Frame-Based Bandwidth Allocation for Agile All-Photonic Networks

Cheng Peng

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the PhD degree in name of program

Computer Science

Ottawa-Carleton Institute for Computer Science
School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa

University of Ottawa

Abstract

Frame-Based Bandwidth Allocation for Agile All-Photonic Networks

by Cheng Peng

The term “agility” in optical networks describes the ability to deploy bandwidth on demand at fine granularity that allows carriers to provision and deploy services rapidly. An overlaid star all-photonic WDM network, called the Agile All-Photonic Network (AAPN), can provide such agility through multiplexing over each wavelength in the time domain. The AAPN consists of a number of hybrid photonic/electronic edge nodes connected together via a number of optical core nodes. Each of the core nodes consists of a number of buffer-less transparent photonic space switches, one for each wavelength, and links to all edge nodes via optical fibers. Each edge node aggregates traffic in buffers, segments it as slots (i.e. units), and transmits the slots in photonic form across the network. For bandwidth sharing, the core nodes are required to perform reconfiguration periodically according to the traffic demand information sent by the edge nodes.

In this thesis, it is studied how the core nodes distribute the bandwidth to adapt to the traffic in the edge nodes and how the edge nodes perform routing in the AAPN so that the traffic is less congested and can reach the destination with short delay.

For the former problem, the alternating projections method and the quick Birkhoff-von Neumann decomposition method are proposed which provide a simple and efficient bandwidth allocation scheme for the AAPN core nodes.

For the second problem, a routing method is proposed which provides a trade-off between the delay and the blocking probability, i.e. the method improves the delay performance of the network dramatically under the low load while not affecting the blocking performance at high load.

TABLE OF CONTENTS

LIST OF FIGURES

Number *Page*

LIST OF TABLES

Number *Page*

ACRONYM

AAPN	*Agile All-Photonic Network*
BvN	*Birkhoff-von Neumann*
CCF	*Critical Cell First*
CIOQ	*Combined Input/Output Queued Switch*
CoS	*Classes of Service*
DiffServ	*Differentiated Service*
DRRM	*Dual Round-Robin Matching*
FIRM	*FCFS (first-come-first-serve) in Round-Robin Matching*
GSA	*Gale/Shapley algorithm*
GLJD	*Greedy Low Jitter Decomposition*
HOL	*Head of Line*
IntServ	*Integrated Service*
IQ	*Input Queued Switch*
iRRM	*Iterative Round-Robin Matching*
iSLIP	*Iterative Round-Robin with SLIP*
JPM	*Joined Perferred Matching*
LE-PIM	*Logical Equivalence of Parallel Iterative Matching*
LOOFA	*Lowest Output Occupancy First Algorithm*
LPF	*Longest Port Firs*
LQF	*Longest Queue First*
OBS	*Optical Burst Switch*
OCF	*Oldest Cell First*
OEO	*Optical-Electrical-Optical*
OQ	*Output Queued Switch*
PIM	*Parallel Iterative Matching*
PSDN	*Public Switched Data Network*
PSTN	*Public Switched Telephone Network*
RSNE	*Reverse Subtree Neighborhood Exploration*
RSVP	*Resource Reservation Protocol*
QBvN	*Quick BvN Decomposition Algorithm*
QoS	*Quality of Service*
QoS-FBAS	*QoS Quaranteed Frame-based Bandwidth Allocation*
SMA	*Simple Moving Average*
TDM	*Time Division Multiplex*
VOQ	*Virtual Output Queue*
WDM	*Wavelength Division Multiplex*

ACKNOWLEDGMENTS

I would like to express my deep gratitude to my supervisors, Dr. Gregor v. Bochmann and Dr. Trevor J. Hall, for their guidance, support and encouragement throughout my study at the University of Ottawa.

I would like to specially thank Dr. Sofia Paredes, Dr. Jun Zheng and Peng He for their insightful comments to and countless discussions on this thesis work.

I am also indebted to Peng Zhang, Guangrui Pan, Ying Qiao, Dr. Bin Zhou, Anna Agusti and Dr. Libo Zhong and colleagues in the AAPN project and the Distributed Systems Research group for their help and support.

I would also like to thank Shuyan my mother, Jishi my father, Yuan my sister, Weihong my sister in law and my super cute niece Mingxi. I love you all infinitely.

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.

1. Introduction

1.1. Problem Statement

We consider an overlaid-star all-photonic WDM network, called the Agile All-Photonic Network (AAPN) [Boch2004] [Hall2005] [Mason2006], with the ability to deploy bandwidth on demand at fine granularity through multiplexing in the time domain, which is based on the principle of time division multiplexing (TDM) [Boch2004].

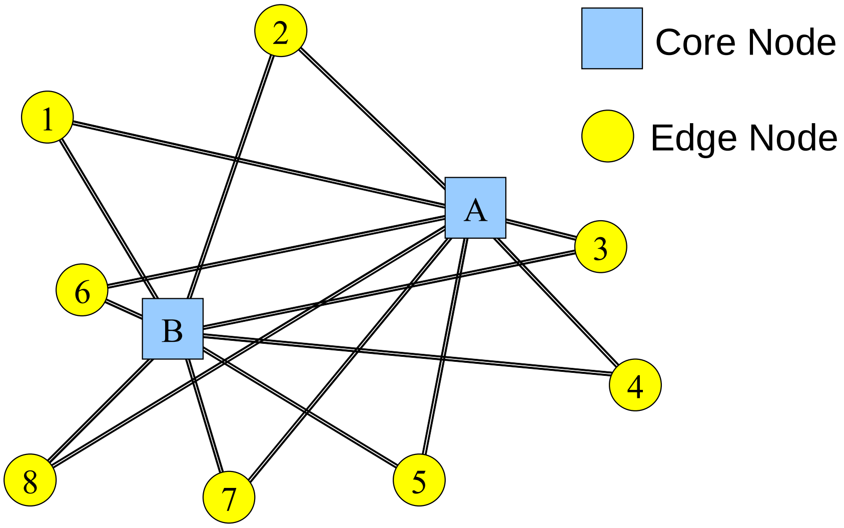


Figure 1.1: The AAPN networks

The AAPN (as shown in Figure 1.1.1) consists of a number of hybrid photonic/electronic edge nodes connected together via a number of optical core nodes. Each of the core nodes consists of a number of buffer-less transparent photonic space switches, one for each wavelength, and links to edge nodes via optical fibers. Each edge node aggregates traffic in buffers, segments it as slots (i.e. units), and transmits the slots in photonic form across the network.

In contrast to current optical networks, the AAPN has the property that not only transmission but also switching are performed in the optical domain. The absence of optical-electrical-optical (OEO) conversion at the core nodes leads to two important advantages: greatly increased capacity and transparency to data format and bit rate.

For bandwidth sharing, these switches residing at the core nodes are required to perform reconfiguration periodically according to the traffic demand information sent by the edge nodes. In this thesis, it is studied how the core nodes distribute the bandwidth to adapt to the traffic in the edge nodes. For the overlaid star topology, the edge nodes must determine the lightpaths between source and destination. Consequently, it is worth studying how the edge nodes perform routing in the AAPN so that the traffic is less congested and can reach the destination with short delay.

1.2. Motivation and Objective

In the AAPN research project, the timeslot duration is specified as 10 microseconds with a configuration overhead of less than 1 microsecond. The designed link capacity is 10Gbps. In such a high-speed environment, the bandwidth adaptation, i.e. a change of schedule, should be done within 10 microseconds to a few milliseconds, which motivates us to find a bandwidth allocation scheme that possesses the following four properties:

- *Fast*: To achieve the bandwidth sharing with fine granularity, it is important that the scheme does not become the performance bottleneck. The scheme should therefore have a computation complexity as low as possible.

- *Simplicity*: To enable a practical implementation over fast optical switches, the scheme should be as simple as possible.
- *Efficiency*: An efficient bandwidth allocation scheme is one that maximizes the volume of traffic served by the switching fabric.
- *Stability*: For a given admissible traffic pattern, the scheme is said stable if the expected occupancy of the buffers at the edge nodes remains finite. The term "*admissibility*" implies that the traffic does not oversubscribe the source or destination ports.
- *No Starvation*: A non-empty input-queue is said to be starved if, for a given traffic pattern and scheduling algorithm, it remains unserved indefinitely.

With these objectives, we may partition the problem into two main components: the *scheduling* and the *routing*. The former is solved by a centrally-controlled scheduler residing at the core node where a simple and efficient timeslot allocation scheme runs for bandwidth sharing; the latter is solved by proposing a core selection scheme residing at each edge node where lightpaths are determined.

1.3. Contribution of the Thesis

There are five main contributions deriving from this thesis work.

1. A bandwidth allocation method, called the alternating projection method, is proposed. The bandwidth allocated in the optical network successfully adapts dynamically to the traffic demands. The method satisfies the efficiency, stability and no-starvation objectives outlined above.

2. A simple and efficient bandwidth configuration approach, called the quick Birkhoff-von Neumann decomposition method, is proposed which has the lowest time complexity compared with other known heuristic methods. The method satisfies the speed objective outlined above.
3. Concentrators/distributors [1] can be introduced in the AAPN to improve the scalability of the network. A bandwidth allocation scheme is proposed for the core node in the context of AAPN with concentrators.
4. An extension of the proposed bandwidth allocation scheme is in the context of QoS. This new scheme, called the QoS guaranteed frame-based bandwidth allocation scheme, is proposed to support guaranteed QoS.
5. A routing method is proposed by which the source edge node can select the shortest available path under low traffic load and balance the traffic when the traffic load is high. Compared to the well-known least-congested-path routing strategy, the method improves the delay performance of the network dramatically under the low load while not affecting the blocking performance at high load.

1.4. Outline of Thesis

The thesis is organized as follows. Chapter 2 gives a literature review of various switch architectures, scheduling schemes and network routing strategies. Chapter 3 on bandwidth allocation describes the approach used to construct the traffic demand matrix and the service matrix, where the alternating projection method is described and the performance and complexity assessed by discrete event simulation. Chapter 4 defines and compares the

[1] Refer to Section 2.1.2.2

quick Birkhoff-von Neumann decomposition method with existing alternatives and proposes a simple and efficient bandwidth configuration approach that can provide QoS guarantee. Chapter 5 addresses the load sharing or routing problem in AAPN topology and chapter 6 concludes and identifies some areas for further research.

2. Background

2.1. Architecture

2.1.1. Switch Architecture

Optical network infrastructure is facing the requirement to support and adapt to the growth of traffic in backbone networks. Advances in fiber throughput and optical transmission technologies have enabled operators to deploy dramatically increasing capacity. However, switching technologies have not advanced at the same pace and have become the bottleneck of network bandwidth.

For the switches working under TDM mode, the term “blocking” is used to describe the difficulty of setting up a circuit because the output port has been occupied. Since switches can only serve one slot per timeslot for each output port, it is necessary to store slots within switches while they wait for their destination port to become free. Hence, the switches usually adopt the architecture consisting of a switch fabric that sets up a connection between input and output ports and buffers that store slots according to a suitable queuing discipline.

2.1.1.1. Output-Queued Switch

The ideal switch architecture is the output-queued (OQ) switch that consists of a crossbar circuit switch equipped with buffers at its outputs that queue slots for transmission

(Figure 2.1). This architecture is considered ideal because the arriving slots can be switched to the corresponding output port without waiting at the input port. Hence, it provides the highest throughput (100%) and the lowest delay [Karol1987]. By "100% throughput" it is meant that the expected occupancy of every buffer is finite. However, the scalability of OQ switches is restricted due to a mandatory internal speed-up of the circuit switch by a factor of N for a $N \times N$ switch, which is necessary for OQ switches to handle the worst case scenario, i.e. slots at all N input ports contend for the same output port. The "speed-up" can be measured by the number of slots served from a single input or to a single output port in the same timeslot.

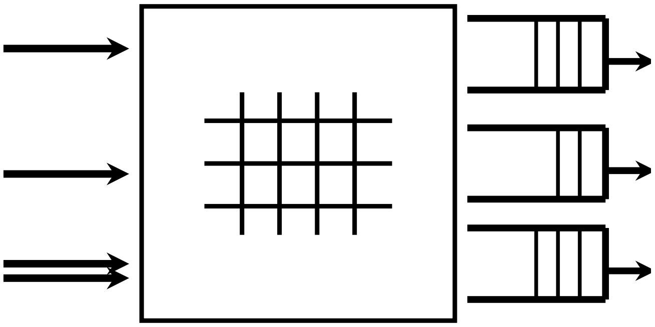


Figure 2.1: Architecture of output-queued switches

2.1.1.2. Input-Queued Switch

The scalability issue can be overcome by adopting Input-queued (IQ) switches that buffer slots at the input ports of the switch fabric only (Figure 2.2). In this case, a scheduling algorithm must be used to determine when the slots pass through the switch fabric without the blocking at the output port. Consequently, it is not necessary to run the crossbar faster than the input line speed.

