp} &\leq c_p(D) - L \quad \forall\, p \in O_c
\end{aligned}$$

The second, third and fourth constraints apply upper bounds on the elements, the *row-sum*s and *column-sum*s of matrix Y respectively. The following theorem establishes a relationship between a solution to the problem *MAXFLOW(D,D,L)* and a solution to the minimum rejection problem *MINREJ(D,L)*. The proof is in the Appendix (Section 8.3).

Theorem 3. *Set A = MAXFLOW(D,D,L). Construct a rejection matrix $D'' = A + Q$, where Q is an arbitrary non-negative matrix such that $Q_{hp} \leq D - A \;\forall\; (h,p)$, $r_h(Q) = r_h(D) - L - r_h(A) \;\forall\; h \in O_r$, and $c_p(Q) = c_p(D) - L - c_p(A) \;\forall\; p \in O_c$. Then if S is a schedule that generates the decomposition $D = D' + D''$, it is a solution to the problem MINREJ(S,D,L).*

We now describe an algorithm to identify a solution A to *MAXFLOW(D,D,L)*. The corresponding maximum flow problem is depicted in Figure 2. We define a network with a source s and a sink t and try to maximize the flow between them. A network flow is a vector $\mathbf{f} = (f_{ij})$ where each f_{ij} is a positive real number representing the flow on arc (i,j), i.e., the flow from i to j. A flow $\mathbf{f}$ is feasible if it satisfies the capacity constraints and it is conserved at all nodes (total flow out of a node equals total flow in). In our problem, the total amount of flow emitted from source s (and therefore arriving at sink t) is equal to the total amount of rejection contributed by A at the critical connections. The rejection at any specific critical connection (A_{hp}) is equal to the flow on arc

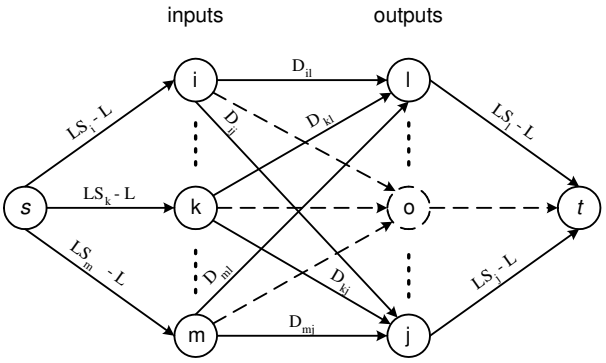

Fig. 2. $s \rightarrow t$ network: In this example the input vertices correspond to the overflowing rows of an arbitrary demand matrix D $(i, k, m \in O_r)$, and the output vertices correspond to the overflowing columns of D $(l, o, j \in O_c)$. The numbers over the edges show the edge capacities which correspond to the upper bounds of flows in our maximization problem. The capacity of each edge (not connected to the source or sink) is equal to the upper bound of the amount of the rejection obtained for the corresponding critical connection.

(h, p). The capacities of the edges (upper bounds) are dictated by the constraints in *MAXFLOW(D,D,L)*. We denote the upper bound on arc (i, j) by $\kappa(i, j)$. So we have:

$$\kappa(s, h) = LS_h - L \quad \forall\, h \in\ O_r$$

$$\kappa(p, t) = LS_p - L \quad \forall\, p \in\ O_c$$

For a feasible flow vector $\mathbf{f}$, an *augmenting path* is a simple path from s to t that can be used to increase flow from s to t. Note that this path is not necessarily directed. On forward arcs in this path ((i, j) points in the direction $s \rightarrow t$) the flow f_{ij} must satisfy $0 \leq f_{ij} < \kappa_(i, j)$, and on backward arcs, i.e. (i, j) is reverse, the flow must satisfy $0 < f_{ij} \leq \kappa(i, j)$.

