Constructing Service Matrices for Agile All-Optical Cores

Cheng Peng, Sofia A. Paredes, Trevor J. Hall and Gregor v. Bochmann
Centre for Research in Photonics, School of Information Technology and Engineering
University of Ottawa, Ottawa, Canada
{cpeng, sparedes, thall, bochmann}@site.uottawa.ca

Abstract

A semi-analytical method based on alternate projections on a linear vector space is used to construct a service matrix from a traffic matrix, where the traffic matrix represents the bandwidth requested by the edge nodes and the service matrix represents how the bandwidth will be distributed by the core of an optical star network that operates in a Time Division Multiplexing mode. The algorithm iterates over a mathematical expression of complexity $O(N^2)$, where N denotes the number of edge nodes. The complexity of the method is therefore $O(kN^2)$ where k denotes the number of iterations needed to converge. With N large enough one observes that k<<N and hence this expression tends to $O(N^2)$. Results show that the service matrices obtained with this projection method have very high measures of similarity to the original traffic matrix, with an average similarity greater than 95% for $N \geq 32$. The method is robust to inadmissible/bursty traffic and yields equal or improved delay performance in the optical network compared to other allocation methods.

1. Introduction

The term "agility" in optical networks has been described as the ability to deploy bandwidth on demand at fine granularities [1], which radically increases network efficiency and brings to the user much higher performance at reduced cost. One possible scheme to provide such agility in WDM networks is time division multiplexing (TDM) in the optical domain. In such a context, the optical switches along lightpaths must be scheduled to reconfigure every timeslot, or every few timeslots, to adapt to dynamic traffic demands. This paper focuses on the centrally-controlled Agile All-Photonic Network (AAPN) architecture [2] which has the virtue of avoiding some of the complexity associated with synchronization of more general optical networks. In contrast to current backbone networks, all-photonic networks have the property that both transmission and switching are performed in the optical domain. The absence of optical-electrical-optical (OEO) conversion leads to two important advantages: greatly increased capacity and transparency to data format and bit rate.

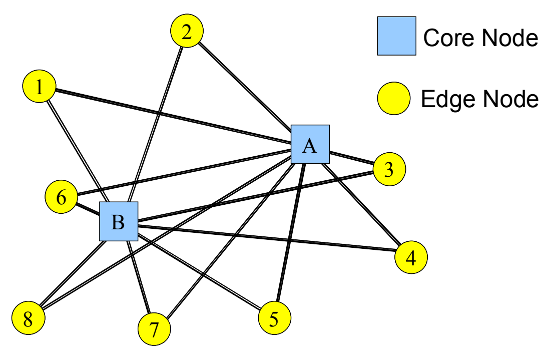

Figure 1. **The overlaid star architecture that characterises the AAPN.**

As shown in Figure 1, the AAPN consists of a number of hybrid photonic/electronic edge nodes connected together via a wavelength stack of buffer-less transparent photonic switches placed at the core nodes (a set of space switches, one switch for each wavelength), of which there is a small number. Each edge node contains a separate buffer for the traffic destined to each of the other edge nodes. Traffic aggregation is performed in these buffers, where packets are collected together in fixed-size slots (or, alternatively, bursts) that are then transmitted as single units across the network via optical links. The optical core network does not provide wavelength conversion or buffering. At the destination edge node, the slots are partitioned, with reassembly as necessary, into the original packets.

The overlay of several stars provides robustness in the case of link or core node failure, and also helps mitigate the potential for back-hauling. Each edge node is connected to every other edge node via a number of routes and data traversing the network only passes through one photonic switch, which results in a major simplification of the resolution of contention. From a control perspective, each star can be managed independently of the others because there is no data interaction.