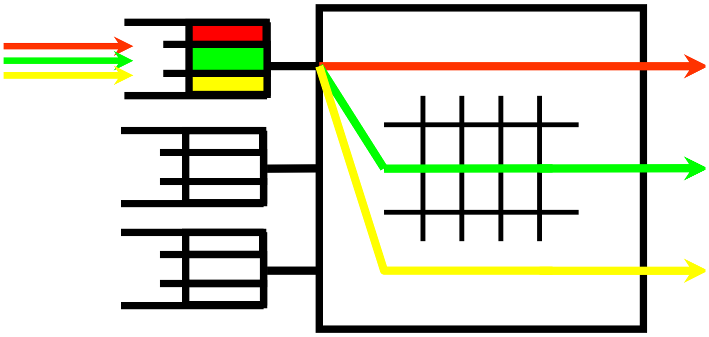


Figure 2.2: Architecture of input-queued switches

IQ switches suffer from inherently poor performance due to head-of-line (HOL) blocking. HOL blocking arises when the input buffer is arranged as a single first-in-first-out (FIFO) queue at each input port of the crossbar: a slot destined to a free output port may be held in line due to the slot ahead waiting for an output port that is not free. Even with independent identically-distributed (i.i.d.) Bernoulli arrivals, a very benign type of traffic, it is well known that the HOL blocking can limit the throughput to just $2-\sqrt{2}\approx 58.6\%$ [Karol1987] [Hluchyj1988]. Some technologies [Chen1994] [Karol1988] have been suggested for reducing HOL blocking, e.g., considering the first K slots in the FIFO queue instead of only the first one. However, these approaches are sensitive to traffic patterns and perform no better than regular FIFO queuing when the traffic is bursty [McKeown1995]. In fact, a simple solution to eliminate HOL blocking is by adopting so-called virtual output queuing (VOQ) [Tamir1988]. The basic idea is that an input port maintains a queue at each input port for each output port so that the slots in each VOQ have the same destination. Hence, the waiting of the head slot means the destination output port corresponding to the VOQ is busy. The use of VOQ with a suitable scheduler permits a 100% throughput for both uniform and non-uniform traffic [Mekkittikul1998] [McKeown1996]. However, a complex scheduling algorithm is necessarily employed to resolve contention at the input port of the switch fabric between slots destined to different outputs [Anderson1993].

2.1.1.3. Combined Input/Output -Queued Switch

As shown in Figure 2.3Figure, a switch may have buffers at both the input and output ports and is known as a combined input/output queued switch (CIOQ). In this case, the speed-up of the crossbar switch may be between 1 and N . The CIOQ architecture with a speed-up of two can emulate perfectly an ideal OQ switch, but the scheduling algorithm is too complex to be practical [Stoica1998] [Chuang1999].

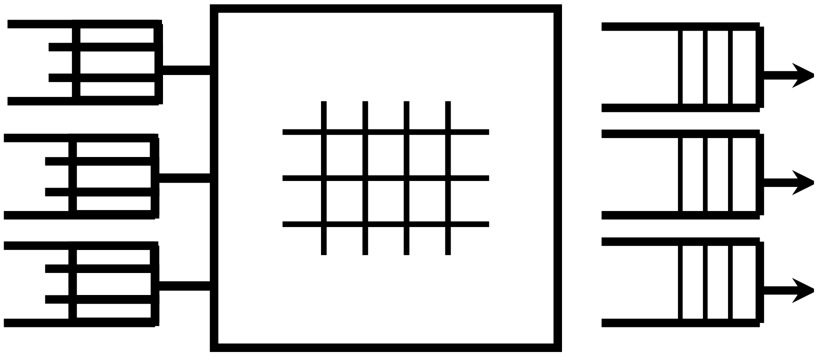


Figure2.3: Architecture of combined input/output queued switches

2.1.1.4. Summary

Table Table 2.1 summarizes the characteristics of the different switch architectures. In this comparison, it is shown that the OQ switch is not deemed practical due to the mandatory high speed-up. The CIOQ switch may gain the same performance as the OQ switch does by employing a speedup of two. The IQ switch, on the other hand, is not capable of emulating the OQ switch.

	OQ switch	**IQ switch**	**CIOQ switch**
Architecture	Switch fabric + buffers at output ports	Switch fabric + buffers at input ports	Switch fabric + buffers at both input and output ports
Blocking	No	No	No
Speed-up	N	1	Between 1 and N (e.g. 2)

Table 2.1 Comparison of different switch architectures

2.1.2. AAPN Architecture

2.1.2.1. AAPN without Concentrators / Distributors

As shown in Figure 2.4, the AAPN consists of a number of hybrid photonic/electronic edge nodes connected together via a core node that contains a stack of bufferless transparent photonic space switches one for each wavelength. An edge node contains a number of VOQs for the traffic destined to each of the other edge nodes. Traffic aggregation is performed in these queues, where packets are collected together in fixed-size slots that are then transmitted as single units across the network via optical links. At the destination edge node the slots are partitioned, with reassembly as necessary, into the original packets.

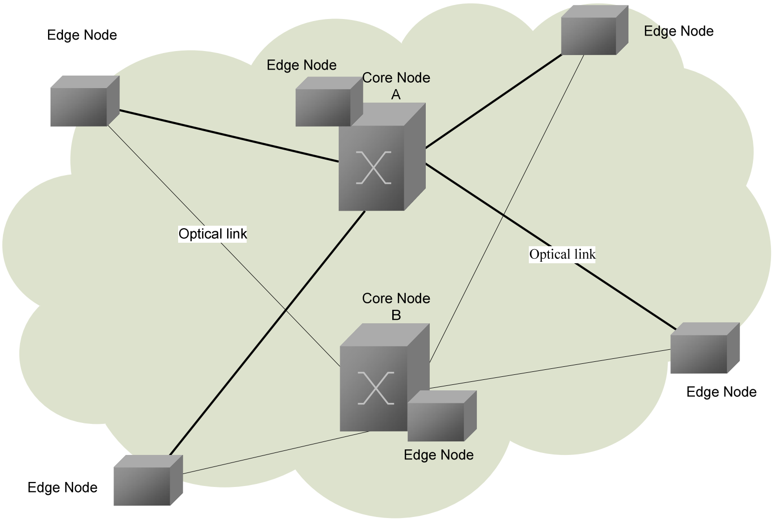


Figure 2.4: AAPN Star Topology without Concentrators

In the AAPN, one of the edge nodes is co-located with the core code, where a scheduler is equipped. The scheduler is used to dynamically allocate timeslots over the various wavelengths to each edge node. When the AAPN operates in TDM mode, each edge node signals a bandwidth request to the edge node co-located with the core (i.e. the scheduler) along control channels before sending the slots. The scheduler at the core node allocates timeslots to each edge node over an appropriate wavelength based on the request. The schedule[1] is signalled back to inform each edge node of the timeslots that it may use to transmit its traffic for each destination, and the core switches are re-configured in coordination with the edge nodes according to the bandwidth allocated.

2.1.2.2. AAPN with Concentrators/Distributors

An issue for the AAPN is the scalability that is limited by a restriction on the number of ports of the core switches. The restriction is hard to overcome due to the physical dilemma between switch speed and large port counts. In order to improve the scalability, a device, called *concentrator/distributor*, is employed [Maier2002] [Kamiyama2005]. As shown in Figure 2.5, a concentrator/distributor in AAPN consists of two devices: an optical *combiner* and *splitter*. The combiner is used to aggregate upstream (from edge nodes to core nodes) traffic in the optical domain. The splitter is used to distribute a downstream (from core nodes to edge nodes) optical signal to an edge or a number of edges. The concentrator may be realized with *active* or *passive* optical devices.

[1] The term "schedule" used in this work denotes the switch descriptor that contains the switch state information, i.e. the connection time and pattern between input and output ports. For example, the schedule may include the following information, 'input port 0 will be connected with output port 7 at the 80th timeslot'.

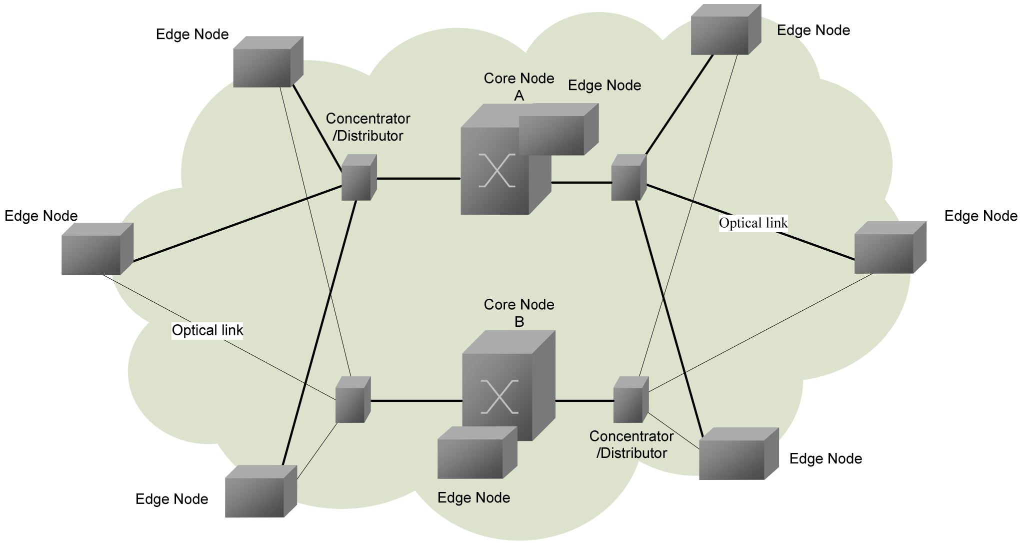


Figure 2.5: AAPN overlaid star topology with concentrators

For passive *optical combiners* and *splitters*, data is broadcast by the distributors to all edge nodes in the downstream direction. In the upstream direction, data is merged onto the same fiber; each edge node can transmit data only during the timeslots granted to them. The timeslots are synchronized so that transmission slots from different edge nodes do not collide.

When the AAPN with passive concentrators/distributors operates in TDM mode, each edge node signals a bandwidth request to the core via a concentrator in sequence for blocking avoidance before sending the slots. The scheduler at the core allocates timeslots to each edge node over an appropriate wavelength based on the request. The schedule is signalled back to inform each edge node of the timeslots that it may use to transmit its traffic for each destination, and the core wavelength switches are re-configured in coordination with the edge nodes according to the bandwidth allocated. Due to the broadcast feature of passive distributors in the downstream direction, the core also needs to inform each destination edge-node of the timeslots that it may use to receive its traffic.

The concentrators/distributors may also be realized with active selector switches [Mason2005]. In the downstream direction, data is sent to its destination edge node by the selector switches. Hence, global synchronization is required across wavelengths, i.e. the selector switches need to know exactly when the data arrive and where to send. In the upstream direction, the selector switch merges different wavelengths into fibres.

2.1.2.3. Summary

As shown in Figure 2.6, the architecture of AAPN is analogous to that of IQ (without speedup) or CIOQ (with speedup) switches where the switch fabric and buffers are distributed to different places and connected by optical fibres. Therefore, the scheduling algorithms that apply to the IQ switches in the literature may be used to inspire the design of the scheduling algorithms in the AAPN. However, the existence of propagation delays will degrade delay performance compared to localized switches.

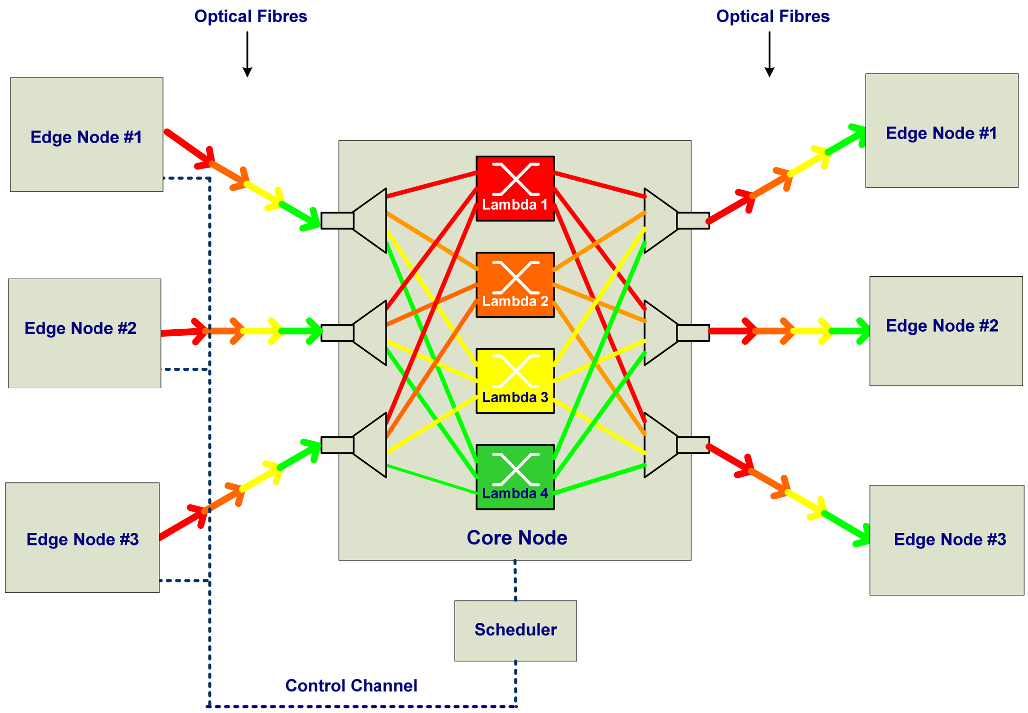


Figure 2.6: The AAPN with one core node

2.2. Bandwidth Sharing Schemes

There are two kinds of bandwidth sharing schemes, i.e. optical burst switching (OBS) [Yoo1999] and TDM, which may be adopted in the AAPN.

2.2.1. Optical Burst Switching

The OBS exploits the large available bandwidths by reducing the electronic processing of optical packets. Data packets are aggregated into much larger bursts (alternatively, slots) at the edge of the network. Each burst is transmitted after the corresponding control packet (requesting the necessary resources at each intermediate node) is sent on a separate control channel. If the required capacity can be reserved, the burst can successfully pass through the core nodes; otherwise it will be dropped. After the burst transmission, the channel is released allowing for other burst transmissions to use it. This statistical multiplexing of the channel among multiple bursts improves the backbone efficiency and offers excellent scalability.