Ford and Fulkerson presented a solution to the max-flow problem in 1954 [25]. The algorithm starts from an arbitrary feasible flow. In subsequent iterations, the Ford-Fulkerson algorithm identifies an augmenting path, and augments the flow. If the augmenting path is denoted as a set of arcs $\{a_1, a_2, ..., a_k\}$, then the flow augmentation possible is $\delta = \min_{1 \leq i \leq k} \delta(a_i)$, where $\delta(a_i) = \kappa_{a_i} - f_{a_i}$ for forward arcs and $\delta(a_i) = f_{a_i}$ for backward arcs. The flow is adjusted using $f_{a_i} \leftarrow f_{a_i} + \delta$ on forward arcs and on backward arcs using $f_{a_i} \leftarrow f_{a_i} - \delta$. The algorithm iterates until no augmenting path exists, upon which the maximum flow is obtained, as specified by the following theorem:

Theorem 4. *Ford-Fulkerson [25]: Flow* $\mathbf{f}$ *is maximum in graph* $\mathcal{G}$ *if and only if there is no augmenting path in* $\mathcal{G}$ *bearing flow* $\mathbf{f}$.

When there are no lower bounds on capacity, the flow $\mathbf{f}$ defined by $f_{ij} = 0 \ \ \forall (i, j) \in \mathcal{A}$ (the set of arcs in the network) is feasible and can be used to initialize the Ford-Fulkerson algorithm. There are numerous methods for searching for augmenting paths; techniques include shortest path (fewest number of edges) and fattest path (maximum bottleneck capacity along the path) algorithms [26]. Note that the solution to the maximum flow problem (and hence also *MAXFLOW(D,D,L)*) is in general not unique.

To form a Minimum Rejection Algorithm, we first use the Ford-Fulkerson algorithm to identify A. Subsequently we set $D \leftarrow D - A$ and apply FMA to the resultant D. As described in Section 5.1, FMA processes overflowing lines sequentially, adjusting the demand on the line so that it sums to L (thereby identify a line of the rejection matrix). Since we have constructed A so that after modification $D(h, p) = 0$ at any intersection point of overflowing lines h and p, when FMA adjusts one of the overflowing line it does not affect any other overflowing line. This means that after FMA has been applied, it has generated a Q that satisfies the requirements of Theorem 1. In the process, FMA has developed a schedule S that performs the decomposition $D = D' + D''$, where $D'' = A + Q$. The combined Minimum Rejection Algorithm is specified in Algorithm 2.

Algorithm 2 Minimum Rejection Algorithm

1: Apply the Ford-Fulkerson algorithm to solve $A =$*MAXFLOW(D,D,L)*.
2: Set $D \leftarrow D - A$.
3: Apply FMA to D to generate Q and a schedule S.

6. SIMULATION PERFORMANCE

In this section we report the results of simulations of the scheduling approaches performed using OPNET Modeler [27]. We performed simulations on a 16 edge-node star topology network. The links in the network have capacity 10 Gbps and the distance between each edge node and the optical switch is 5 msec. A time slot is of length 10 μsec, and a frame has a fixed length of 1 msec (or 100 slots). Each experiment was run for a duration of 0.2 sec (equal to 200 frame durations) and the results were averaged over 5 repetitions of the simulations. The virtual output queues in the simulations have fixed buffer size

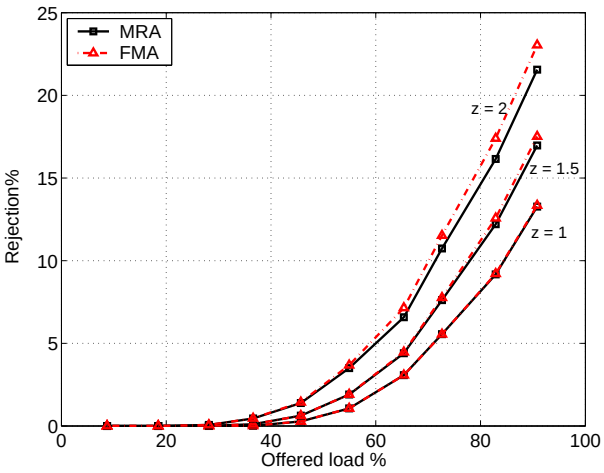

Fig. 3. Comparison between the rejection obtained by FMA and MRA under varying offered load for different factors of imbalanced load (z). Traffic is bursty (generated by on-off sources) and has uniform distribution, aside from the impact of z.