2. Scheduling in a TDM-AAPN

When the AAPN operates in TDM mode, each edge node signals traffic demand information (e.g. from queue state information) to the core node along control channels before sending the slots of data. Once at the core, the global traffic demand is arranged in the form of a *traffic matrix*, with each element corresponding to a source-destination traffic flow. The controller uses this information to compute the *service matrix*, which defines the bandwidth allocated to each flow in units of slots within a frame; i.e., the larger the number of slots, the larger the bandwidth granted to that flow. The service matrix is then *decomposed* into its constituent *permutations*, each of which corresponds to a switch configuration for the duration of a timeslot. This schedule is signalled back to inform each edge node of the timeslots that it may use to transmit its traffic for each destination. The core wavelength switches are configured in coordination with the edge nodes according to the bandwidth allocated.

The slot allocation scheme to be used should possess two properties: the time complexity of the algorithm should be low enough to permit a practical implementation in the context of high-speed optical switching (the AAPN research network has specified a timeslot duration of 10 microseconds with a switch configuration overhead of less than 1 microsecond) and the scheme should handle non-uniform, bursty and, in general, dynamic traffic demands.

In the literature, many slot allocation methods such as Parallel Iterative Matching (PIM) [3], Iterative Round-Robin Matching (iRRM) [4], Iterative Round-Robin with SLIP (iSLIP) [5] and Dual Round-Robin Matching (DRRM) [6] have been proposed in the context of input-queued switches, which consist of a switching fabric equipped with buffers at its input ports. This work is relevant to AAPN since the star network formed by every wavelength space switch and the edge nodes attached to it can be viewed as a distributed Input Queued (IQ) switch (Figure 2) when there is no internal speed-up in the network (the use of speed-up requires buffers at the outputs). The general practice in these works is to find a switch configuration every timeslot such that it maximizes the volume of traffic that can be served by the switching fabric. The limitation of these schemes is that, for switches without internal speed-up, the input queues may become unstable under non-uniform and/or bursty traffic distributions since they result in *inadmissible* traffic matrices. *Admissibility* implies that the traffic does not oversubscribe the source or destination ports. Unfortunately, inadmissibility is prevalent in current networks.

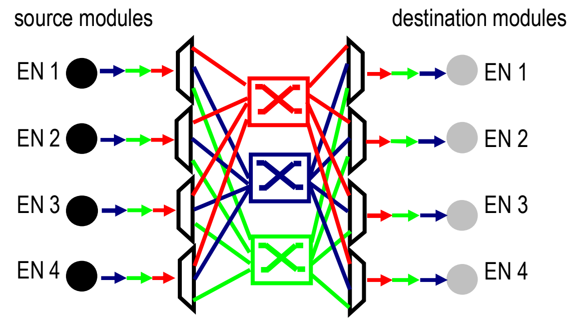

Figure 2. **AAPN with the edge node's (EN) modules logically split and the core node's wavelength space switches explicitly drawn.**

To handle non-uniform traffic, the Birkhoff-von Neumann (BvN) decomposition method to compute the scheduling in input-queued switches was introduced in [7] and [8]. This method produces $\sim N^2$ permutations with real valued expansion coefficients. If these coefficients are appropriately quantized as rational numbers then a periodic schedule may be used – the rational numbers being equated to the relative number of slots within a fixed length *frame*.

Whilst admission and congestion control mechanisms may be expected to enforce admissibility at the source and, in the long-term, at the destination; short term oversubscription of a destination can occur especially in the context of bursty traffic. It is therefore necessary to construct an admissible matrix, herein called a η *-server service matrix*, from potentially inadmissible estimates of the demand. A real/integer η *-server service matrix* $\mathbf{s}=(s_{ij})$ is defined as a real/integer matrix such that $s_{ij} \geq 0$ and $\sum_i s_{ij} = \eta$, $\sum_j s_{ij} = \eta$, $\eta > 0$, where η may be identified with the "frame size" in the integer case. The frame size determines the number of timeslots in a frame and therefore the maximum total number of slots that will be *served* from the set of queues in the edge node.

A simple rescaling method followed by quantization and an integer rate filling method is described in [9] to construct a service matrix in a Clos packet switch. Since the service matrix represents a regular bipartite graph, it is suitable for decomposition by an edge coloring algorithm. The limitation of the method is that some accuracy may be lost when the maximum row/column sum in a traffic matrix is very large in comparison to the others.