However, data loss due to burst contention is a major concern [Liu2005a]. Contention occurs when two or more bursts arrive via the same wavelength at a given output port of a switch at the same time. Several mechanisms have been proposed to reduce burst contention: fiber delay lines [Turner1999], wavelength conversion [Turner1999], deflecting routing [Chen2003], burst segmentation [Maach2002] or retransmission [Agusti2005]. Nevertheless, the AAPN assumes that there exist neither fibre delay lines, nor deflection routing, nor wavelength converters. It is shown in [Agusti2005] that the OBS has a poor delay performance under uniformly distributed Poisson traffic when the load is greater than 0.6 and the network retransmits the dropped bursts. Though burst segmentation does not need retransmission and thus improves the delay performance at high load, the data loss is too high to be used in practice.

2.2.2. Time Division Multiplexing

The TDM scheduling algorithms can be classified into two categories: slot-by-slot allocation and frame-by-frame allocation (Figure 2.7). The former essentially calculates a new switch configuration for each timeslot and the switch fabric is updated to change the interconnections between input and output ports every timeslot. The latter groups a number of timeslots together to form a frame and switch configurations are only updated when necessary (i.e. a request for setting up or releasing a connection arrives, c.f. SONET). In this thesis, the concept of the frame-by-frame allocation is somewhat different: a number of switch configurations for a given frame are calculated based on traffic demand and these configurations are updated every frame.

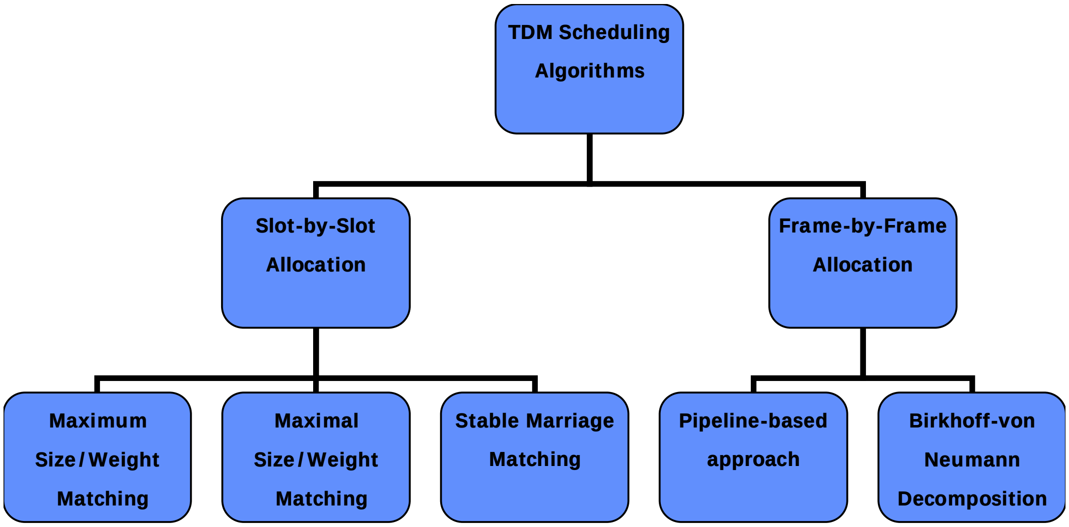

TDM Scheduling Algorithms
Slot-by-Slot Allocation
Frame-by-Frame Allocation
Maximum Size/Weight Matching
Maximal Size/Weight Matching
Stable Marriage Matching
Pipeline-based approach
Birkhoff-von Neumann Decomposition

Figure 2.7: TDM scheduling algorithms

The difference between the slot-by-slot allocation and the frame-by-frame allocation lies in:

- The slot-by-slot allocation has to signal core nodes every timeslot. The frame-by-frame allocation does not need to do so, but an extra average delay, equal to half the frame size, is introduced. The slot-by-slot allocation method outperforms the frame-by-frame allocation method in terms of delay (but only half the frame size).

- The frame-by-frame allocation method could adjust the order of its generated schedules to fulfill additional requirements, e.g. to make switches re-configure as seldom as possible. The slot-by-slot allocation cannot achieve this.

- It is reported in [Liu2005b] that, for link distances larger than approximately 600km, frame-by-frame allocation may produce marginally smaller end-to-end delay than slot-by-slot scheduling[1].

The slot-by-slot allocation method can be modeled as the bipartite graph matching problem, i.e. finding a matching for each timeslot according to a bipartite graph. A *bipartite graph* $G = (V_i, V_j, E)$ is a mathematical structure consisting of two disjoint vertex sets, V_i and V_j, identified here with input and output switch ports respectively, and one edge set E, where each edge joins one vertex from each vertex set. The vertices $u \in V_i$ correspond to the input ports of switches; the vertices $v \in V_j$ correspond to the output ports of switches; and the edges $(u, v) \in E$ represent the interconnections between them (Figure 2.8).

[1] This result is achieved based on Poisson traffic, with 90% offered traffic load, for a single high quality, best effort transport service class.

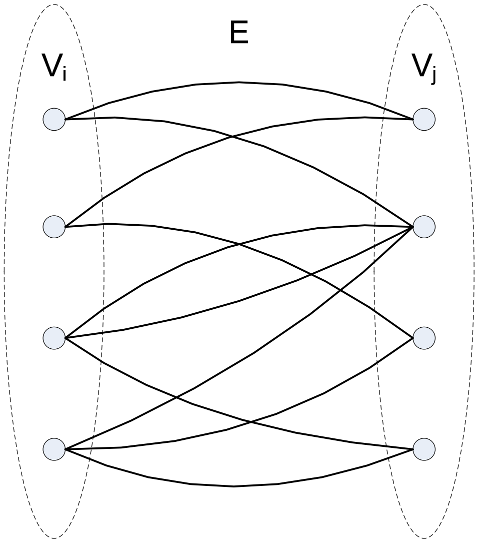


Figure 2.8: A bipartite graph G

A *bipartite graph matching* G_M is a sub-graph of G such that no more than one edge is incident on any vertex. A bipartite graph matching G_M is said *perfect* if there is an edge incident to every vertex. (Figure 2.9). The bipartite graph matchings represent the configurations of a switch, i.e., the connectivity pattern between input and output ports without contention.

The slot-by-slot allocation algorithms can be classified into three categories, i.e. maximum size/weight matching, maximal size/weight matching and stable marriage matching. The problem with these approaches is that the input queues may become unstable under non-uniform traffic distributions without internal switch speedup [MeKeown1996].

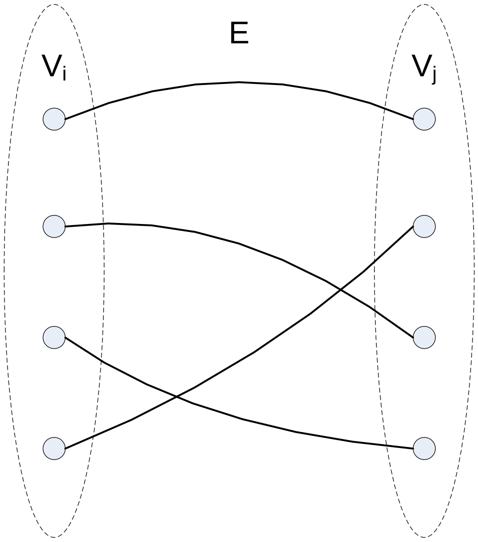


Figure 2.9: A perfect matching G_M of the bipartite graph G of Figure 2.8

2.2.2.1. Maximum Size/Weight Matching

A maximum size/weight[1] matching for a bipartite graph can be found by solving an equivalent network flow problem [Cormen2001][Saberi2006a]. The most efficient maximum size matching algorithm has a complexity $O(N^{2.5})$ [Hopcroft1973] where N denotes port count of a switch, which is believed not practical because it is too complex to be implemented in hardware and takes too long to complete [McKeown1999]. The maximum size matching is not able to maximize the throughput of an IQ switch because it may cause some queues to be starved of service indefinitely [McKeown1996].

[1] The maximum size matching maximizes the cardinality of a matching while the maximum weight matching maximizes the weight of a matching. The weight may be measured by the occupation of each VOQ, the traffic arrival rate, the waiting time of the slot at the head of line and so on.

Different from maximum size matching where the cardinality of a matching is maximized, a maximum weight matching maximizes the total weight of a matching. The weight may be represented by measurable parameters such as occupation of each VOQ, rejection rate [Saberi2006b] and so on. The main concern of this matching algorithm is the high computation complexity.

2.2.2.2. Maximal Size/Weight Matching

In order to reduce the computation complexity, heuristic algorithms called maximal matching algorithms have been proposed. A common characteristic of these algorithms is that they add each edge to a matching one at a time[1], and if the edge has been added it will not be later removed from the matching[2]. The algorithm terminates when no input or output ports remain unnecessarily unmatched. The edge-adding procedure contains a number of edge-adding iterations. At each iteration, three arbitration steps operate in parallel on each input and output port. As shown in Figure 2.10, these steps are:

Step1: Request. Each unmatched input port sends a request to each output port for which it has slots to send;

Step2: Grant. Any unmatched output port that receives more than one request grants one request according to a certain input port selection rule;

Step 3: Accept. Any input port that receives more than one grant accepts one of them according to a certain output port selection rule.

[1] The initial matching does not contain any edges. As more edges are added in, the matching tends to become maximal, i.e. no further edge can be added.

[2] As a comparison, the maximum matching may remove earlier added edges in order to maximize the weight or cardinality of a matching.

Each iteration only takes into account those input and output ports that remained unmatched in the previous iterations. The algorithm stops when no input or output ports remain unnecessarily unmatched.

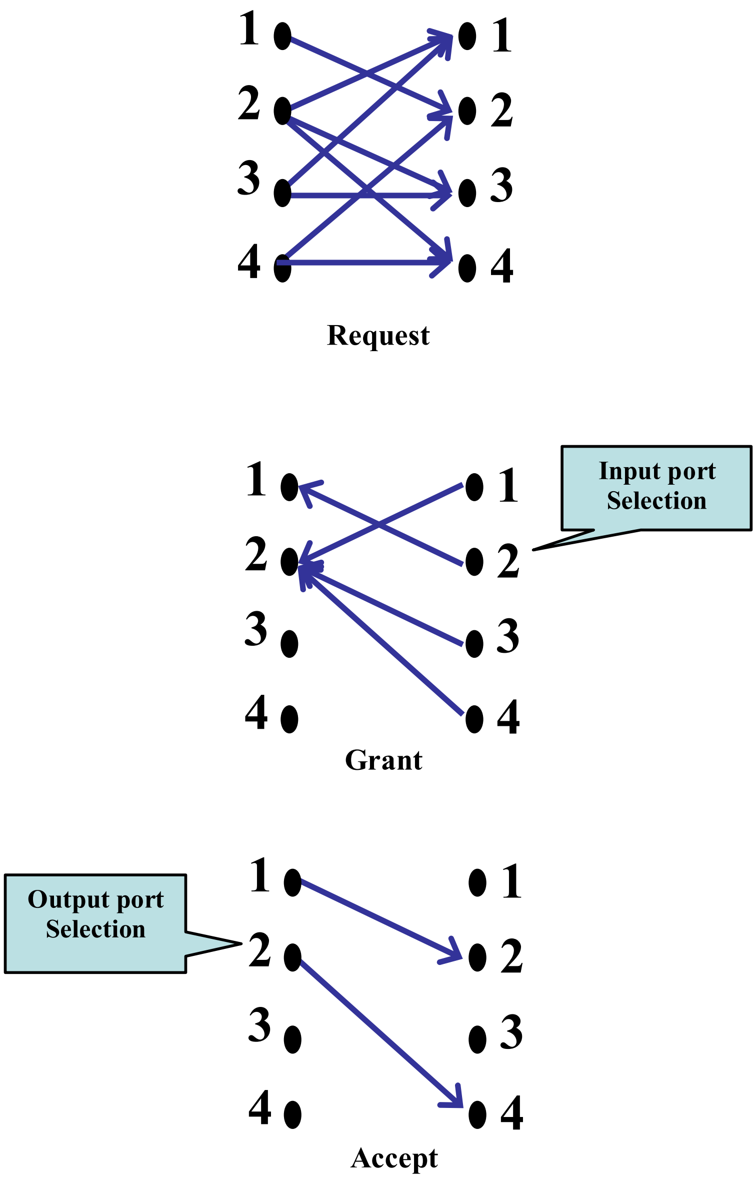


Figure 2.10: Three arbitration steps of the maximal size/weight matching for a given iteration

In the literature, many scheduling algorithms have been proposed which follow the maximal size matching scheme. The difference among them lies in the different selection rules on input and output ports.

Parallel Iterative Matching (PIM) [Anderson1993] and its variance the statistical slot-by-slot scheduling [Liu2005b] use a random process to select input and output ports, and thus avoids starvation. The algorithm finds a maximal match in $O(\log N)$ iterations, on average, independent of the pattern of requests.

Karp [Karp1990] proposed a simple randomized algorithm for on-line bipartite matching. Karp's approach is different from PIM in that Karp's approach visits input ports in random order and finds bipartite graph matchings by visiting output ports in order.

Iterative Round-Robin Matching (iRRM) [McKeown1993] uses round-robin arbiters to select input and output ports. One drawback of the algorithm is that it cannot achieve 100% throughput for admissible traffic when the load is heavy.