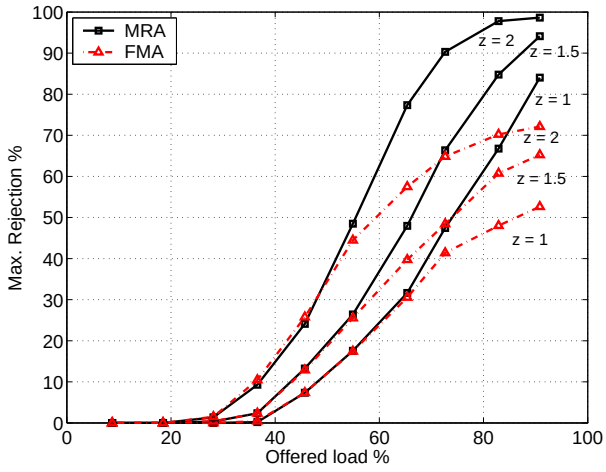

Fig. 4. Comparison between the maximum percentage rejection experienced by any demand after scheduling by MRA and FMA for different values of z and varying offered load.

(90000 packets). Whenever the buffer is full, arriving packets are dropped.

We performed simulations with bursty traffic using on/off traffic sources. Every edge node is equipped with 6 on/off sources. The "on" and "off" periods have Pareto distributions with $\alpha = 1.9$. The mean of the "off" periods is 5 times greater than the mean of the "on" periods. During "on" periods the sources generate packets with an average rate up to the full link capacity (10 Gbps). The rate distribution is exponential. The demand matrix has a non-uniform distribution; each destination receives on average the same amount of traffic, but each source sends five times as much traffic to one specific destination as compared to the others.

Since the behaviour of MRA and FMA only differs when there are critical elements in the demand matrix, we investigate scenarios where critical demands are likely to exist. In order to do this, in each frame we choose one arbitrary source i and one arbitrary destination j. Each source generates z times as many packets for destination j compared to other destinations. Similarly source i generates z times as many packets (to all destinations) as any other source. As z increases, the elements of the demand matrix corresponding to these two edge nodes are more likely to be critical connections; the demand element D_{ij} has even higher likelihood of being critical.

Figure 3 compares the percentage of rejected demand achieved by FMA and MRA as the offered load changes for various values of z. At high load (greater than 70%) with $z = 2$, there are numerous critical elements and MRA begins to achieve less rejection than FMA. The discrepancy is still only 2 percent at 90% load. Figure 4 compares the maximum percentage rejection experienced by any demand when scheduling is performed by FMA and MRA. As the offered load increases, MRA concentrates rejection on the critical elements; the maximum percentage rejection is thus much (up to 25 percent) higher than that achieved by FMA, which distributes rejection fairly amongst all competing connections. Figure 5 compares the average queuing delay experienced by packets when scheduling is performed using FMA and MRA; the approaches yield similar average delay.

7. CONCLUSION AND FUTURE WORK

We have formulated the bandwidth allocation problem in the AAPN network as a scheduling problem with the objective of minimizing rejection whilst reducing the number of switch reconfigurations. We demonstrated that when the demand matrix is inadmissible, the Fair Matching Algorithm minimizes the maximum percentage rejection experienced by any connection. We also proposed a novel algorithm (MRA) that generates a schedule that minimizes the total rejection of demand. Simulations indicate that the discrepancy in total rejection achieved by MRA and FMA is relatively minor, whereas there is a major difference in the fairness of the allocation of rejection. In addition, MRA appears to be less robust to demand prediction errors (when traffic ar-

