# FAIR MATCHING ALGORITHM: FIXED-LENGTH FRAME SCHEDULING IN ALL-PHOTONIC NETWORKS

Nahid Saberi and Mark Coates
Department of Electrical and Computer Engineering
McGill University
Montreal, QC, Canada
email: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 they introduce network scheduling challenges. In this paper we focus on scheduling fixed-length frames in all-photonic star-topology networks. We describe the Fair Matching (FMA) and Equal Share (ESA) algorithms, novel scheduling procedures that result in max-min fair allocation of extra demand and achieve zero rejection for admissible demands. We analyze through simulation the delay and throughput performance.

**KEY WORDS**
scheduling, star topology, max-min fairness, matching algorithms, optical networks.

## 1 Introduction

In high speed optical networks, opto-electronic conversion has the potential to be a capacity bottleneck, so the insertion of all-photonic switches becomes attractive. All-photonic switches are currently incapable of performing queuing, so packet transmissions at edge switches must be carefully controlled. Burst switching and just-in-time reservation approaches, and routing and wavelength assignment techniques have been proposed to address this challenge in general mesh topologies [1, 2]. Simpler network architectures allow simpler, more efficient schemes. The authors of [3, 4] advocate an *agile all-photonic network (AAPN)* architecture, an overlaid star topology consisting of edge nodes, where the opto-electronic conversion takes place, connected via selector/multiplexor devices to independent photonic core crossbar switches.

Two tasks must be addressed in the process of scheduling traffic. First, the traffic arriving at an edge node must be divided among the stars. We assume that this task has been accomplished using a flow-based load balancing approach. We focus on the second task: designing (and adapting) schedules for the traffic allocated to a single star. In wide-area networks with substantial signaling delay, scheduling must be performed based on *predictions* of traffic demand in advance of traffic arrivals. It is more efficient to schedule *frames*, or blocks of slots, rather than allocate capacity on a slot-by-slot basis [5]. In this paper, we address frames of *fixed length*, because this simplifies protocol design.

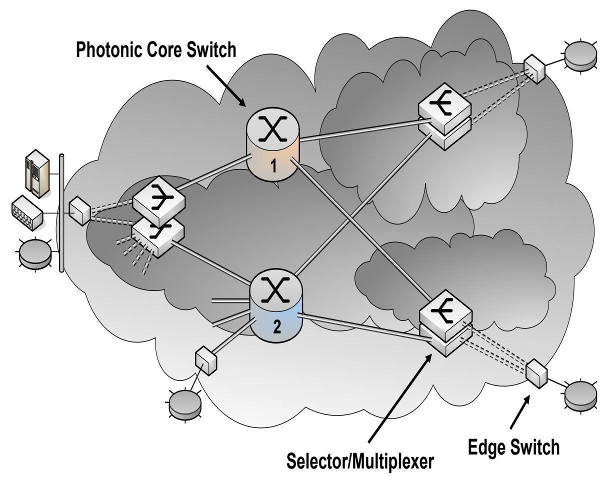


Figure 1. Architecture of the Agile All-Photonic Network described in [3, 4]. Edge nodes perform electronic-to-optical conversion and transmit scheduling requests to the core photonic node(s). Selectors/multiplexor devices are used to merge traffic from multiple sources onto single fibres and to extract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

**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 have focused on *variable-length* frames (for example, [6–9] and the references therein). The authors of [10–12] 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. Finally, algorithms achieving max-min fairness properties have been proposed in other networking contexts such as priority service allocation [13].

**Contribution**: We formulate the AAPN bandwidth allocation task as a scheduling problem. We then propose novel scheduling algorithms that explicitly address inadmissible demands and small offered loads, achieving zero rejection if the demand is admissible and providing fair allocation of free time-slots. We discuss the fairness properties of the derived schedules and analyze performance through simulations.