Iterative Round-Robin with SLIP (iSLIP) [McKeown1999] is identical to iRRM except for the input port selection rule in Step 2 which requires the round-robin arbiter on an output port to adjust itself to one location beyond the granted input port if the grant is accepted in Step 3; otherwise, the arbiter runs in a round-robin fashion. This small change to the iRRM leads to an improvement of throughput. The algorithm can achieve 100% throughput for uniform traffic, and quickly adapts to an efficient round-robin policy among the busy queues under non-uniform traffic.

FCFS (first-come-first-serve) in Round-Robin Matching (FIRM) [Serpanos2000] differs from iSLIP only in the input port selection rule, i.e. if the grant is not accepted in Step 3, the round-robin arbiter on an output port adjusts itself to the granted input port. It is shown that FIRM achieves a lower mean slot delay than iSLIP does at high load.

Dual Round-Robin Matching (DRRM) [Chao2000] simplifies the arbitration to two steps.

Step 1: each unmatched input port selects one of the output ports for which it has slots to be sent in a round-robin fashion, and sends a request to that port;

Step 2: any unmatched output port that receives more than one request grants one in a round-robin fashion.

The DRRM algorithm is more sophisticated than iSLIP in the sense that less information exchange is needed between input and output ports. A simulation [Chao2000] shows that DRRM's and iSLIP's performances are comparable at speedup of 2, while they have almost the same performance when the speedup is larger than 3.

Logical Equivalence of Parallel Iterative Matching (LE-PIM) [Nong1999] is similar to DRRM in that, at each iteration, an input port sends a request to one output port only. The difference is that LE-PIM uses a random process to select input and output ports, while DRRM uses round-robin.

The maximal size matching algorithms only need to know whether the VOQs at the input ports are empty or not. If the traffic is non-uniform and the occupation of some queues starts to increase, these algorithms cannot increase the service to these queues and further reduce their backlogs. In order to solve the problem, the concept of a "weight" is introduced to the maximal size matching algorithms; the new algorithms are called maximal weight matching algorithms. The weight could be measured by the occupation of each VOQ, the traffic arrival rate, the waiting time of the slot at the head of line and so on. Unlike the maximal size matching, the maximal weight matching attempts to maximize the weight of a matching.

It is shown in [Leonardi2001] that an internal speed-up of two permits strong stability to most scheduling algorithms if VOQs are implemented. It is proven in [Dai2000] that a CIOQ switch running any maximal matching algorithm with a speedup of 2 delivers a 100% throughput. A nontrivial bound is given by [Leonardi2001A] on the delay for maximal weight matching algorithms under admissible i.i.d. Bernoulli traffic. A class of approximation to maximal weight matching algorithms are studied in [Shah2002].

In the literature, many scheduling algorithms were proposed which follow from the maximal weight matching algorithms. The difference lies in the different means of weight measurement.

The Longest Queue First (LQF) [McKeown1999A] considers the occupation as the weight of each VOQ and gives preference to long queues. The algorithm has a computation complexity $O\left(N^3 \log N\right)$. One problem of LQF is that it may lead to indefinite starvation of one or more input ports. Other problems [Mekkittikul1998] include the implementation difficulties in hardware at high speed due to the high computation complexity and the limitation of hardware implementation.

To solve the first problem of LQF, the Oldest Cell First (OCF) [McKeown1999A] was proposed which assigns the weights equal to the waiting time of the slot at the head of line and gives preference to queues with the highest weight. To solve the second problem of LQF, the Longest Port First (LPF) [Mekkittikul1998] was proposed where the weight is the sum of the number of slots sent from an input port and the number of slots destined to an output port.

The high computation complexity of the above algorithms prohibits implementing maximal weight matching algorithms in practical systems. To solve the problem, [Tassiulas1998] proposed a heuristic scheduling algorithm that first introduced randomness to the computation of the optimal maximal weight matching. The key point of Tassiulas's method is a weight comparison between a current memorized matching and a matching randomly selected from all possible matchings. Finally, the heavier one is selected as a matching for the next timeslot. Tassiulas's method is sophisticated in the sense that iteration is no longer needed in the process of achieving a maximal weight matching.

Another randomized algorithm, called LAURA, was proposed in [Giaccone2002]. LAURA uses memory and outperforms Tassiulas' method in delay because the weight of a matching is carried in a few of its edges; therefore, it is better to remember heavy edges than to remember matchings. The computation complexity of LAURA is $O\left(N\log^2 N\right)$, far lower than maximal weight matching algorithms.

Tassiulas's method and LAURA are both stable under admissible Bernoulli i.i.d. traffic. The main concern lies in the randomness because the random number generator is expensive to be implemented in hardware at high speed.

2.2.2.3. Stable Marriage Matching

The stable marriage problem [Gale1962] is a classical combinatorial problem. The problem involves N men and N women and every person has strict preferences over the members of the opposite sex. A matching M of the men and women, i.e. a one-to-one mapping between the two sexes, is said to be unstable if there exists a man-woman pair, who both prefer each other to their currently assigned partners in M . A *stable matching* is a matching that is not unstable. It is known that a stable matching can always be found in time linear in the size of the problem using an efficient algorithm known as the Gale/Shapley algorithm (GSA) [Gale1962].

The stable marriage problem can be viewed as a bipartite graph matching, i.e., switch configurations in an CIOQ switch are analogous to finding a bipartite graph matching between the set of switch inputs and outputs. For example, if the set of men represents the input ports and the set of women represents the output ports, a stable matching thus represents a legal switch configuration.

In the literature, many scheduling algorithms were proposed which follow from the Gale/Shapley algorithm. The differences lie in the different preference criteria of men and women.

Joined Perferred Matching (JPM) was proposed in [Stoica1998], where the preference list of an input port is organized in the inverse order of the arrival time of the slots and that of an output port is ordered by the schedule time of the slots in the corresponding OQ schedule. It is shown that the internal speedup of JMP for a CIOQ switch to achieve exact emulation of an OQ switch is two.

Critical Cell First (CCF) was proposed in [Chuang1999], where the preference list of an input port is ordered by a critical value, i.e. the difference between the waiting time that a slot leaves the output buffer in the corresponding OQ schedule and the time that a slot can wait in the input buffer. If the difference is small, it means that the slot needs to be delivered immediately. It is shown that the internal speedup of CCF for a CIOQ switch to achieve exact emulation of an OQ switch is two.

Lowest Output Occupancy First Algorithm (LOOFA) was proposed in [Krishna1999], where the preference list of an input port is ordered by the backlog of output ports and that with the lowest occupation was given the highest priority. LOOFA is work-conserving for all traffic patterns and provides delay guarantees in a crossbar for a speedup of two.

2.2.2.4. Summary

Stable marriage matching is similar to maximal size\weight matching in the sense that both contain two selection processes residing in input and output ports. However, they are different [McKeown1995] in that:

- It is not clear whether a stable matching will lead to an efficient usage of switch bandwidth, or whether it will prevent connections from being starved of service.
- The stable marriage matching is usually defined for a perfect matching in which every input and every output is matched. Such a perfect matching, however, cannot always be obtained in an input-queued switch. This implies that some inputs and outputs will have missing entries in their preference lists, which leads to a large reduction in the size of a matching.
- A stable-marriage matching may or may not have the maximum weight, and a maximum-weight matching may or may not be a stable marriage [Kam1999].

2.2.3. Frame-by-Frame Allocation

The computation time of generating a schedule is restricted by timeslot duration for the slot-by-slot allocation algorithms. The simplest idea to solve the problem is that several timeslots are grouped together as a frame with fixed size and the switch fabric is only updated on frame boundaries [Saberib]. Thus the time to compute a new matching is relaxed to several timeslots [Li2003]. However, the bandwidth sharing granularity increases compared to slot-by-slot allocation. In order to keep the same granularity, a pipeline-based approach and the Birkhoff-von Neumann decomposition approach have been proposed in the literature.

2.2.3.1. Pipeline-Based Approach

A pipeline-based approach for maximal size matching scheduling, called PMM, for an IQ switch was proposed in [Oki2001] to relax the short timeslot duration. Within a scheduler, more than one subscheduler operates in a pipelined manner (Figure 2.11). Each subscheduler performs maximal size matching independently[1] and is allowed to take more than one timeslot. At each timeslot, one of them provides the matching result. The problem of pipeline-based approach is that it can lead to instability and unfairness, particularly for non-uniform traffic [Oki2001].

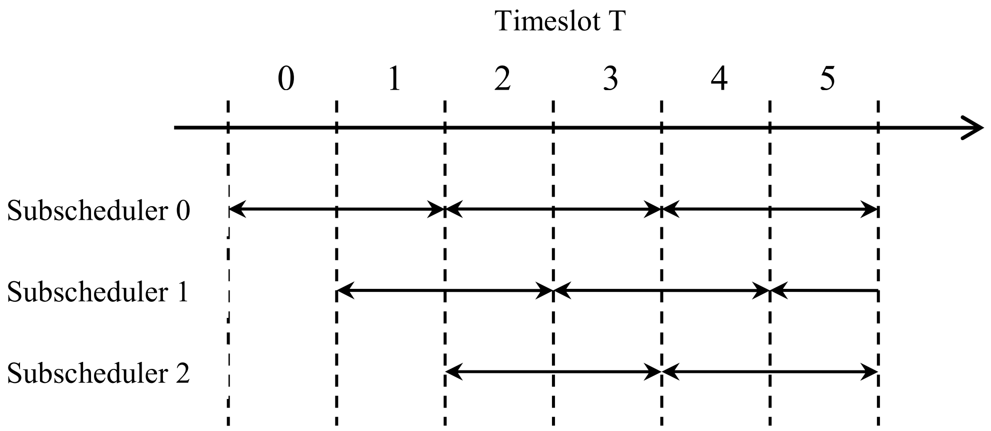


Figure 2.11: Time diagram of PMM with a frame size of three

[1] The subschedulers can actually adopt any pre-existing slot-by-slot algorithm.

2.2.3.2. Birkhoff-von Neumann Decomposition Approach

Non-uniform traffic is prevalent in current networks. In order to handle such traffic, the Birkhoff-von Neumann (BvN) decomposition approach for IQ switch scheduling has been introduced in [Chang2000] and [Towles2003]. The BvN decomposition algorithms rely on the Birkhoff-von Neumann Theorem by [Birkhoff1946] and [Neumann1953]. Below is a brief summary of this theorem.

A nonnegative matrix $R = (r_{ij})$ where $0 \le r_{ij} \le 1, (r_{ij} \in R)$ is called a *doubly sub-stochastic matrix* if the following two conditions are satisfied:

$$\forall j, \sum_{i=1}^{N} r_{ij} \le 1 \text{, and}$$

$$\forall i, \sum_{j=1}^{N} r_{ij} \le 1$$

Furthermore, R is called *doubly stochastic* if both of the above inequalities are equalities. A *permutation matrix* is a matrix with (0,1)-entries whose row sums and column sums are equal to one. Similarly, a *partial permutation matrix* is defined as a matrix with (0,1) entries whose row sums and column sums are at most equal to one.

Von Neumann's theorem: ([Neumann1953]) If a matrix $R = (r_{ij})$ is doubly sub-stochastic, then there exists a doubly stochastic matrix $R^{'} = (r_{ij}^{'})$ such that

$$\forall i, j, r_{ij} \le r_{ij}^{'}$$

The von Neumann's theorem indicates the possibility of transforming a doubly sub-stochastic matrix to a doubly stochastic matrix. Sinkhorn proposed a method to achieve this transformation [Sinkhorn1964. His idea is to repeatedly perform the following two steps: 1) for each row in the matrix, sum the row elements and divide each element by this sum; and 2) for each column in the matrix, sum the column elements and divide each element by this sum. However, as indicated in [Sinkhorn1964], the method may not converge for certain doubly sub-stochastic matrices. Therefore, the convergence condition and the rate of convergence are studies in [Knopp1967] and [Soules1991], respectively.

A $N \times N$ doubly sub-stochastic matrix may be viewed as a normalized traffic demand matrix for a switch with the same dimension, i.e. N input ports and N output ports. Each entry of the matrix denotes the normalized traffic demand between input and output port with the restriction that the maximum normalized bandwidth for each port is equal to one. The von Neumann's theorem says that any normalized traffic demand matrix may be transformed to a matrix where (1) all these normalized traffic demand may be satisfied and (2) the available bandwidth is fully used.

Birkhoff's theorem: ([Birkhoff1946]) For a doubly stochastic matrix $R^{'} = (r_{ij}^{'})$, there exists a set of positive numbers Φ_k and permutation matrices P_k such that

$$R^{'} = \sum_k \Phi_k P_k \text{ , where } \sum_k \Phi_k = 1$$

The Birkhoff's theorem indicates that a doubly stochastic matrix can be decomposed into a series of permutation matrices with real coefficients, the sum of which is one. To achieve this transformation, an algorithm has been proposed in [Chang2000] [Chang2001] and [Giaccone2002]. This algorithm produces up to $N^2 - 2N + 2$ permutations [Johnson1960][Hoffman1953] with real-valued expansion coefficients and possesses a computation complexity of $O(N^{4.5})$ where N denotes dimension of the matrix.

The von Neumann's and Birkhoff's theorems can be restricted to the integer domain. In this case, the concept of normalized traffic demand matrix can be extended to a *traffic* *demand matrix* $T = (t_{ij})$ by multiplying each entry with F, the frame size counted by the number of timeslots, and taking the integer part. The entry of the traffic demand matrix t_{ij} denotes the number of slots waiting in a frame to be transported from source i to destination j (in number of slots). The total service that a switch may provide can be described by a *service matrix*. The service matrix has two properties: (1) the row and column sums are equal to F; (2) all entries are non-negative. The service matrix indicates the bandwidth allocated for each source-destination pair by the scheduler; while the traffic demand matrix indicates the bandwidth requested for each source-destination pair by the edge nodes (the customers).