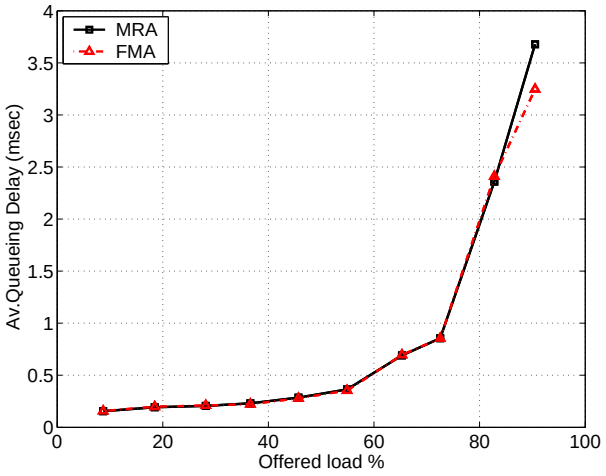

Fig. 5. Average queuing delay performance achieved by MRA and FMA for varying offered load and $z = 2$.

rivals differ substantially from the demand matrix used for scheduling). Thus it appears that whilst MRA achieves minimum rejection schedules, FMA is a better choice for all-photonic scheduling in practice.

8. APPENDIX

8.1. Proof of Theorem 1

Proof. Consider the set of schedules that achieve minimum $N_s(S) = N_s^*$ and label the schedule within this set that achieves minimum rejection S_a. The minimum achievable rejection is no larger than $REJ(S, D, L) = \max(||D||_1 - L, 0)$, where $||D||_1 = \sum_i \sum_j D_{ij}$ (at least one demand element must be satisfied each time-slot). Thus $C(S_a) \leq \max(||D||_1 - L, 0) + gN_s^*$. Now consider schedules that increase the number of switch reconfigurations to $N_s(S) = N_s^* + 1$ and suppose that one of these, S_b, achieves zero rejection, so that $C(S_b) = g(N_s^* + 1)$. The differential in cost $C(S_b) - C(S_a) \geq g - \max(||D||_1 - L, 0)$. If $g > \max(||D||_1 - L, 0)$, then this difference is strictly positive and any schedule solving *PROBLEM 1* lies within the set of schedules that achieve minimum N_s.

In order to prove that the problem is NP-hard for this range of g, we consider *PROBLEM 2*, which for very large values of τ is reduced to minimizing the schedule length subject to the constraint that N_s is minimum. Gopal et al. prove that this problem, which they refer to as the MINSWT problem, is NP-complete [10].

Suppose we had a deterministic polynomial algorithm called *solve-G(D,L)* that could solve *PROBLEM 1* for the identified range of g for demand matrix D and a frame of length L. We could then define the algorithm Solve-MINSWT (Algorithm 2).

Algorithm 3 Solve-MINSWT

```
L = 1;
S = solve-G(D,L);
while REJ(S, D, L) > 0 do
    L = L + 1;
    S = solve-G(D,L);
end while
```

Upon termination of this algorithm, the identified schedule S is guaranteed to have the minimum number of switch reconfigurations (as argued above). Since it is also the minimum length schedule that achieves $REJ(S, D, L) = 0$ it is also a solution to *PROBLEM 2* and hence the MINSWT problem. Algorithm 2 is thus a deterministic polynomial algorithm to solve the MINSWT problem. Therefore, solving *PROBLEM 1* for the considered range of g is as hard as solving MINSWT (and any other problem in NP) and hence is NP-hard. □

8.2. Proof of Theorem 2

We first define a *bottleneck link* and state a lemma relating weighted max-min fairness and the existence of bottleneck links; the proof of the lemma appears in [28].