A weighted rate filling method is proposed in [10] to construct a 1-server service matrix from a real γ-server traffic matrix (normalized on a per slot basis such that $0 < \gamma \le 1$). This ad hoc method is not robust against inadmissible traffic demand. A max-min fairness method is presented in [11] to construct the service matrix by extending the notion of max-min fairness to take into account the traffic demand from each stream and giving priority to those streams that have the highest demand. This method can be modified to be robust to inadmissible traffic by permitting the first rescaling to *reduce* the value of the matrix elements (c.f. re-scaling as in [9]). The complexity of this method is $O(N^3)$, which is too complex to be practical. The maximum weight matching method proposed in [12] guarantees 100% throughput whilst minimizing delay, but is not practical to implement either since its complexity is $O(N^3 \log N)$.

Quality of Service in the context of this paper is an open question. The most important issue in this work is to construct a service matrix with the highest possible *similarity* to the corresponding traffic matrix, e.g. in terms of the coefficient of correlation, so that the bandwidth allocated by the core node reflects the traffic demand as closely as possible. The similarity measure does capture a notion of fairness.

3. Constructing a Service Matrix

A. The Overview

The traffic matrix $\mathbf{t}$ may be defined as an element in the Hilbert space $M \equiv R^{N\times N}$ of all $N\times N$ real-valued matrices equipped with the standard matrix inner product. The set of all matrices with all row and column sums equal ($\sum_i s_{ij} = \sum_j s_{ij}$) forms a subspace $S \subset M$. The set of all matrices with non-negative entries forms a subset $P \subset M$ that is a convex cone. The intersection $P \cap S \in M$ is therefore the convex cone of all possible service matrices. This motivates the application of Dykstra's [13] variant of the method of alternating projections on the convex sets S, P, to produce a sequence of matrices that converge to the projection of $\mathbf{t} \subset M$ onto the nearest service matrix $\mathbf{s} \in P \cap S$. An η-server service matrix may be found by suitably scaling any element of $P \cap S$ so that $\sum_i s_{ij} = \sum_j s_{ij} = \eta$.

B. Preliminary Concepts

Definition 1: The set of real valued $N\times N$ matrices $R^{N\times N} = \{ a_{ij} \mid a_{ij} \in R, \quad i,j = 1,N \}$ equipped with the standard inner product $(\mathbf{a},\mathbf{b}) = \sum_{i,j} a_{ij} b_{ij}$ is complete in the topology induced by the associated norm: $\|\mathbf{a}\| = (\mathbf{a},\mathbf{a})^{\frac{1}{2}}$ and is therefore a Hillbert space M. This fact provides meaning to the concepts of the distance between two matrices and their similarity.

Definition 2: The distance δ between two matrices $\mathbf{a},\mathbf{b} \in M$ is defined by: $\delta(\mathbf{a},\mathbf{b}) = \|\mathbf{a}-\mathbf{b}\|$

Definition 3: The similarity ρ, $0 \le \rho \le 1$, of two matrices $\mathbf{a},\mathbf{b} \in M$ is defined by:

$$\rho(\mathbf{a},\mathbf{b}) = \frac{(\mathbf{a},\mathbf{b})}{(\mathbf{a},\mathbf{a})^{1/2}(\mathbf{b},\mathbf{b})^{1/2}}$$

Note that the similarity is invariant to re-scaling of the matrices.

Definition 4: A closed convex set $K \subseteq M$ is characterized by the property that convex linear combinations of its elements are also members of the set: $\lambda\mathbf{a} + (1-\lambda)\mathbf{b} \in K \quad \forall\lambda \in [0,1] \quad \mathbf{a},\mathbf{b} \in K$

Given a closed convex subset of the Hillbert space $K \subseteq M$ and $\mathbf{a} \in M$, the projection $\mathbf{P}_K : \mathbf{a} \to \mathbf{b}$ onto K with $\mathbf{b} \in K$ is constructed by minimizing the distance $\delta(\mathbf{b},\mathbf{a})$; equivalently, the distance squared. The convexity of K guarantees the existence and uniqueness of the minimum.