The general BvN decomposition approach for calculating a schedule includes two steps:

- constructing a service matrix from the traffic demand matrix; and
- decomposing the service matrix into permutation matrices.

Below we describe an example of the BvN decomposition approach. In this example, it is assumed that there is a 3×3 switch equipped with a frame-based scheduler that may generate switch configurations for each timeslot of a frame. The frame size F is assumed to be six (in number of timeslots). The following traffic demand matrix T is given.

$$T = \begin{pmatrix} 1 & 0 & 2 \\ 3 & 1 & 1 \\ 2 & 2 & 0 \end{pmatrix}, \quad t_{ij} \geq 0$$

It is aimed to transmit this traffic to their destinations in the time of a frame. Therefore, the service matrix S, according to the traffic demand, may be constructed as follows:

$$S = \underbrace{\begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 2 & 2 & 2 \end{pmatrix}}_{F=6}$$

Based on this service matrix S, six permutation matrices may be generated, one for each timeslot.

$$\underbrace{\begin{matrix} \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \end{matrix}}_{\text{Frame size } = 6}$$

2.2.4. Techniques Applied to the AAPN

The pipeline-based approach, which is based on the frame-by-frame allocation, relaxes the strict scheduling timing constraint but it cannot efficiently handle non-uniform/bursty traffic due to the independency of each sub-scheduler [Oki2001] [Mckeown1999]. The BvN decomposition approach, as an alternative, calculates a schedule for each frame, and can efficiently handle non-uniform/bursty traffic because it allocates bandwidth according to the traffic demand between input and output ports. Therefore, it is reasonable to believe that the BvN decomposition approach is a good candidate for the AAPN scheduling.

However, two challenges of the BvN decomposition approach should be solved in this work, i.e.

- How to construct a service matrix that closely reflects the traffic demand of each source-destination pair?
- How to find a decomposition algorithm with low computation complexity to enable practical implementation?

2.3. Routing Schemes

The main issue for routing in AAPN is sharing the load between different AAPN core nodes. The purpose of load sharing is to relieve network congestion and improve network performance [Zheng2004]. In a packet/burst-switched network, load balancing can reduce packet/burst delay and loss. In a circuit-switched network, it can reduce blocking probability and thus accommodate more connections.

Brunato [Brunato2003] proposed a load-balancing algorithm called Reverse Subtree Neighborhood Exploration (RSNE) for dynamic lightpath establishment in wavelength-routed networks. It is based on IP-like routing and a local searching mechanism, and allows an incremental implementation where local searching steps are continuously performed as traffic conditions change.

In [Hsu2001], Hsu et al. investigated adaptive routing algorithms for wavelength-routed networks, including the least-loaded path strategy and the weighted shortest path strategy. The former can balance traffic load well among all links. The latter can minimize resource cost while maintaining traffic load among all links as balanced as possible.

In this thesis work, the routing strategy for the AAPN is studied that considers not only the blocking probability but also the traffic delays.

3. Bandwidth Allocation

3.1. Introduction

In the AAPN, each edge node signals traffic information to the core node along control channels before sending slots. Once at the core, the global traffic demand is arranged in the form of a traffic demand matrix, with each element corresponding to a source-destination traffic flow. The scheduler 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 switches are configured in coordination with the edge nodes according to the allocated permutations.

For optical long-haul networks, the efficiency of bandwidth allocation schemes degrades due to the large propagation delay for signaling between core and edge nodes, as the schedule calculated is not based on the up-to-date traffic information (see Section 3.2.2). In order to mitigate this problem, traffic estimation mechanisms should be employed at the core node to estimate the future traffic.

The term "*admissibility*" implies that the traffic does not oversubscribe the source or destination ports. While 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 a potentially inadmissible traffic matrix. A real/integer η-server service matrix $\mathbf{s}=\left(s_{ij}\right)$ 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 a frame based TDM system. 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. The service matrix determines the maximum service bandwidth a switch can provide and the corresponding bandwidth allocations, i.e. the timeslots allocated to all edge node pairs.

3.2. Constructing a Traffic Demand Matrix

The traffic demand, which represents the bandwidth required between input and output ports, is kept in the form of a traffic demand matrix $T=(t_{ij})$. The BvN decomposition approach may allocate bandwidth based on this matrix. For optical long-haul networks, large propagation delays exist between edge and core nodes and make it difficult for the allocation algorithm to efficiently adapt to fast traffic changes, i.e. the traffic information is out-of-date when it arrives at the core node and the calculated schedules cannot reflect the traffic in the future. Therefore, traffic estimation mechanisms should be employed in the AAPN to estimate the traffic volume arriving in the future in order to compensate for the out-of-date of traffic information.

3.2.1. Signaling Protocol

The time line for frame-based signaling is shown in Figure 3.1, for the case of two edge nodes, A and B, and a core node. A sample signaling scenario where data are sent from *A* to *B* is described as follows.

(a) Edge node *A* collects the traffic information, i.e. the number of aggregated slots during the current frame time.

(b) Edge node *A* sends this information to the core node.

(c) The code node calculates the schedule for the future time.

(d) The schedule is then sent to these two edge nodes.

(e) Edge node A waits before initiating the new schedule due to the synchronization required between edge nodes, i.e. some other edges may be further away from the core node and require more time to receive the schedule.

(f) Some data bursts are scheduled to be sent from A to B.

(g) When the bursts pass through the core, the core switch will be configured precisely to transfer these bursts to B.

(h) The bursts travel to B.

(i) The reception schedule at B provides information about in which timeslot the bursts from A arrive at B.

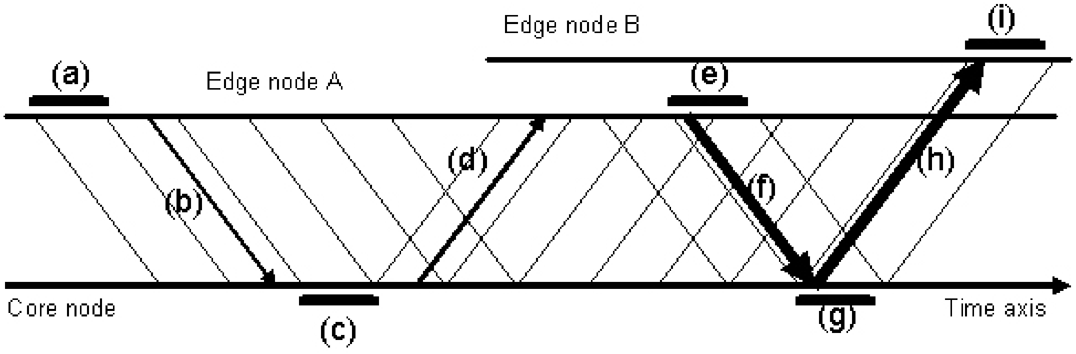


Figure 3.1: Time line of frame-based signaling

3.2.2. Traffic Demand Matrix Construction

3.2.2.1. Basic Ideas

In Figure 3.1, the slot sent by the edge node *A* at time (e) is based on the schedule calculated at time (c) at the core node. Ideally, this schedule should be calculated according to the queue state at time (e) so that it can be most adaptive to the traffic pattern then. However, the schedule calculated at time (c) is based on the queue state (i.e. the queue length) at time (a) that is earlier than time (e) due to the signaling delays. In order to mitigate the difference between the queue state at time (a) and (e), the traffic arrivals (i.e. the number of aggregated slots) in between should be estimated at time (c).

The core node maintains a *virtual queue-length matrix* $Q = (q_{ij})$, each entry of which corresponds to the length of a VOQ at the edge nodes. When the traffic information (i.e. the number of aggregated slots to each destination edge node that have arrived during the last frame period) arrives at the core node, the corresponding entries of the matrix are updated (by adding the number of aggregated slots in order to increase the queue length).

Meanwhile, at each timeslot, the core node selects some entries of this matrix and decreases them by one; the selection is based on the schedule applied to the edge nodes at that timeslot, i.e. the entry should be decreased by one if a slot is sent out from the corresponding VOQ.

The traffic estimation matrix $E = (e_{ij})$ is defined as the estimated traffic (counted by slots) that is expected to arrive between time (a) and time (e).

Therefore, the traffic demand matrix T can be calculated by

$$T = Q + E \tag{3.1}$$

3.2.2.2. Traffic Estimation

In this section, a simple moving average (SMA) method is discussed to estimate the traffic between time (a) and time (e) (Figure 3.1). The method is one of the most popular and easy methods to estimate trends of a data series.

The simple moving average is formed by computing the mean value of the last few numbers in a data series. A k-period SMA is calculated by adding the last k data point in the data series and dividing the total by k. By applying k-period SMA, the estimation of the p th data, $SMA(a_p)$, ($p > k$), in a data series $\{a_n\}, n = 1, 2, 3....$, is equal to $(\sum_{i=p-k}^{p-1} a_i)/k$.

Moving averages are lagging indicators and therefore fit in the category of trend following indicators. The SMA may be applied to estimate the traffic estimation matrix E. For the AAPN with N edge nodes, the traffic information from a given edge node $i, i = 1, 2, ...N$, to a given edge node $j, j = 1, 2, ...N, j \neq i$ is sent frame-by-frame and therefore forms a data series λ_{ijk}, where k is the frame number. Suppose λ_{ijp} is sent at the

p -th frame, which is at time (a), and a period of frame time $\hat{\tau}_{ij}$ (counted by the number of frames) is defined for the estimation, then the τ_{ij} -period SMA can be used to estimate $\lambda_{ij(p+1)}$. The estimated $\lambda_{ij(p+1)}$ that is denoted by $\hat{\lambda}_{ij(p+1)}$ can be calculated by

$$\hat{\lambda}_{ij(p+1)} = (\sum_{k=p-(\hat{\tau}_{ij}-1)}^{p} \lambda_{ijk}) / \hat{\tau}_{ij} \tag{3.2}$$

Suppose the duration between time (a) and time (e) is τ_{ij} frame time (counted by the number of frames) .The estimated traffic e_{ij} between time (a) and time (e) is

$$e_{ij} = \hat{\lambda}_{ij(p+1)} \cdot \tau_{ij} \tag{3.3}$$

Finally, the traffic estimation matrix $E = (e_{ij})$ can be constructed.

3.3. Constructing a Service Matrix

There are many methods proposed in the literature to construct a service matrix. A simple rescaling method followed by quantization and an integer rate filling method is described in [Paredes2005] to construct a service matrix in a Clos packet switch. Since an integer service matrix represents a regular bipartite graph, it is suitable for decomposition by an edge coloring algorithm (or heuristic). The limitation of the method is that the value difference between small entries may be lost when the maximum row/column sum in a traffic matrix is very large in comparison to the others [Paredes2005].

A weighted rate filling method is proposed in [Li2001] which can be used to construct doubly stochastic matrices from doubly substochastic matrices, i.e. constructing a 1-server service matrix from a real γ-server traffic matrix (normalized on a per slot basis such that $0 < \gamma \leq 1$), proportionally to the weights (i.e. the value of each entry) . This ad hoc method is not robust against inadmissible traffic demand.

A max-min fairness method is presented in [Yim2004] to construct the service matrix by extending the notion of max-min fairness to take into account the traffic demand from each stream. 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 [Paredes2005]). The complexity of this method is $O(N^3)$.

In the AAPN, it is required that the constructed service matrix have 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 (i.e. the service is as "proportionally fair" as possible[1]); and the construction method should be robust against inadmissible traffic demand. With this in mind, a new method, called the Alternating Projections method, is introduced in the following sections. The method satisfies the requirements outlined above.

3.3.1. Preliminary Concepts

The traffic matrix **t** may be defined as an element in the 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 ($\sum_i s_{ij} = \sum_j s_{ij}, \forall i, j \in [1, N]$) forms a subspace $S \subset M$. The set of all matrices with non-negative entries forms a subset $P \subset M$ that is a

[1] To be exactly proportionally fair, the service matrix must be exactly proportional to the traffic matrix. But to be a service matrix, it must have equal row and column sums, which is most unlikely for a traffic matrix. If the row and column sums are not equal, then there will be unused slots. It might be thought that one could relax the notion of proportionate fairness to apply only amongst flows from a source (i.e rows) or only amongst flows to the same destination (i.e. columns). But given that the sum of all the row sums must equal the sum of all the column sums, then proportionally fair allocations to sources will lead to some oversubscribed and other undersubscribed destinations and proportionally fair allocations to destinations will lead to some undersubscribed and some oversubscribed sources. Hence feasible service matrices cannot in general be exactly proportionately fair.

convex cone. The intersection $P \cap S \subseteq M$ is therefore the convex cone of all possible service matrices. This motivates the application of the method of alternating projections [Boyle1986] on the convex sets S, P, to produce a sequence of matrices that converge to the projection of $\mathbf{t} \in M$ 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 $\sum_i s_{ij} = \sum_j s_{ij} = \eta$.