Definition 4. Bottleneck Link: *Given a feasible rate vector υ and a weight vector ω, we say that link ℓ is a* bottleneck link *with respect to (υ , ω) for a connection u crossing ℓ, if $C_\ell = \sum_k \upsilon_k \triangleq F_\ell$ and $\omega_u \geq \omega_k$ for all connections k crossing ℓ.*

Lemma 1. *A feasible rate vector υ with weight vector $\omega = \{\frac{\upsilon_u}{R_u}\}$ is weighted max-min fair if and only if each connection has a bottleneck link with respect to (υ , ω).*

Proof of Theorem 2. Let $u \in \{(i, j), 1 \leq i, j \leq N\}$ index the source-destination connections specified by the demand matrix. We focus on the properties of the modified demand matrix and associated sets at various iterations of the while loop in Algorithm 1, so we index entities by iteration number and note that this indicates the value of the entity at the *start* of the iteration. For example, $\mathcal{A}_D(h)$ denotes the set of unmodified overloaded lines at the start of iteration h of the algorithm.

We prove that FMA achieves weighted max-min fair allocation of the overloaded demand. During each iteration h of the while-loop, FMA identifies the line $\gamma \in \mathcal{A}_D(h)$ such that $G_\gamma(h) = \min\{G_\ell(h); \ell \in \mathcal{A}_D(h)\}$. It

alters the demands in $a_\gamma(h)$ according to (9) and after this modification, there is no subsequent modification of these demands. Substituting (9) into the definition of the weight, we have $\omega_u = 1 + G_\gamma(h)$ for all $u \in a_\gamma(h)$.

We demonstrate that the adjustment at iteration h leads to γ being a bottleneck link (line) for $u \in a_\gamma(h)$, i.e., after this adjustment it holds that $\omega_z \leq \omega_u$ for $u \in a_\gamma(h)$ and $z \in b_\gamma(h)$. Equivalently, we prove that $\min\{G\}$ is monotonically increasing with respect to the iteration number, i.e., $\min\{G(h)\} \leq \min\{G(h+1)\}$. The equivalence follows since the ω_z are obtained from adjustments prior to iteration h.

Suppose that line β has minimum G at iteration $h+1$. Lines γ and β have at most one connection (demand) in common. If there is no common connection, then $G_\beta(h+1) = G_\beta(h) \geq G_\gamma(h)$. If there is a common connection k, then:

$$LS_\beta(h+1) = LS_\beta(h) + D_k(\omega_k - 1) \quad (10)$$
$$S_{a_\beta}(h+1) = S_{a_\beta}(h) - D_k \quad (11)$$

and hence

$$\begin{aligned} G_\beta(h+1) &= \frac{L - LS_\beta(h) - D_k(\omega_k - 1)}{S_{a_\beta}(h) - D_k} \\ &= \frac{S_{a_\beta}(h)G_\beta(h) - D_k(\omega_k - 1)}{S_{a_\beta}(h) - D_k} \\ &\geq G_\gamma(h) \end{aligned} \quad (12)$$

where the last inequality follows from substitution based on $G_\beta(h) \geq G_\gamma(h) = \omega_k - 1$.

Thus the application of FMA upon an inadmissible demand matrix D leads to the generation of a bottleneck link for each connection u with weight $\omega_u = \frac{D'_u}{D_u}$. By Lemma 1, this establishes that FMA achieves weighted max-min fair allocation of adjusted demands D'.

□

8.3. Proof of Theorem 3

Proof. Consider an arbitrary rejection matrix D^w and set $B =$*MAXFLOW*(D, D^w, L). Then we can write $D^w = B + Q$ where Q is a non-negative matrix. Now consider the conditions necessary for D^w to achieve minimum rejection. First, $D^w_{hp} = 0$ if $h \notin O_r$ and $p \notin O_c$ (any non-zero values constitute unnecessary rejection).

Now consider a node pair $h \in O_r$, and $p \in O_c$ in Figure 2, and the edges (S, h), (h, p) and (p, O). Since B achieves maximum flow, then the flow of at least one of these edges is at full capacity. Therefore, at least one of the following holds:

1. $B_{hp} = D^w_{hp}$
2. $\sum_{j \in O_c} B_{hj} = r_h(D) - L$
3. $\sum_{i \in O_r} B_{ip} = r_p(D) - L$.