Fact 1: Given a closed convex subset S of a Hilbert space M

$$S = \left\{ \mathbf{a} \in M \,\middle|\, \forall i, \sum_j a_{ij} = \frac{1}{N}\sum_{i,j} a_{ij}, \forall j, \sum_i a_{ij} = \frac{1}{N}\sum_{i,j} a_{ij} \right\},$$

the projection $\mathbf{P}_S : \mathbf{a} \to \mathbf{b}$ of $\mathbf{a} \in M$ onto S is given by $b_{ij} = a_{ij} - \frac{1}{N}\sum_j a_{ij} - \frac{1}{N}\sum_i a_{ij} + 2\frac{1}{N^2}\sum_{i,j} a_{ij}$. Proof of this statement is found in Appendix 1.

Fact 2: Given the closed convex subset of non-negative matrices $P = \{ \mathbf{a} \in M \mid a_{ij} \ge 0 \}$, the projection $\mathbf{P}_P : \mathbf{a} \to \mathbf{b}$ of $\mathbf{a} \in M$ onto P is given by

$$\begin{cases} b_{ij} = a_{ij} \ , a_{ij} \geq 0 \\ b_{ij} = 0 \ , a_{ij} < 0 \end{cases}$$

C. Alternating Projections Method to Construct a Service Matrix

Fact 1 implies that a traffic matrix may be mapped to *the nearest* matrix whose row and column sums are all equal by the projection operator $\mathbf{P}_S$. The "nearest" means that the distance between the two matrices is a minimum. However, Fact 1 does not guarantee that the entries of the resultant matrix are all non-negative and therefore it is not necessarily a valid service matrix. Fact 2 implies that the operator $\mathbf{P}_P$ projects a matrix to the nearest non-negative matrix but it does not guarantee that the resulting matrix possess equal row and column sums. In order to find a service matrix with both qualities, the method of alternating projections onto convex sets may be applied; that is, the two projections $\mathbf{P}_S$ and $\mathbf{P}_P$ are applied alternately. Applying Dykstra's variant of the alternating projections algorithm ensures that the limit of the resulting sequence is also the valid service matrix nearest to the initial matrix $\mathbf{t}$.

<u>**Algorithm**</u>:

Input: A traffic matrix $\mathbf{t} \in M$, the residual error ε and η

1. $\mathbf{t}^-_{prev} \leftarrow 0; \quad \mathbf{t}^+ \leftarrow \mathbf{t}$
2. $\mathbf{P}_S : \mathbf{t}^+ \rightarrow \mathbf{t}$;
3. $\mathbf{t} \leftarrow \mathbf{t} + \mathbf{t}^-_{prev}$
4. $\mathbf{P}_P : \mathbf{t} \rightarrow \mathbf{t}^+$
5. $\mathbf{t}^-_{prev} \leftarrow \mathbf{t} - \mathbf{t}^+$
6. If $\exists i \quad \sum_i t^+_{ij} \Big/ \left((1/N) \sum_{i,j} t_{ij} \right) \geq 1 + \varepsilon$ or $\exists j \quad \sum_j t^+_{ij} \Big/ \left((1/N) \sum_{i,j} t_{ij} \right) \geq 1 + \varepsilon$ go to step 2
7. Output $\mathbf{s} \leftarrow \left[\eta \Big/ \left((1/N) \sum_{i,j} t_{ij} \right) \right] \times \mathbf{t}^+$

Concretely, a traffic matrix $\mathbf{t}$ can be decomposed into two matrices, its positive part $\mathbf{t}^+$ and its negative part $\mathbf{t}^-$ such that $\mathbf{t} = \mathbf{t}^+ + \mathbf{t}^-$. For $\mathbf{t}^+$, the projection $\mathbf{P}_S$ can be applied to obtain a matrix $\mathbf{t}$ with row sums equal to column sums. The new traffic matrix $\mathbf{t}$ for the next iteration can be obtained then by applying $\mathbf{P}_P$ to $\mathbf{t} + \mathbf{t}^-$, where $\mathbf{t}^-$ is the accumulation of negative entries obtained from projections in previous iterations.