**Structure of the paper**: Section 2 provides a statement of the scheduling problem that we address. Section 3 details our proposed frame-based scheduling algorithms and examines their properties. Section 4 describes the simulation experiments we have executed to assess the performance of the scheduling approaches and discusses the results. Section 5 draws conclusions and indicates intended extensions.

## 2 Problem Definition

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 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 define the following *line sums* of the demand matrix. The *row sum*, $r_i = \sum_{j=1}^{N} D_{ij}$, is the total demand at source $i$, and the *column sum*, $c_j = \sum_{i=1}^{N} D_{ij}$, is the total demand for destination $j$. It is important to achieve zero rejection if the demand is *admissible*; a demand matrix $D$ is *admissible* for a frame of length $L$ if

$$\max\{\max_i\{r_i\}, \max_j\{c_j\}\} \leq L, \tag{1}$$

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 number of rejections achieved by such a schedule, for a demand matrix $D$ and frame of length $L$, is:

$$REJ(S, D, L) = \sum_i \sum_j \max(0, D_{ij} - \sum_{k=1}^{L} \mathbb{I}[S_{jk} = i]), \tag{2}$$

where $\mathbb{I}$ is the indicator function. The schedule should achieve zero rejection, $REJ(S, D, L) = 0$, when the demand matrix is admissible, and provide *fair* allocation of extra time slots to the competing connections. We focus on achieving the property of (weighted) max-min fairness (see Section 3 for a definition). We can thus state our design problem as follows.

*PROBLEM 1:* For an admissible demand matrix $D$ and frame of length $L$, generate a schedule $S$ that achieves zero rejection, $REJ(S, D, L) = 0$, and allocates spare capacity in the network to the connections in a (weighted) max-min fair manner.

Closely related to *PROBLEM 1* is the task of finding an optimum schedule for a variable-length frame [6–9]. The goal is to minimize the overall transmission time $T$:

$$T(S) = T_x(S) + \tau N_s(S), \tag{3}$$

where $N_s$ is the number of switch reconfigurations, $\tau$ is the switching time, and $T_x$ is the time spent transmitting the traffic [6,8]. The minimum traffic transmission time $T_x^* = \max\{\max_i\{r_i\}, \max_j\{c_j\}\}$ [14].

The *EXACT* algorithm, presented in [9, 15], addresses primarily the case of negligible $\tau$, and achieves a minimum traffic transmission time, $T_x^*$. Thus in the case of admissible demand matrices, the *EXACT* algorithm generates a schedule $S$ that has length less than $L$ and therefore satisfies the first requirement of *PROBLEM 1*. The *EXACT* algorithm is an iterative procedure that repeatedly performs maximum cardinality bipartite matching (MCBM) to obtain the schedule.

## 3 AAPN Scheduling Algorithms

In this section, we describe two algorithms for bandwidth reservation in the AAPN architecture that address fixed-length frame scheduling and provide solutions to *PROBLEM 1*. If all of the line-sums in the demand matrix are equal to $L$, then the schedule obtained using *EXACT* is a full schedule (in each time slot, every source and every destination are serving requests). However, when the load is not enough to fill the schedule completely, we need a *fair* policy to divide the extra time slots amongst active connections. We now recall the definitions of *feasibility* of rate allocation, *(weighted) max-min fairness*, and *bottleneck link* [16,17].

**Definition 1. 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. Let $D_u$ be the demand (request) of connection $u$ and $\upsilon_u$ be its assigned rate.*

*We call a rate allocation* $\{\upsilon_1, \upsilon_2, \cdots, \upsilon_\zeta\}$ feasible*, when for every link* $\ell$ *we have:*

$$\sum_{u \in H_\ell} \upsilon_u \leq C_\ell \quad \forall \ell \in \pounds. \tag{4}$$

Under the (weighted) max-min fairness condition, the objective is to maximize the minimum rate $\upsilon_u$, subject to network capacity constraints.