Definition 1: The set of real valued $N \times N$ matrices $R^{N \times N}$ equipped with the standard inner product $(\mathbf{a}, \mathbf{b}) = \sum_{i,j} a_{ij} b_{ij}$ ($\forall \mathbf{a}, \mathbf{b} \in R^{N \times N}$) is complete in the topology induced by the associated norm: $\|\mathbf{a}\| = (\mathbf{a}, \mathbf{a})^{\frac{1}{2}}$ and may therefore form a metric 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:

$$\rho(\mathbf{a}, \mathbf{b}) = \rho(\alpha \mathbf{a}, \beta \mathbf{b})$$

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: $\lambda \mathbf{a} + (1 - \lambda) \mathbf{b} \in K, \forall \lambda \in [0,1], \quad \mathbf{a}, \mathbf{b} \in K$

Definition 5: A cone is a subset $K \subseteq M$ defined by the property:

$$\mathbf{a} \in K \Rightarrow \lambda \mathbf{a} \in K, \forall \lambda \in [0, \infty]$$

Definition 6: A closed convex cone $K \subseteq M$ defined by the property:

$$\begin{aligned}&\lambda(1-t)\mathbf{a}+\lambda t\mathbf{b}\in K,\forall \mathbf{a},\mathbf{b}\in K\\ &\forall t\in\left[0,1\right],\forall\lambda\in\left[0,\infty\right]\end{aligned}$$

Definition 7: A linear sub-space $K \subseteq M$ is defined by the property:

$$\alpha\mathbf{a}+\beta\mathbf{b}\in K,\forall\mathbf{a},\mathbf{b}\in K,\forall\alpha,\beta\in R$$

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

Theorem 1: Given a closed convex cone S of a linear space M

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

the projection $\mathbf{b}=\mathbf{P}_S(\mathbf{a})$ 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: By the method of Lagrange multipliers, the computation proceeds by first minimizing the cost:

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

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

By minimizing the cost, we have

$$\frac{\partial C}{\partial b_{ij}}=0$$

Therefore

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

equivalently:

$$b_{ij} = a_{ij} - p_i - q_j$$

where

$$p_i = \alpha_i - \frac{1}{N}\sum_i \alpha_i$$
$$q_j = \beta_j - \frac{1}{N}\sum_j \beta_j$$

We have

$$\sum_i p_i = 0,$$
$$\sum_j q_j = 0$$

now:

$$\sum_i b_{ij} = \sum_i a_{ij} - Nq_j = \frac{1}{N}\sum_{i,j} b_{ij},$$

$$\sum_j b_{ij} = \sum_j a_{ij} - Np_i = \frac{1}{N}\sum_{i,j} b_{ij}$$

Therefore

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

Since

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

We have

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

It is easy to prove that the projection $\mathbf{P}_S$ has the following properties:

1. $\mathbf{P}_S$ is a projection:

$$\mathbf{P}_S\mathbf{P}_S = \mathbf{P}_S$$

2. $\mathbf{P}_S$ is a linear operator:

$$\mathbf{P}_S(\alpha\mathbf{a} + \beta\mathbf{b}) = \alpha\mathbf{P}_S(\mathbf{a}) + \beta\mathbf{P}_S(\mathbf{b})$$

3. $\mathbf{P}_S$ is an orthogonal projection:

$$\mathbf{P}_S(\mathbf{a} - \mathbf{P}_S(\mathbf{a}), \mathbf{P}_S(\mathbf{a})) = 0$$

4. The sum over all matrix elements is invariant under the action of $\mathbf{P}_S$:

$$\sum_{i,j}(\mathbf{P}_S(\mathbf{a}))_{ij} = \sum_{i,j}\mathbf{a}_{ij}$$

Theorem 2: Given the closed convex cone of non-negative matrices

$$P = \left\{\mathbf{a} \in M \,\middle|\, a_{ij} \geq 0\right\},$$

the projection $\mathbf{b} = \mathbf{P}_P(\mathbf{a})$ of $\mathbf{a} \in M$ onto P is given by

$$b_{ij} = a_{ij}\text{, if } a_{ij} \geq 0 \text{ and } b_{ij} = 0\text{, if } a_{ij} < 0$$

Proof: Decompose $\mathbf{a}$ into its positive and negative parts.

$$\mathbf{a} = \mathbf{a}_+ - \mathbf{a}_-, \quad \mathbf{a}_+, \mathbf{a}_- \in P$$

Observe that because an element of a matrix is either non-negative or negative and never both, that is, the supports (the subset of indices of non-negative elements) of $\mathbf{a}_+, \mathbf{a}_-$ form a disjoint partition of the index set. Hence we also have the decomposition:

$$\mathbf{b} = \mathbf{b}_+ + \mathbf{b}_-, \quad \mathbf{b}_+, \mathbf{b}_- \in P$$

where $\mathbf{a}_+, \mathbf{b}_+$ and $\mathbf{a}_-, \mathbf{b}_-$ share the same supports.

Hence:

$$\begin{aligned}\delta^2(\mathbf{a},\mathbf{b}) &= \|\mathbf{a}-\mathbf{b}\|^2 \\ &= (\mathbf{a}-\mathbf{b}, \mathbf{a}-\mathbf{b}) \\ &= \left(\left(\mathbf{a}_+ - \mathbf{b}_+\right) - \left(\mathbf{a}_- + \mathbf{b}_-\right), \left(\mathbf{a}_+ - \mathbf{b}_+\right) - \left(\mathbf{a}_- + \mathbf{b}_-\right)\right) \\ &= \left(\left(\mathbf{a}_+ - \mathbf{b}_+\right), \left(\mathbf{a}_+ - \mathbf{b}_+\right)\right) + \left(\left(\mathbf{a}_- + \mathbf{b}_-\right), \left(\mathbf{a}_- + \mathbf{b}_-\right)\right)\end{aligned}$$

where use has been made of the fact that matrices with disjoint supports are orthogonal.

$\delta^2(\mathbf{a},\mathbf{b})$ achieves its minimum when

$$\mathbf{b}_+ = \mathbf{a}_+ \text{ and } \mathbf{b}_- = \mathbf{0}$$

It is easy to prove that the projection $\mathbf{P}_P$ has the following properties

1. The projection $\mathbf{P}_P$ is a nonlinear operator.

2. $\mathbf{P}_P$ is an orthogonal projection:

$$\mathbf{P}_P(\mathbf{a}-\mathbf{P}_P(\mathbf{a}), \mathbf{P}_P(\mathbf{a})) = 0$$

3.3.2. Alternating Projections Method to Construct a Service Matrix

Theorem 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, Theorem 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. Theorem 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 possesses equal row and column sums. In order to find a service matrix with both qualities, Dykstra's algorithm [Bauschke2002] [Boyle1986] may be applied to find the projection onto $S \cap P$; that is, the two projections $\mathbf{P}_S$ and $\mathbf{P}_P$ are applied alternately. Applying the alternating projections algorithm ensures that the limit of the resulting sequence is the valid service matrix nearest to the initial matrix $\mathbf{t}$.

Algorithm: ([Peng2006a])

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

```
1. $\mathbf{t}_{prev}^{-} = 0; \quad \mathbf{t}^{+} = \mathbf{t}$
2. $\mathbf{t} = \mathbf{P}_S(\mathbf{t}^{+})$;
3. $\mathbf{t} = \mathbf{t} + \mathbf{t}_{prev}^{-}$
4. $\mathbf{t}^{+} = \mathbf{P}_P(\mathbf{t})$
5. $\mathbf{t}_{prev}^{-} = \mathbf{t} - \mathbf{t}^{+}$
6. If $\forall i, j, \sum_i t_{ij}^{+} \Big/ \left( (1/N) \sum_{i,j} t_{ij} \right) - 1 < \varepsilon$ and $\sum_j t_{ij}^{+} \Big/ \left( (1/N) \sum_{i,j} t_{ij} \right) - 1 < \varepsilon$, then go to step 7;
   otherwise, loop back to step 2.
7. Output $\mathbf{s} = \left[ \eta \Big/ \left( (1/N) \sum_{i,j} t_{ij} \right) \right] \times \mathbf{t}^{+}$
```

Note that the procedure will be repeated until the residual error is less than ε. Convergence is proven by the general result found in reference [Boyle1986]. A η-server service matrix $\mathbf{s}$ with the same relative precision may be formed by scaling each entry of the $\mathbf{t}^{+}$ by $\eta \Big/ \left((1/N) \sum_{i,j} t_{ij} \right)$.

Theorem 3: The alternating projections method chooses the service matrix most similar to the input traffic demand matrix.

Proof: Given a closed convex cone K of a linear space M where K is the set of matrices with non-negative entries and row and column sums all equal (i.e. valid service matrices) and M is the set of non-negative real valued matrices (i.e. traffic demand matrices), the alternating projections method can be defined by a projection $\mathbf{s} = P_{\mathrm{K}}(\mathbf{t})$ of $\mathbf{t} \in M$ onto K (where $\mathbf{s} \in \mathrm{K}$).

It can be derived from [Boyle1986] that, given a $\mathbf{t} \in M$ and a ε that is small enough, the method converges to a unique matrix $\mathbf{s}_*$, $\mathbf{s}_* \in \mathrm{K}$. Therefore, the distance between the two matrices, $\mathbf{s}_*$ and $\mathbf{t}$, is the minimum, i.e.

$$\forall \mathbf{s} \in \mathrm{K}, \min\{\delta(\mathbf{s}, \mathbf{t})\} = \delta(\mathbf{s}_*, \mathbf{t})$$

For the $\mathbf{s}_* \in \mathrm{K}$, we have $\lambda \mathbf{s}_* \in \mathrm{K}$, $\forall \lambda \geq 0$ and hence

$$\delta(\lambda \mathbf{s}_*, \mathbf{t}) \geq \delta(\mathbf{s}_*, \mathbf{t}) \quad \forall \lambda \geq 0 \tag{3.5}$$

with equality when $\lambda = 1$

It is easy to deduct that the function $\delta(\lambda \mathbf{s}_*, \mathbf{t})$ attains its minimum $\delta(\mathbf{s}_*, \mathbf{t})$ with respect to λ when:

$$\lambda = \frac{(\mathbf{t}, \mathbf{s}_*)}{(\mathbf{s}_*, \mathbf{s}_*)}. \tag{3.6}$$

In terms of (3.5) and (3.6), we have

$$\frac{(\mathbf{t}, \mathbf{s}_*)}{(\mathbf{s}_*, \mathbf{s}_*)} = 1 \Rightarrow (\mathbf{t}, \mathbf{s}_*) = (\mathbf{s}_*, \mathbf{s}_*) \tag{3.7}$$

Since

$$\left(\mathbf{t},\mathbf{s}_*\right)=\left(\mathbf{t}-\mathbf{s}_*+\mathbf{s}_*,\mathbf{s}_*\right)=\left(\mathbf{t}-\mathbf{s}_*,\mathbf{s}_*\right)+\left(\mathbf{s}_*,\mathbf{s}_*\right) \tag{3.8}$$

In terms of (3.7) and (3.8), we have

$$\left(\mathbf{t}-\mathbf{s}_*,\mathbf{s}_*\right)=0 \tag{3.9}$$

P_K is therefore an orthogonal projection.

By (3.7) and (3.9), we have

$$\begin{aligned}
&\delta^2\left(\mathbf{s}_*,\mathbf{t}\right)=(\mathbf{s}_*-\mathbf{t},\mathbf{s}_*-\mathbf{t})\\
&=\left(\mathbf{t},\mathbf{t}\right)\left[1-\frac{\left(\mathbf{t},\mathbf{s}_*\right)\left(\mathbf{s}_*,\mathbf{t}\right)}{\left(\mathbf{s}_*,\mathbf{s}_*\right)\left(\mathbf{t},\mathbf{t}\right)}\right]\\
&=\left(\mathbf{t},\mathbf{t}\right)\left[1-\rho^2\left(\mathbf{s}_*,\mathbf{t}\right)\right]\\
&\Rightarrow\\
&\rho\left(\mathbf{t},\mathbf{s}_*\right)=\left(1-\frac{\delta^2\left(\mathbf{s}_*,\mathbf{t}\right)}{\left(\mathbf{t},\mathbf{t}\right)}\right)^{\frac{1}{2}}
\end{aligned} \tag{3.10}$$

Since $\left(\mathbf{t},\mathbf{t}\right)$ is fixed, the similarity $\rho\left(\mathbf{t},\mathbf{s}\right)$ attains its maximum when $\mathbf{s}=\mathbf{s}_*$. The result here implies that the projection method may choose a service matrix $\mathbf{s}_*$ most similar to the input traffic matrix $\mathbf{t}$.

3.3.3. Constructing a Valid Integer-Valued Service Matrix

The alternating projection method is used to construct a real-valued service matrix. For the AAPN, the minimum bandwidth unit is a timeslot. Therefore, the real-valued service matrix should be converted into an integer-valued service matrix, the entry of which denotes bandwidth allocated in a frame (counted by the number of timeslots) for the given input-output port pair (which is called *interconnection*).

For a η -server service matrix $S = (s_{ij})$ outputted by the alternating projections method, a simple step to convert it into an integer-valued matrix $\hat{S} = (\hat{s}_{ij})$ is by taking the integer fraction of each entry:

$$\hat{s}_{ij} = \lfloor s_{ij} \rfloor \tag{3.11}$$

It is easy to prove

$$\begin{aligned} &\forall i, \sum_j \hat{s}_{ij} \le \eta \\ &\forall j, \sum_i \hat{s}_{ij} \le \eta \end{aligned} \tag{3.12}$$

The formula (3.12) implies that the integer-valued matrix $\hat{S}$ is not necessarily a valid integer-valued service matrix because some timeslots remain free. Therefore, a *filling algorithm* should be used to assign these free timeslots to interconnections in order to construct a valid service matrix in terms of $\hat{S}$.

The filling algorithm gives some priority to each interconnection and allocates the free timeslots according to the priority. The priority is given by two criteria: (1) the interconnections that have not been assigned any timeslots will be given the highest priority, which will prevent queues from being starved; (2) for other interconnections, the priority is based on the decimal fraction of each entry in S; those with larger decimal fraction will be given higher priority because bandwidth requested by them is closer to the minimum bandwidth unit, i.e. timeslot. The algorithm allocates, if possible, one timeslot per interconnection according to the priority sequence and terminates in a finite number of steps when every row and column sum of $\hat{S}$ is equal to η (when all timeslots have been allocated).