Note that the $\mathbf{t}^+$ obtained is a $\left((1/N) \sum_{i,j} t_{ij} \right)$-server service matrix with certain precision error: that is the row sums and column sums of the $\mathbf{t}^+$ are not exactly the same as $\left((1/N) \sum_{i,j} t_{ij} \right)$. The residual error therefore measures how far away the projection $\mathbf{t}^+$ is from being a valid service matrix (i.e. positive entries with equal row and column sums). The procedure may be repeated until the residual error is less than ε. Convergence is proven by the general result found in reference [13]. A η-server service matrix $\mathbf{s}$ with the same relative precision may be formed by scaling each entry of the $\mathbf{t}^+$ by $\eta \Big/ \left((1/N) \sum_{i,j} t_{ij} \right)$.

4. Results and Discussion

In this section, two key issues are discussed. The first issue is the time complexity of the method and the second is how close the constructed service matrix is to the initial traffic matrix.

To obtain the results shown in sections 4.A and 4.B, the traffic used is an aggregation of Pareto-distributed ON-OFF processes for each source as described in [9]. The generated traffic matrices are the accumulation of traffic over η timeslots and they may be inadmissible. The offered load is 80%. Yim's max-min fairness method [11] is adopted as a comparison in this paper because it also targets similarity and fairness and a simple modification makes it robust to inadmissible traffic. Yim's min-max fairness method has a time complexity of $O\left(N^3\right)$.

A. Time Complexity

Since both transformations $\mathbf{P}_S$ and $\mathbf{P}_P$ have a complexity of $O\left(N^2\right)$, the overall time complexity of our projection method is $O\left(kN^2\right)$, where k denotes the number of iterations required to achieve a target residual error. It is therefore important to ensure that k is smaller than N and make k as small as possible.

Figure 3 shows the similarity of the projection versus iterations. The similarity decreases because, as the algorithm converges, $\mathbf{t}^+$ is moving away from the most similar matrix without restriction to the most similar matrix that is also a valid service matrix (i.e. positive entries with equal row and column sums).

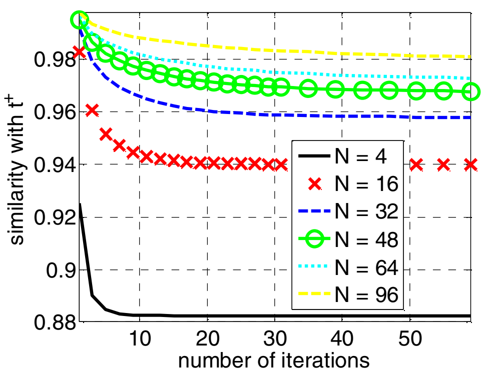

Figure 3. **Similarity of real valued projection versus number of iterations**

Figure 4 shows the number of iterations required for the algorithm to converge. Two different convergence tests have been used and the residual error ε has a different meaning for each test. For the *similarity test*, we use **Definition 3** as a test of convergence (in place of step 6) and ε indicates the difference in similarity with respect to the previous iteration. Since the projection will be quantized to obtain an integer service matrix, for the *valid service test* (step 6) the value of ε may be as high as 0.5 which guarantees that the quantized bandwidth demand never exceeds the maximum link bandwidth.