**Definition 2. Weighted max-min fairness***: Let* $\omega_u(\upsilon_u)$ *be an increasing function representing the weights assigned to connection* $u$ *at rate* $\upsilon_u$*. A feasible allocation* $\{\upsilon_1, \upsilon_2, \cdots, \upsilon_\zeta\}$ *is* weighted max-min fair *if for each connection* $u$ *any increase in* $\upsilon_u$ *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* $\omega_u(\upsilon_u) = \upsilon_u$*.*

**Definition 3. Bottleneck Link***: Given a feasible rate vector* $\upsilon$ *and a weight vector* $\omega$*, we say that link* $\ell$ *is a* bottleneck link *with respect to (*$\upsilon$ *,* $\omega$*) for a connection* $u$ *crossing* $\ell$*, if* $C_\ell = \sum_k \upsilon_k \triangleq F_\ell$ *and* $\omega_u \geq \omega_k$ *for all connections* $k$ *crossing* $\ell$*.*

The following lemma from [18] relates weighted max-min fairness to the presence of bottleneck links.

**Lemma 1.** *A feasible rate vector* $\upsilon$ *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 (*$\upsilon$ *,* $\omega$*).*

## 3.1 Fair Matching Algorithm (FMA) and Equal Share Algorithm (ESA)

We now describe two novel algorithms for adjusting the demand matrix (prior to application of the *EXACT* algorithm) to address incomplete or inadmissible demand. The two algorithms modify the demand matrix so that all of its lines sum to $L$, the frame length.

FMA and ESA are examples of *water-filling* algorithms. If the demand matrix is admissible, they incrementally assign additional demand to all elements until one of the links reaches capacity (its line-sum is equal to $L$). At that point, the demand elements contributing to that line are clamped. Extra demand is then gradually added to the remaining elements in the matrix until another link (line) reaches its capacity and it too is clamped. The procedure repeats until all lines have reached capacity. The difference between the two algorithms is simply the manner in which extra demand is assigned. In the case of FMA, extra demand is assigned *in proportion to the original demand*, whereas in ESA, extra demand is assigned equally.

The algorithms can be implemented by processing one line at a time. We first choose the most constrained line (the line that would reach its capacity first under the water-filling procedure) and increase its demand to capacity. Then we choose the next most constrained line and increase its demand to capacity. We repeat until all lines have reached capacity.

We denote the line sum of line $\ell$ by $LS_\ell$. Note that line $\ell$ 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 $j$. We define $\mathcal{A}_D$ as the set of unmodified lines and $\mathcal{B}_D$ as the set of modified lines. Initially $\mathcal{A}_D$ contains all lines and $\mathcal{B}_D$ is empty. We define $a_\ell$ as the set of unmodified demands in line $\ell$ and $b_\ell$ as the set of modified demands. Initially, $a_\ell$ contains all the demands and $b_\ell$ is empty. Define $S_{a_\ell} \triangleq \sum_{(i,j)\in a_\ell} D_{i,j}$ and $S_{b_\ell} \triangleq \sum_{(i,j)\in b_\ell} D_{i,j}$. We define for each line the values $H_\ell \triangleq \frac{L-LS_\ell}{|a_\ell|}$ and $G_\ell \triangleq \frac{L-LS_\ell}{S_{a_\ell}}$, where $|a_\ell|$ is the cardinality of $a_\ell$.

For the ESA algorithm, the line with minimum $H_\ell$ is the most constrained line, whereas for FMA, it is the line with minimum $G_\ell$. The demand adjustment we perform on each line for ESA is:

$$D_{ij}^{'} = D_{ij} + \frac{L - LS_\ell}{|a_\ell|} \quad \forall\, (i,j) \in a_\ell \tag{5}$$

In the case of FMA, we perform the line adjustment:

$$D_{ij}^{'} = D_{ij} \times \frac{L - S_{b_\ell}}{S_{a_\ell}} \quad \forall\, (i,j) \in a_\ell \tag{6}$$

Thus far, we have focused on the case where the demand is admissible. However (5) and (6) can also be applied when the demand on a link exceeds capacity. In this case, (5) reduces the demand equally amongst all contending connections, whereas (6) reduces demand in proportion to the original request. Algorithm 1 jointly describes ESA and FMA.