Filling algorithm:

Input: A η-server service matrix S and the frame size η.

1. The matrix $\hat{S} = (\hat{s}_{ij})$, $\hat{s}_{ij} = \lfloor s_{ij} \rfloor$ is constructed by the integer fraction of S; the matrix $D = (d_{ij}) = S - \hat{S}$ is constructed by the decimal fraction of S.
2. If $\forall i, \sum_{j} \hat{s}_{ij} = \eta$ and $\forall j, \sum_{i} \hat{s}_{ij} = \eta$, then go to step 10; otherwise, go to step 3.
3. Create a matrix $\hat{D} = (\hat{d}_{ij})$ such that

$$\hat{d}_{ij} = \begin{cases} d_{ij} & , \hat{s}_{ij} \neq 0 \\ 1 & , \hat{s}_{ij} = 0 \end{cases}$$

4. Create a list Ω with the elements of $\hat{D}$ in descending order.
5. Making ω the first element in Ω.
6. Make r and c equal to the correspondent indices of ω in $\hat{D}$.
7. If $\sum_{j} \hat{s}_{rj} < \eta$ and $\sum_{i} \hat{s}_{ic} < \eta$, make $\hat{s}_{rc} = \hat{s}_{rc} + 1$
8. If ω is the last element in Ω, go to step 2; otherwise go to step 9.
9. Make ω the next element in Ω, go to step 6.
10. Output $\hat{S}$

3.3.4. 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 the targeted residual error and N denotes the port count of switches. It is therefore important to ensure that *k* is as small as possible.

Figure 3.2 shows the number of iterations *k* required for the algorithm to converge with given residual error ε. It is observed that the slope of the curve decreases with the increase of *N*, which means the number of iterations does not increase as fast as the port count does and, therefore, the alternating projections method may have a lower computation complexity than the max-min fairness method [Yim2004].

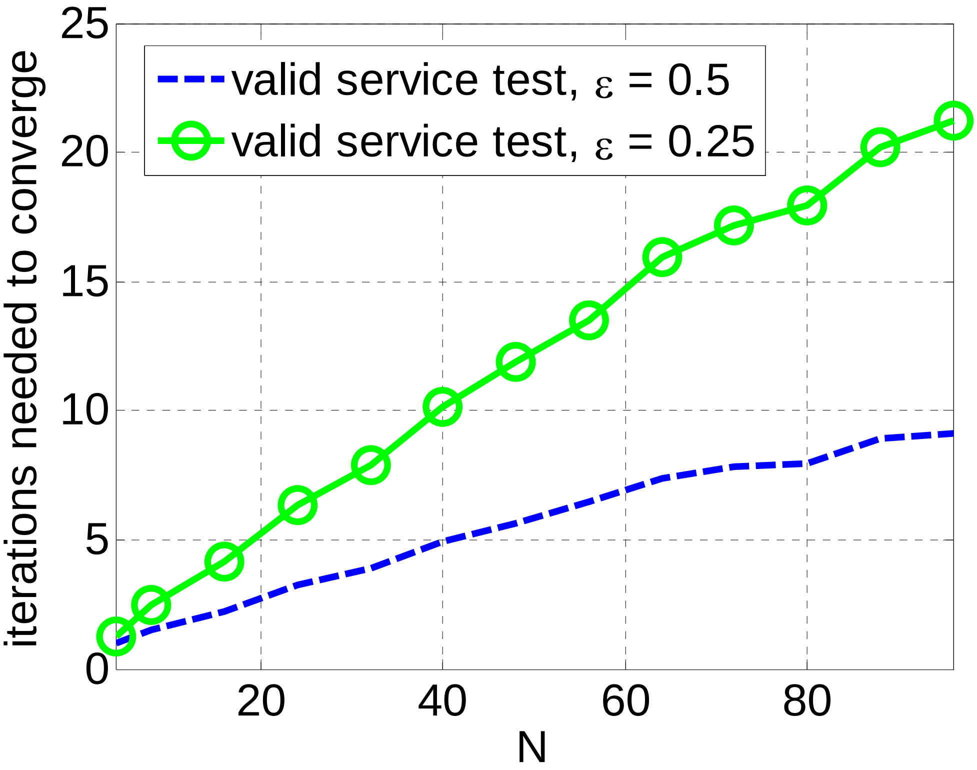


Figure 3.2: The number of iterations required for the algorithm to converge

3.3.5. Similarity

In the AAPN, it is required that the constructed service matrix has 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. It is proposed to use Definition 3 to measure this similarity.

Simulation studies have been done in order to investigate the similarity between constructed service matrix and traffic matrix. A number of traffic matrices are first constructed according to a trace generated from the bursty traffic model (see Appendix). Then the max-min fairness method and the projection method are used, respectively, to construct the corresponding real-valued service matrix and such a matrix is then transformed to a valid integer-valued service matrix. Finally, the similarity is measured between the traffic matrix and the constructed service matrix. The experiment is repeated 100,000 times.

Figure 3.3 shows the similarity of the projection versus the number of iterations in bursty traffic with an offered load of 0.8. 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).

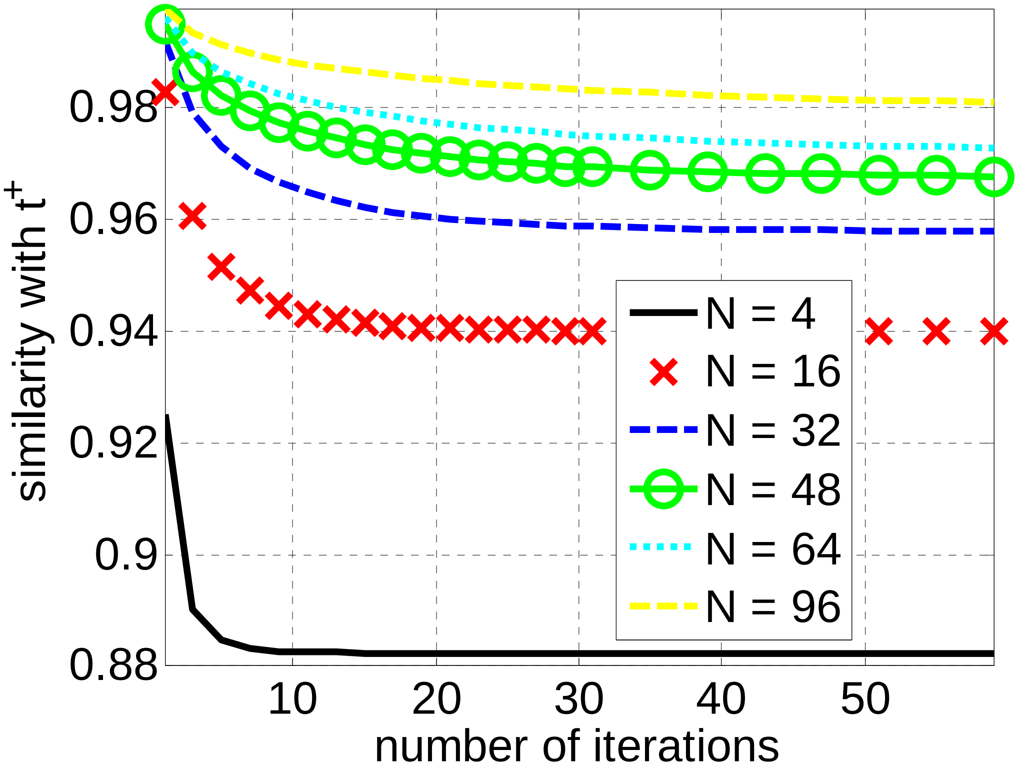


Figure 3.3: Similarity of the obtained real-valued service matrix with the original traffic matrix as a function of the number of iterations

Figure 3.4 shows the similarity versus switch dimension N between the given traffic matrix and the real-valued service matrix constructed by the max-min fairness and the projection methods. It is shown that the similarity obtained with the projection method is higher than that obtained with the max-min fairness method. An observation is that the similarity decreases when the residual error ε decreases. This is because as the residual error ε decreases, the algorithm needs more number of iterations and thus $\mathbf{t}^+$ (see Figure 3.3) 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).

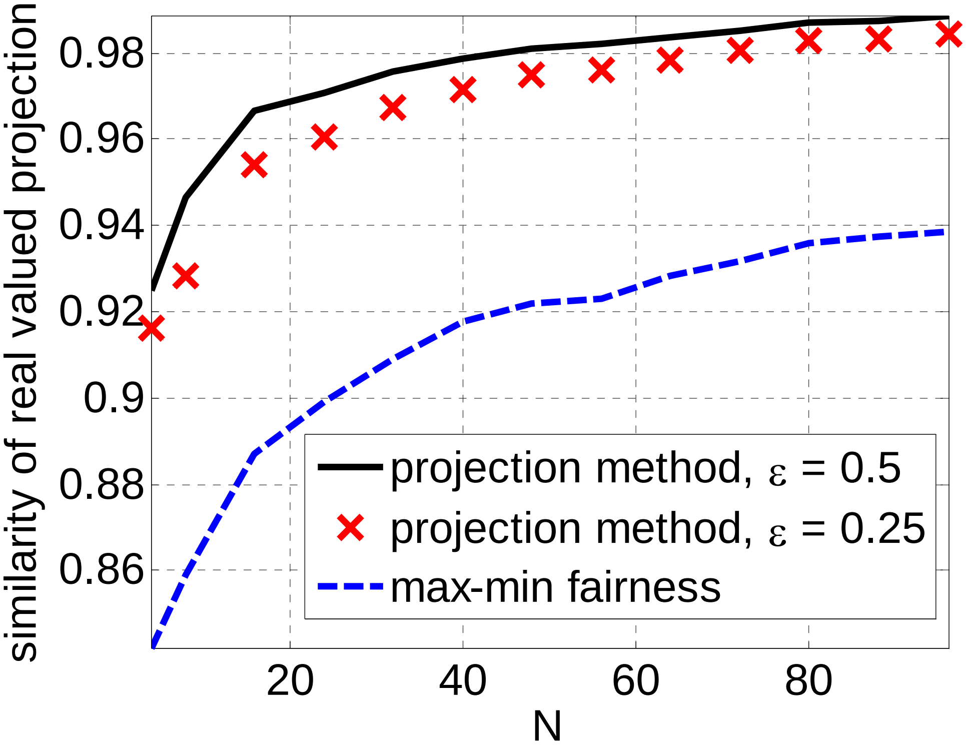


Figure 3.4: similarity versus switch dimension N between the given traffic matrix and the real-valued service matrix constructed by the max-min fairness and the projection methods

Figure 3.5 shows the similarity versus switch dimension *N* between the given traffic matrix and the integer-valued valid service matrix. Recall that the real-valued service matrix constructed by the max-min fairness and the projection methods must be quantized to an integer matrix and transformed to a valid service matrix through filling algorithm. Compared with Figure 3.4, it is shown that the quantisation and filling will slightly make the similarity move away from its real-valued solution (i.e. the real-valued service matrix) but not by much.

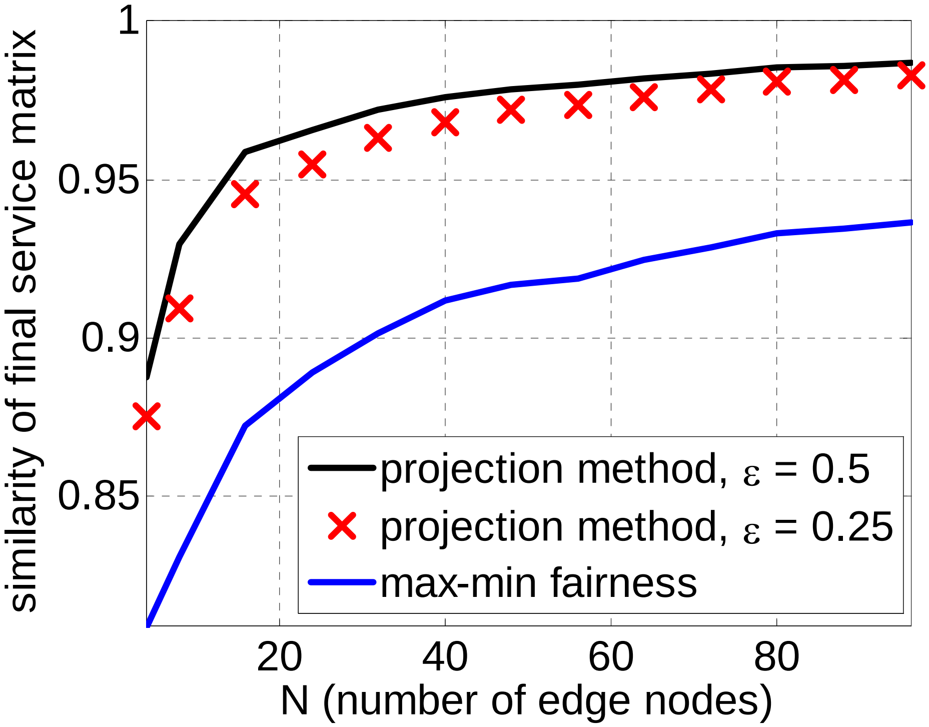


Figure 3.5: similarity versus switch dimension *N* between the given traffic matrix and the integer-valued valid service matrix constructed by the max-min fairness and the projection methods