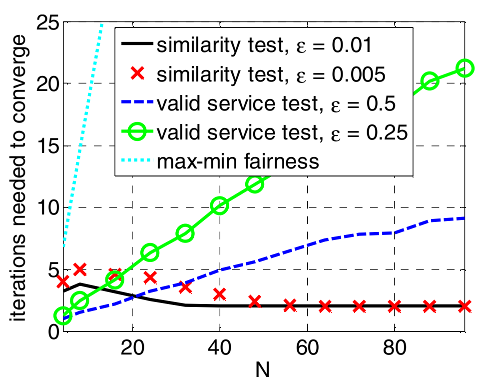

Figure 4. **Average number of iterations required for convergence versus switch dimension *N***

When N is large, k is much smaller than N in both test cases and therefore we can say that it only contributes $O(1)$ to the overall asymptotic complexity of the method. The time complexity of the projection method therefore tends to $O(N^2)$. The method was also tested for uniform traffic, resulting in $k \to 1$ as N increases.

B. Similarity Issue

The similarity issue can be formulated as follows. Given η servers, how does the number of ports N affect the similarity (definition 3) between the η-server service matrices and the original traffic matrices? If the two matrices were exactly the same, the similarity between them would be 1.

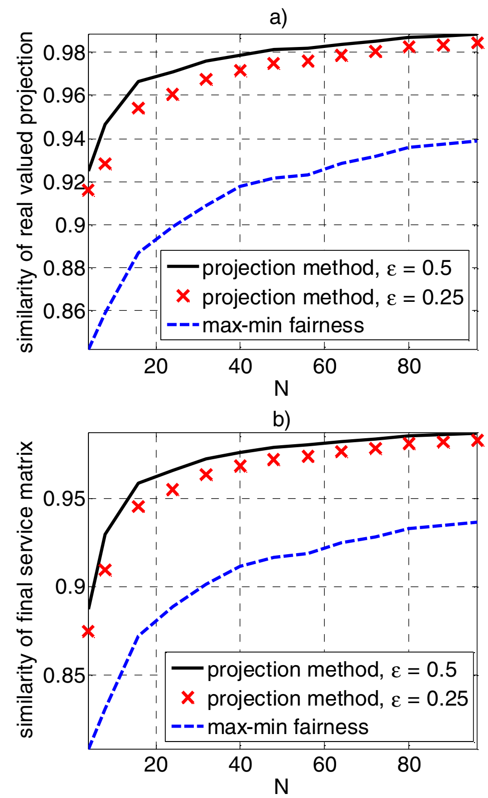

Figure 5. **Similarity versus switch dimension for η=2*N*, a) real valued projection , b) final service matrix**

Figure 5a shows the similarity of the real valued projection scaled to η (η=2N) versus N obtained with the max-min fairness and the projection methods. Figure 5b shows the similarity of the final η-server *integer* service matrix.

To obtain the final service matrix, the first step is quantization to integer numbers. Some slots will remain unassigned as a result of the fractions discarded with quantization. These slots will be distributed using a simple "rate-filling" algorithm that will assign the available slots to those flows with the largest discarded fractions (by incrementing the corresponding entry in S), provided that the associated row sum and column sum do not thereby exceed the maximum number of servers η.

The similarity obtained with the projection method after quantization and rate-filling (Figure 5b) is higher than that obtained with the max-min fairness method.

C. Network Performance

Figure 6 shows the delay performance of an AAPN for the two service matrix construction methods with a frame size $\eta = 2N$.

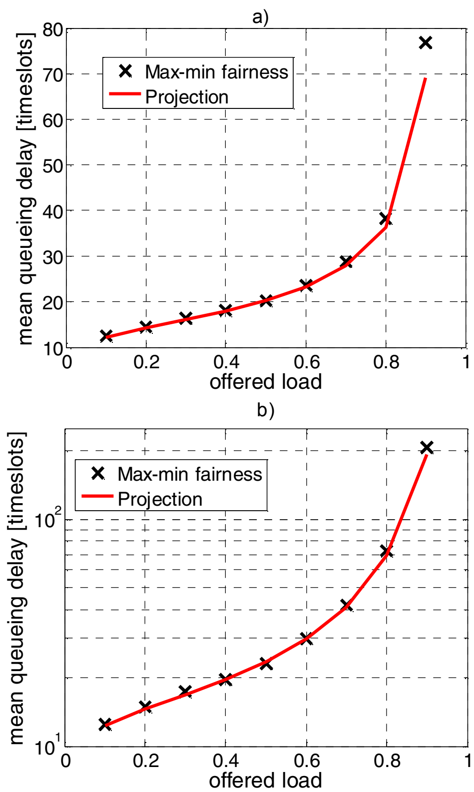

Figure 6. **Mean queuing delay in an AAPN versus the offered load for the projection and the max-min fairness methods. a) Metro-access scenario, b) Long-haul scenario**