The following theorem states that prior to rounding, ESA achieves max-min fair allocation of extra capacity and FMA achieves weighted max-min fair allocation of extra capacity (weighted relative to the original demand). See the Appendix for the proof.

**Theorem 1.** *If the demand matrix is admissible then ESA generates an adjusted demand matrix* $D'$ *with max-min fair allocation of extra-capacity. FMA generates a* $D'$ *with weighted max-min fair allocation of extra-capacity, where the weight of the connection between source* $i$ *and destination* $j$ *is* $\omega_{ij} = \frac{D'_{ij} - D_{ij}}{D_{ij}}$*.*

**Algorithm 1** FMA or ESA

**while** $LS_\ell \neq L$ for some $\ell$ **do**
  **if** ESA **then**
    Identify the line $\ell^* = \arg\min_{\ell \in \mathcal{A}_D} H_\ell$
    Apply (5) to line $\ell^*$.
  **else if** FMA **then**
    Identify the line $\ell^* = \arg\min_{\ell \in \mathcal{A}_D} G_\ell$.
    Apply (6) to line $\ell^*$.
  **end if**
  Transfer $\ell^*$ from $\mathcal{A}_D$ to $\mathcal{B}_D$.
  Update $a_\ell$ and $b_\ell$ for all lines $\ell \in \mathcal{A}_D$.
**end while**
Apply *EXACT* to $\lfloor D' \rfloor$ to generate $S$.

We now present properties of the demand matrix $D' = \{D'_{ij}\}$ obtained by Alg. 1 prior to rounding.

*Property* 1 Algorithm 1 guarantees full allocation of all links provided $D$ contains no zero elements.

*Property* 2 If there is no natural blocking the maximum total throughput of the network is obtained:

$$\sum_i \sum_j D'_{ij} = N.L. \tag{7}$$

*Property* 3 The while-loop in Algorithm 1 has $O(N^2)$ computational complexity. The *EXACT* algorithm has complexity $O(N^{\frac{5}{2}})$, and hence this is also the complexity of Algorithm 1.

*Property* 4 Algorithm 1 guarantees minimum rejection if the set $O$ of overloaded links contains only rows (input links) or only columns (output links) of $D$. In this case:

$$\min(REJ) = \sum_\ell (LS_\ell - L) \quad \forall \ell \in O, \tag{8}$$

## 4 Simulation Performance

In this section we report the results of simulations of the scheduling approaches performed using OPNET Modeler [19]. 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 $\mu$sec, and a frame has a fixed length of 1 msec (or 100 slots). Every experiment was run for a duration of 0.5 sec (equal to 500 frame durations) and the results were averaged over 5 repetitions of the simulations. The virtual output queues in the simulations have fixed buffer size (90000 packets). Whenever the buffer is full, arriving packets are dropped.

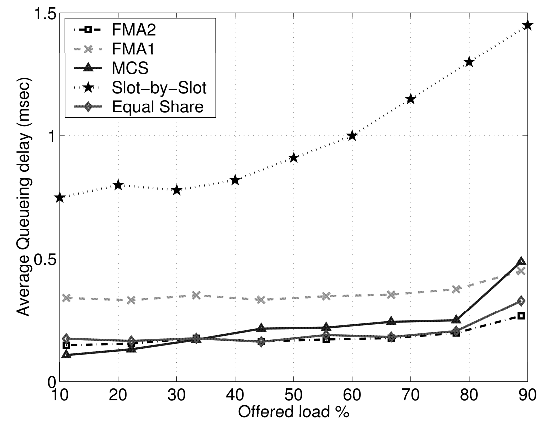


Figure 2. Average queuing delay performance achieved by FMA1, FMA2 , ESA (Equal Share Matching), Slot-by-Slot and MCS under non-uniform, Poisson traffic.