If the first equation is true, then $Q_{hp} = 0$. The second equation implies that B has provided the necessary rejection at row h, but $\sum_{j \in O_c} Q_{hj} = 0$ does not necessarily hold; the other overflowing columns may enforce additional rejections on D^w_{hp} which causes $Q_{hj} > 0$ for some $j \in O_c$. We have a similar property for the third equation. Therefore Q is composed of two distinct types of lines which cover all of its nonzero elements:

Type I: The lines composed of $Q_{hp} \geq 0$, and $Q_{hj} \geq 0$ or $Q_{ip} \geq 0$, for $h \in O_r, p \in O_c, i \notin O_r$, and $j \notin O_c$; these lines correspond to the lines in D^w with $r_h(D^w) = r_h(D) - L$, or $c_p(D^w) = c_p(D) - L$, which impose additional rejections to (h,p) elements after obtaining $B =$*MAXFLOW*(D, D^w, L). Consequently we have $r_h(Q) = r_h(D) - L - r_h(B)$, or $c_p(Q) = c_p(D) - L - c_p(B)$.

Type II: The lines composed of $Q_{hp} = 0$, and $Q_{hj} \geq 0$ or $Q_{ip} \geq 0$, for $h \in O_r, p \in O_c, i \notin O_r, j \notin O_c$; for these lines $B_{hp} = D^w_{hp}$ $\forall h \in O_r, p \in O_c$ holds. Therefore additional rejection on these lines is calculated from: $r_h(Q) = r_h(D) - L - r_h(B)$, or $c_p(Q) = c_p(D) - L - c_p(B)$.

Based on this discussion, we can express the total number of rejections, $|D^w|$ as:

$$\begin{aligned} |D^w| &= \sum_h \sum_p (B + Q) \\ &= |B| + \sum_{h \in O_r} (r_h(D) - L - r_h(B)) \\ &\quad + \sum_{p \in O_c} (c_p(D) - L - c_p(B)) \\ &= \sum_{h \in O_r} (r_h(D) - L) + \sum_{p \in O_c} (c_p(D) - L) - |B| \end{aligned} \quad (13)$$

Therefore, in order for D^w to achieve minimum rejection, $|B|$ must be maximized (the first two terms are functions solely of D and L). Compare the solutions $B =$*MAXFLOW*(D, D^w, L) and $A =$*MAXFLOW*(D, D, L). Since $D^w_{hp} \leq D_{hp}$ for any (h, p), the constraints in the second problem are looser, which implies that $|A| \geq |B|$, irrespective of the particular values in D^w. Note that A is also a solution to *MAXFLOW*(D, A, L).

Hence if we ensure that $D^w_{hp} \geq A_{hp}$ for all (h, p), we derive $|B| = |A|$, which implies that $|B|$ attains its maximum value (and hence $|D^w|$ is the minimum rejection).

We can thus construct a rejection matrix that achieves minimum rejection by solving A = *MAXFLOW*(D, D, L), and setting $D^{''} = A + Q$, where Q satisfies the constraints specified in the theorem. If a schedule S decomposes the demand into an allocated matrix D' and this rejection matrix $D^{''}$, then it achieves minimum rejection. □

9. REFERENCES

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

[2] 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.

[3] 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.

[4] L.G. Mason, A. Vinokurov, N. Zhao, and D. Plant, "Topological design and dimensioning of agile all photonic networks," *Computer Networks*, vol. 50, no. 2, pp. 268–287, Feb. 2006.

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

[6] X. Liu, N. Saberi, M.J. Coates, and L.G. Mason, "A comparison between time-slot scheduling approaches for all-photonic networks," in *Int. Conf. on Inf., Comm. and Sig. Proc (ICICS)*, Bangkok, Thailand, Dec. 2005.

[7] A. Ganz and Y. Gao, "A time-wavelength assignment algorithm for a WDM star network," in *Proc. IEEE Infocom*, Florence, Italy, 1992.