The network simulated is one AAPN star (one wavelength) with 10Gbps links and 16 edge nodes. The traffic is generated with a composite Markov Poisson Process model [14] that simulates statistics of the following types of traffic: continuous bit rate (e.g. voice), variable bit rate (e.g. video) and variable bursty traffic (data). The method used to decompose the integer service matrix into η permutations is the "EXACT" method described in [15].

Figure 6a corresponds to the metro-access scenario (edge-core distances are 20 km) and Figure 6b corresponds to a long-haul scenario (edge-core distances are 2000 km). This figure shows that the projection method performs slightly better than the max-min fairness method in addition to having the advantage of being faster to compute.

5. Conclusions and Remarks

The bandwidth distribution in an AAPN is calculated in the form of a service matrix by an alternating projections method with complexity $O(kN^2)$, where N denotes the number of edge nodes and k denotes the number of iterations needed to converge. As N increases it is observed that $k \ll N$ and therefore the complexity tends to $O(N^2)$. The service matrix exhibits high measures of similarity (in terms of correlation) to the original traffic matrix even after quantization and therefore the bandwidth allocated in the optical network successfully adapts dynamically to the traffic demands.

The method is robust to inadmissible/bursty traffic and results show that it yields improved delay performance compared to the max-min fairness method while the complexity is lower.

Figure 4 and Figure 5 imply that the projection method can generate η-server service matrices with high similarity after only a few iterations. With this in mind, it may be possible to simplify the projection method to only a few iterations ($k \approx 10$) without performing the test for convergence, which would improve the total computation time in practice.

It is important to note that this method can also be used with the schedule applied along wavelengths, in which case the frame size would be equal to the number of wavelengths available in the core node and a new frame schedule would be applied every timeslot (or every few timeslots as in [9]). This variation would be more suitable in the presence of persistent high-rate flows that would make efficient use of the high bandwidth provided by a whole wavelength.

Acknowledgement

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada and industrial and government partners, through the Agile All-Photonic Networks (AAPN) Research Network. Dr. Trevor J. Hall holds a Canada Research Chair in Photonic Network Technology at the University of Ottawa and he and Dr. Sofia Paredes are grateful to the Canada Research Chairs Programme for their support.

References

[1] G. v. Bochmann, M.J. Coates, T.J. Hall, L. Mason, R. Vickers and O. Yang, "The Agile All-Photonic