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. Multiple (approximately 100) packets are wrapped into one optical slot. 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. The frame-based scheduling algorithms compute the schedule ahead of time based on the predicted traffic of 10 msec (round-trip delay) in future. In the first set of simulations we used the average of the traffic arrivals over the past 10 frame durations to form the prediction of the demand matrix $D$.

In our simulations FMA1 collocates similar matchings (applying *EXACT* in a standard fashion) and FMA2 and ESA separate them into two batches, one placed towards the start of the frame and one towards the end. This reduces average waiting time. We compare performance to two previous algorithms: Minimum Cost Search (MCS) [20] and a slot-by-slot scheduling approach based on PIM (Parallel Iterative Matching) [5].

Figure 2 shows the queuing delays over a wide range of offered load, from 10% to 90% link capacity under nonuniform traffic (uniform traffic gives similar results). The slot-by-slot algorithm has large average queuing delays, since it is more appropriate for metro and local-area networks [5]. FMA1 generates additional average delay compared to FMA2, which is

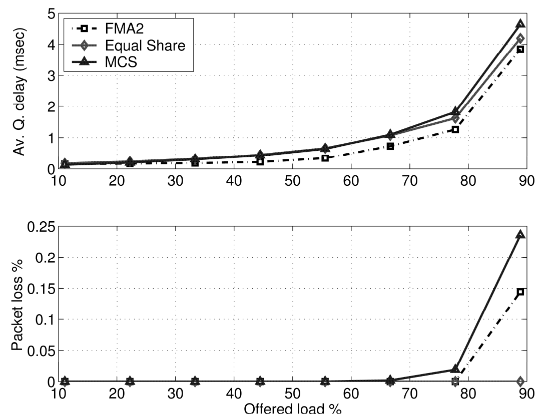


Figure 3. Average queuing delay and packet loss performance for FMA2, ESA and MCS under bursty traffic and non-uniform distribution of the destinations.

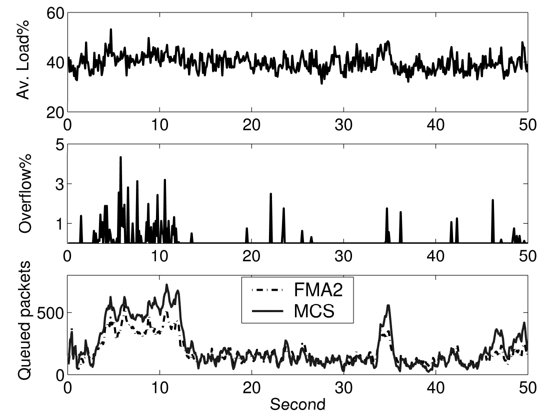


Figure 4. The behaviour of FMA2 and MCS in response to traffic loads derived from Internet traces. Top: the offered load averaged over all source-destination pairs. Middle: the percentage of overflow traffic. Bottom: the overall number of queued packets at the edge nodes.

due to the collocation of matchings. ESA, FMA2 and MCS exhibit similar performance, achieving low average delays under all but the highest load. Under higher loads, the performance of MCS deteriorates due to the additional blocking it induces. On average the percentage of blocking generated by MCS is 0.9%. The matching algorithms (FMA and ESA) generate 0.02% blocking (due to natural blocking in the demand matrices). When the load is high, FMA2 assigns more time slots to the heavier connections, which can use the extra time slots more efficiently. ESA assigns the same number of extra time slots to each connection irrespective of its load. In this scenario only the slot-by-slot scheduling algorithm experiences packet loss (up to $0.31\%$ for loads exceeding 70% of capacity).