[8] A.Ganz and Y.Gao, "Efficient algorithms for SS/TDMA scheduling," *IEEE Trans. Comm.*, vol. 40, pp. 1367–1374, August 1992.

[9] C. A. Pomalaza-Raez, "A note on efficient SS/TDMA assignment algorithms," *IEEE Trans. Comm.*, vol. 36, pp. 1078–1082, 1988.

[10] I. S. Gopal and C. K. Wong, "Minimizing the number of switchings in an SS/TDMA system," *IEEE Trans. Comm.*, vol. 33, pp. 1497–1501, June 1985.

[11] B. Towles and W. J. Dally, "Guaranteed scheduling for switches with configuration overhead," *IEEE/ACM Trans. Networking*, vol. 11, pp. 835–847, October 2003.

[12] K. Bogineni, K. M. Sivalingham, and P. W. Dowd, "Low-complexity multiple access protocols for wavelength-division multiplexed photonic networks," *IEEE J. Sel. Areas Comm.*, pp. 590–604, May 1993.

[13] G. N. Rouskas and M. H. Ammar, "Analysis and optimization of transmission schedules for single-hop wdm networks," in *Proc. IEEE Infocom*, San Francisco, CA, May 1993, pp. 1342–1349.

[14] 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. Telecom.*, vol. 11, no. 1, pp. 109–116, Jan. 2000.

[15] A. Bianco, D. Careglio, J.M. Finochietto, G. Galante, E. Leonardi, F. Neri, J. Sol-Pareta, and S. Spadaro, "Multiclass scheduling algorithms for the david metro network," *IEEE J. Sel. Areas Comm.*, Oct. 2004.

[16] N. Saberi and M.J. Coates, "Fair matching algorithm: Fixed-length frame scheduling in all-photonic networks," in *IASTED Int. Conf. Optical Comm. Sys. and Networks*, Alberta, Canada, July 2006.

[17] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, Englewood Cliffs, NJ, 1992.

[18] P. Marbach, "Priority service and max-min fairness," *IEEE/ACM Trans. Networking*, pp. 733–746, Oct. 2003.

[19] P. Crescenzi, X. Deng, and C. H. Papadimitriou, "On approximating a scheduling problem," *J. Combinatorial Optimization*, vol. 5, pp. 287–297, 2001.

[20] G. Bongiovanni, D. Coppersmith, and C.K. Wong, "An optimal time slot assignment algorithm for an SS/TDMA system with variable number of transponders," *IEEE Trans. Comm.*, vol. 29, pp. 721–726, Oct. 1981.

[21] I.S. Gopal, G. Bongiovanni, M. A. Bonuccelli, D. T. Tang, and C. K. Wang, "An optimal switching algorithm for multibeam satellite systems with variable bandwidth beams," *IEEE Trans. Comm.*, vol. 30, pp. 2475–2481, Nov. 1982.

[22] T. Inukai, "An efficient SS/TDMA time slot assignment algorithm," *IEEE Trans. Comm.*, vol. 27, pp. 1449–1455, May 1979.

[23] T. Gonzalez and S. Sahni, "Open shop scheduling to minimize finish time," *J. ACM*, vol. 23, pp. 665–679, Oct. 1976.

[24] R.M. Karp, "Reducibility among combinatorial problems," in *Proc. Complexity of computer computations*, R.E. Miller and J.W. Thatcher, Eds., New York, NY, 1972, pp. 85–103, Plenum Press.

[25] L. R. Ford, Jr., and D. R. Fulkerson, "Maximal flow through a network," *Canadian. J. Math.*, pp. 399–404, 1956.

[26] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *J. Assoc. Comput. Mach.*, pp. 248–264, 1972.

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

[28] N. Saberi and M.J. Coates, "Fair matching algorithm: An optimal scheduling algorithm for the AAPN network," Tech. Rep., McGill University, Montreal, Canada, Sept. 2005, available at http://www.tsp.ece.mcgill.ca/Networks/publications.html.