Network: An architectural outline", Proc. Of the 22nd Biennial Symposium on Communications, Queen's University, Kingston, Ontario, Canada, April, 2004, pp.217-218.
[2] The Agile All-Photonic Networks (AAPN) Research Network, http://www.aapn.mcgill.ca/, 2003-2005
[3] T. Anderson, S. Owicki, J. Saxe and C. Thacker, "High Speed Switch Scheduling for Local Area Networks", ACM Transactions on Computer Systems 11(4), November 1993, pp. 319 – 352.
[4] N. McKeown, P. Varaiya and J. Warland, "Scheduling cells in an input-queued switch", IEEE Electron. Letter, December 1993, pp.2174-2175.
[5] N. McKeown, "iSLIP: A scheduling algorithm for input-queued switches", IEEE/ACM Transactions on Networking, 7(2), April 1999, pp. 188-201.
[6] H. J. Chao, "Saturn: a terabit packet switch using dual round-robin", IEEE Communication Magazine, vol. 38, no. 12, December 2000, pp. 78-84.
[7] Cheng-Shang Chang, Wen-Jyh Chen, Hsiang-Yi Huang, **"**Birkhoff-von Neumann input-buffered crossbar switches for guaranteed-rate services**",** IEEE Transactions on Communications, 49(7):1145-1147, July 2001.
[8] P. Giaccone, B. Prabhakar and D. Shah, "Towards simple, high performance schedulers for high-aggregate bandwidth switches", Proceedings of IEEE INFOCOM, 2002, pp. 1160-1169.
[9] S.A. Paredes, T.J. Hall, "Flexible bandwidth provision and scheduling in a packet switch with an optical core". OSA Journal of Optical Networking, 4(5), May 2005, pp. 260-270.
[10] J. Li, N. Ansari, 'Enhanced Birkhoff-von Neumann decomposition algorithm for input queued switches', IEE Proceedings – Communications, 148(6), 2001, pp. 339-342.
[11] R Yim, N. Devroye, V. Tarokh and H.T Kung, "Achieving Fairness in Two-Dimensional Generalized Processor Sharing", Proc. Of the 22nd Biennial Symposium on Communications, Queen's University, Kingston, Ontario, Canada, April, 2004, pp.185-187.
[12] N. Mckeown, A. Mekkittikui, V. Anantharam and J. Walrand, "Achieving 100% throughput in an input-queued switch", IEEE Transactions on Communications, 47(8), 1999, pp.1260-1267
[13] J.P. Boyle, R.L. Dykstra, "A method of finding projections onto the intersection of convex sets in Hilbert spaces", Advances in Order Restricted Statistical Inference, Lecture Notes in Statistics, Vol. 37, Springer, Berlin, Germany, 1986, pp. 28-47.
[14] M. Schwartz, "Broadband Integrated Networks", Prentice Hall, New Jersey, USA,1996, pp. 21-32.
[15] B. Towles, W.J. Dally, "Guaranteed scheduling for switches with configuration overhead", IEEE/ACM Trans. on Networking, 11(5), Oct. 2003, pp.835-847.

Appendix 1

Proof: By the method of Lagrange multipliers, the computation proceeds by first minimising the cost:

$$C(b_{ij}) = \frac{1}{2}\sum_{i,j}\left(b_{ij} - a_{ij}\right)^2 + \sum_{i,j}\alpha_i b_{ij} + \sum_{i,j}\beta_j b_{ij} - \left(\frac{1}{N}\sum_{i,j} b_{ij}\right)\left(\sum_i \alpha_i + \sum_j \beta_j\right)$$

where α_i, β_j are Lagrange multipliers that are subsequently found from the constraint $\mathbf{b} \in S$.

Now:

$$\frac{\partial C}{\partial b_{ij}} = 0 \Rightarrow b_{ij} = a_{ij} - \left(\alpha_i - \frac{1}{N}\sum_i \alpha_i\right) - \left(\beta_j - \frac{1}{N}\sum_j \beta_j\right)$$

equivalently:

$$b_{ij} = a_{ij} - p_i - q_j$$
$$\sum_i p_i = 0, \sum_j q_j = 0$$

now:

$$\sum_i b_{ij} = \sum_i a_{ij} - Nq_j = \frac{1}{N}\sum_{i,j} b_{ij},$$
$$\sum_j b_{ij} = \sum_j a_{ij} - Np_i = \frac{1}{N}\sum_{i,j} b_{ij}$$
$$\Rightarrow p_i = \frac{1}{N^2}\sum_{i,j} b_{ij} - \frac{1}{N}\sum_j a_{ij}, q_j = \frac{1}{N^2}\sum_{i,j} b_{ij} - \frac{1}{N}\sum_i a_{ij}$$

but:

$$\sum_{i,j} b_{ij} = \sum_{i,j} a_{ij}$$

hence: $b_{ij} = a_{ij} - \frac{1}{N}\sum_j a_{ij} - \frac{1}{N}\sum_i a_{ij} + 2\frac{1}{N^2}\sum_{i,j} a_{ij}$