We also 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 equal to the full link capacity (10 Gbps). The rate distribution is exponential. Figure 3 depicts queuing delays and packet losses for the FMA2, ESA and MCS algorithms. FMA2 demonstrates marginally superior average queuing delay performance (0.3-0.9 msec less when the load exceeds 50%).

We explored the performance of our algorithms using 50 seconds of packet traces captured from an OC3 link at Colorado State University [21]. The flows were divided into 16 components based on IP source/destination addresses, and each component served as one of the edge nodes. We considered a frame of length 0.1 seconds (equal to 100 time slots of 1 msec.). The average offered load was around $40\%$. The derived traffic is such that the demand is inadmissible for a duration of 10 seconds (from 2–12 seconds), because one of the edge nodes is overloaded. Growth in the queue sizes is unavoidable during this period. Figure 4 shows the total number of queued packets at the edge nodes. FMA2 and MCS adapt to the variations of the arrivals in a very similar fashion, but FMA2 has a lower number of queued packets because it does not induce blocking.

## 5 Conclusion and Future Work

We have formulated the bandwidth allocation problem in the AAPN network as a scheduling problem with the objective of designing a schedule that achieves zero rejection for admissible demands and provides (weighted) max-min fair allocation of free capacity. We proposed two novel scheduling algorithms that achieve this task and analyzed their behaviour. In future work, we will assess how FMA and ESA behave when there are errors in the predicted demand matrix.

## 6 Appendix: Proof of Theorem 2

*Proof.* Let $u \in \{(i,j), 1 \leq i,j \leq N\}$ index the source-destination connections specified by the demand matrix. We index entities by iteration number of the while-loop in Alg. 1. For example, $\mathcal{A}_D(h)$ denotes the set of unmodified lines at the *start* of iteration $h$ of the algorithm.

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 (6); there is no subsequent modification. We have $\omega_u = G_\gamma(h)$ for all $u \in a_\gamma(h)$.

The adjustment at iteration $h$ leads to $\gamma$ being a bottleneck link (line) for $u \in a_\gamma(h)$, i.e., $\omega_z \leq \omega_u$ 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 $\omega_z$ are obtained from adjustments prior to iteration $h$.

Suppose that line $\beta$ has minimum $G_\ell$ at iteration $h+1$. Lines $\gamma$ and $\beta$ 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:

$$
\begin{aligned}
LS_\beta(h+1) &= LS_\beta(h) + D_k\omega_k && (9)\\
S_{a_\beta}(h+1) &= S_{a_\beta}(h) - D_k && (10)\\
G_\beta(h+1) &= \frac{L - LS_\beta(h) - D_k\omega_k}{S_{a_\beta}(h) - D_k} && (11)\\
&= \frac{S_{a_\beta}(h)G_\beta(h) - D_k\omega_k}{S_{a_\beta}(h) - D_k} && (12)\\
&\geq G_\gamma(h) && (13)
\end{aligned}
$$

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

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

A similar argument, replacing $G$ with $H$, demonstrates the max-min fairness achieved by ESA. □

## References


[1] L. Xu, H.G. Perros, and G. Rouskas, "Techniques for optical packet switching and optical burst switching," *IEEE Comm. Mag.*, vol. 39, 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] X. Liu, N. Saberi, M.J. Coates, and L.G. Mason, "A comparison between time-slot scheduling approaches for all-photonic networks," in *Proc. Int. Conf. on Inf., Comm. and Sig. Proc.*, Bangkok, Thailand, Dec. 2005.

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

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

[8] 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.

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

[10] 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.

[11] 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, pp. 109–116, Jan. 2000.

[12] 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.

[13] D. Marx, "Graph coloring problems and their applications in scheduling," in *Proc. John von Neumann PhD Students Conference*, Budapest, Hungary, 2003.

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

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

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

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

[18] 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.

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

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

[21] "Passive measurement and analysis (PMA) project," http://pma.nlanr.net/Traces/Traces/daily.