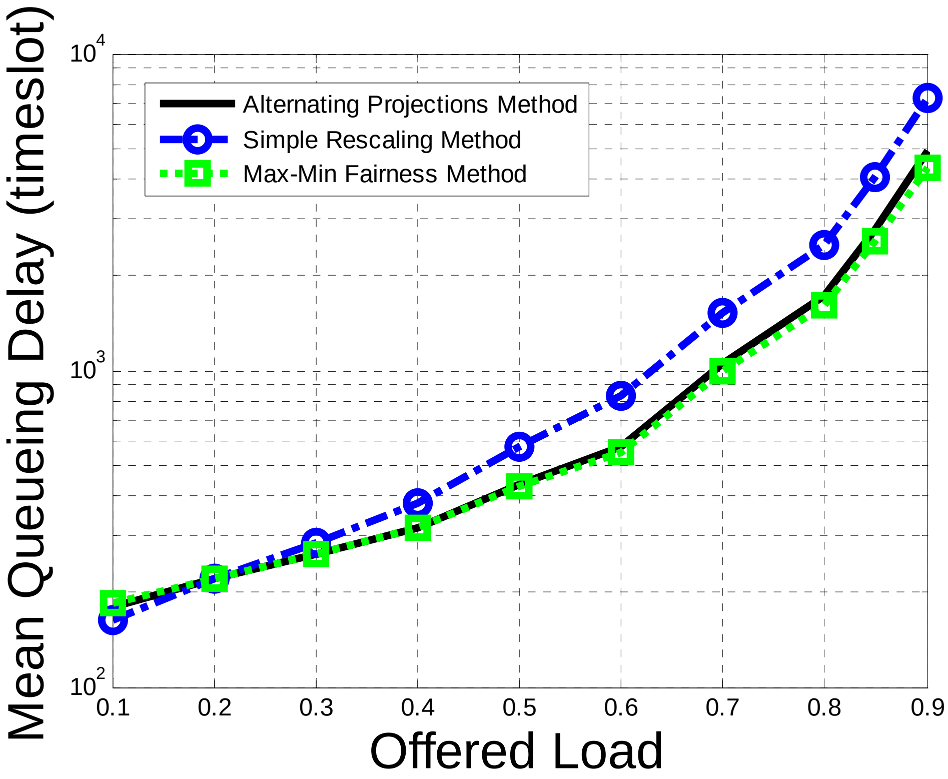
3.3.6. Network Performance

The network simulated is a single AAPN star (one wavelength) with 10Gbps links and 16 edge nodes ($N = 16$). The offered traffic is generated according to the bursty traffic model (Section 4.2 of Chapter 2) with parameters $\alpha_{ON} = 1.2, \beta_{ON} = 1, \alpha_{OFF} = 1.4$.

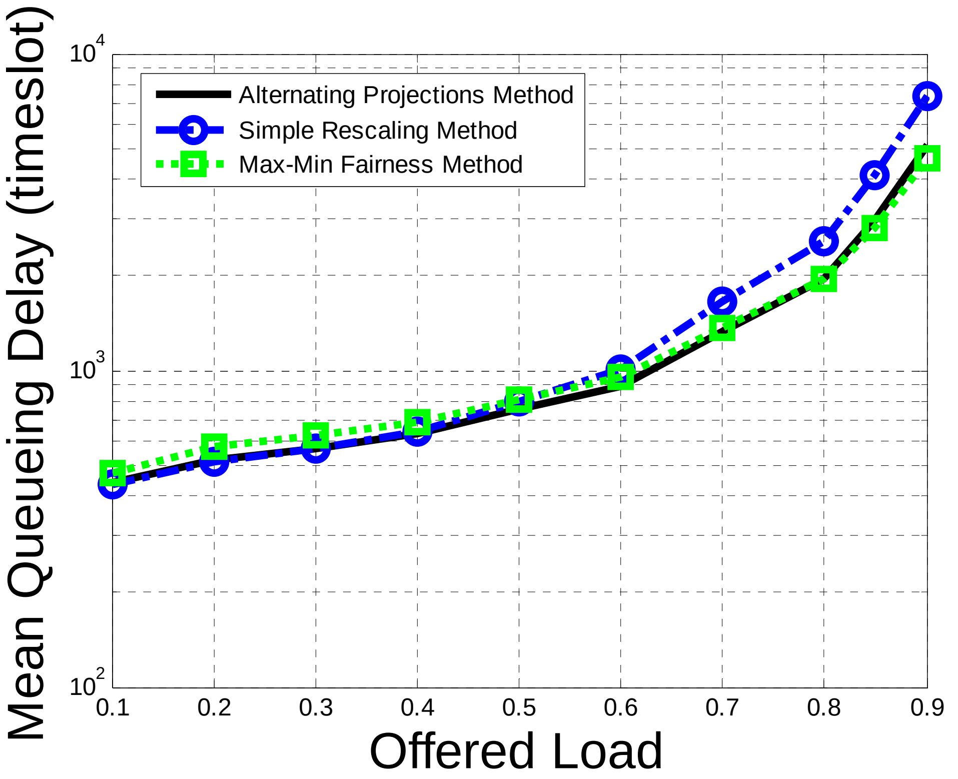
For the bandwidth allocation method at the core node, the alternating projections method (with the parameter $\varepsilon = 0.25$), simple rescaling method and max-min fairness method are used to construct service matrices in terms of traffic demand matrices. The filling algorithm is then applied to make valid integer-valued service matrices. Based on these valid service matrices, the "EXACT" method[1] described in [Towles2003] is used to decompose the integer service matrix into η permutations.

Figure 3.6 shows the delay performance of an AAPN for the three service matrix construction methods with a frame size η=128. Figure 3.6a corresponds to the metro-access scenario and Figure 3.6b corresponds to a long-haul scenario. This figure shows that the projection method performs better than the simple rescaling method and similar to the max-min fairness method and have the advantage of being faster to compute.

[1] EXACT method is a benchmark decomposition method since it may fully decompose a service matrix into permutations.


Mean Queueing Delay (timeslot)
10^4
10^3
10^2
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Offered Load
Alternating Projections Method
Simple Rescaling Method
Max-Min Fairness Method

(a)



(b)

Figure 3.6: Mean queuing delay in an AAPN as a function of the offered load for those methods of service matrix calculations. a) Metropolitan scenario (e.g. edge-core distances are 20 km), b) Long-haul scenario (e.g. edge-core distances are 1000 km

3.4. Summary

The bandwidth allocation in an AAPN is 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. The service matrix exhibits a high measure of similarity (in terms of correlation) with the original traffic matrix 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 similar delay performance compared to the max-min fairness method but the complexity is lower.

4. Birkhoff-von Neumann Decomposition based Scheduling

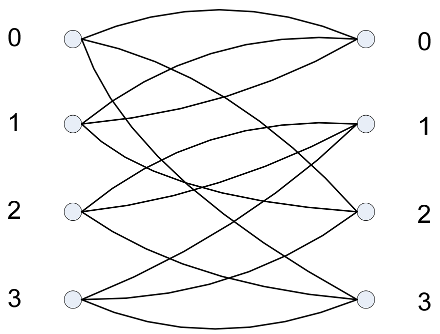
An integer-based BvN decomposition approach can adapt to the dynamically changing traffic more flexibly than the real-valued one due to its fixed frame size. The approach is based on integer-valued service matrices, i.e., η -server service matrices. A η -server service matrix S can be viewed as a bipartite graph G by applying

$$\forall i, j, \begin{cases} \text{k edges on G between vertex i and j}, & \text{if } \mathrm{s_{ij}} = k > 0 \\ \text{no edges on G between vertex i and j}, & \text{if } \mathrm{s_{ij}} = 0 \end{cases} . \qquad (4.1)$$

For example, suppose we have a 3-server service matrix

$$S = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 2 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

Then, the corresponding bipartite graph is



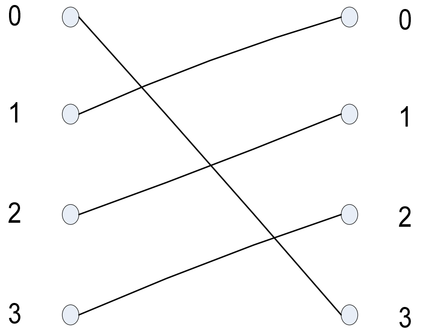
Similarly, a permutation matrix P can be viewed as a bipartite graph matching G_M by applying

$$\forall i, j, \begin{cases} \text{one edge on } \mathrm{G_M} \text{ between vertex i and j}, & \text{if } \mathrm{p_{ij}} = 1 \\ \text{no edges on } \mathrm{G_M} \text{ between vertex i and j}, & \text{if } \mathrm{p_{ij}} = 0 \end{cases} \quad (4.2)$$

For example, suppose we have a permutation matrix

$$P = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Then, the corresponding bipartite graph matching is



The problem of the integer-based BvN decomposition approach is therefore transferred to a bipartite matching problem, i.e., decomposing a given bipartite graph G into bipartite graph matchings $\{G_M\}$. One concern of the bipartite matching is the high computation complexity, which may restrict its application to high-speed switching environment. The problem can be solved by using heuristic algorithms instead of exact ones.

Towles et al. introduced a BvN decomposition approach to input-queued switch scheduling [Towles2003]. Towles introduced three different approaches, EXACT [Cole1982], MIN and DOUBLE. The latter two are heuristics. The time complexity of these three approaches is O(Nηlogη) (η denotes the frame size), $O(N^{3.5})$ and $O(N^2 \log N)$, respectively.

Paredes [Paredes2005] uses an edge-coloring scheme to perform the BvN decomposition with time complexity O(Nηlogη) where η denotes the frame size. The algorithm described by Paredes is restricted to the case that η is a power of two; however an algorithm with the same time complexity order due to Cole et al. [Cole2001] may be used for η that is not a power of two. Cole's algorithm is sophisticated and, although it achieves a time complexity that scales with $O(N\eta\log\eta)$, it has a large overhead.

Keslassy et al [Keslassy2003] proposed a simple heuristic approach, called Greedy Low Jitter Decomposition (GLJD), to reduce the overhead. GLJD can only find partial permutation matrices. In the worst case, the time complexity is $O(N^3)$.

4.1. Quick Birkhoff-von Neumann Decomposition Algorithm

In this section, a simple and efficient BvN-decomposition based heuristic, called *Quick BvN decomposition algorithm (QBvN)*, is presented. The motivation for QBvN is that, given a service matrix, finding partial permutation matrices instead of perfect permutation matrices, the combination of which is close to the service matrix, may be acceptable for high-speed agile all-optical cores provided that a low time complexity can be achieved.

4.1.1. The Basic Idea of QBvN Algorithm

The general idea of QBvN is to decompose a given bipartite graph G into bipartite graph matchings $\{G_M\}$ in the following way: For a $N \times N$ switch, the algorithm visits all N input ports on G one-by-one to generate a G_M; for each input port i, i=0, 1, .., N-1, the following two steps are applied:

i. *Matching* obtains all augmenting edges incident on port i. An edge <i, j> on G is said to be an *augmenting edge* if both port i and j on G_M are not yet part of the matching.

ii. *Selection* chooses one of the augmenting edges with the smallest output port number and moves it from G to G_M.

After all N input ports are visited, no augmenting edges can be added into G_M and thus G_M is a maximal bipartite graph matching. Therefore its associated partial permutation is constructed. This procedure is repeated η times to find the first η partial permutations.

Note[1] that QBvN changes the visiting sequence of the input ports in a round-robin fashion before generating the next G_M.

4.1.2. Computation Complexity

The complexity of QBvN depends on the implementation of the matching and selection.

The matching can be implemented as a *Boolean-vector-matching* procedure. Each input port i, i=0, 1, …, N-1, keeps a vector with N bits, called L_i, and each bit corresponds to an output port. If there exist edges from the input port i to output port j on the bipartite graph G, the jth bit of L_i is set to 1 (to 0 otherwise). For each bipartite graph matching G_M, there is also a vector with N bits, called L_{free}, and each bit corresponds to an output port. If the output port j is free on the G_M, the jth bit of L_{free} is set to 1 (to 0 otherwise). An output port on G_M is said to be *free* if the output port is not yet part of the matching. The vector L_{free} marks all free output ports ready for current matching on G_M. The Boolean-vector-matching procedure may thus obtain all augmenting edges incident on port i by applying the following operation:

$$L_M = L_i \text{ AND } L_{free}$$

Here L_M is also a vector of N-bits and each bit indicates an output port. The edge <i, j> is an augmenting edge incident on port i if the jth bit of L_M is 1. The notation "AND" denotes a bitwise "and" operation. Here we assume the operation AND takes O(1) time, which is

[1] Karp [Karp1990] proposed a simple randomized algorithm for on-line bipartite matching. Karp's approach is different from QBvN in that (1) Karp's approach randomly selects the augmenting edges and (2) it has a fixed visiting sequence of the input ports for each matching.

true if N is smaller than the effective word length of the computer (in number of bits). Thus the time complexity of matching is O(1).

The selection can be formulated as finding the index of the least significant “1” bit (the lowest “1” bit) in L_M. The index denotes the output port with the smallest number joining an augmenting edge.

The selection procedure is divided into two steps. The first step is to find the least significant “1” bit; the second step is to find the index of that “1” bit. We use the operation

$$L_{least} = L_M \text{ AND } (-L_M)$$

to obtain an N bit vector L_{least} where only the position of the least significant “1” bit is marked by “1”. Note that $-L_M$ is the 2’s complement of L_M. Then we perform the following two operations

$$L_{least} = L_{least} - 1$$

popcount

to find the index of the least significant “1” bit. The assembly instruction *popcount* is used to count the number of “1” bits in L_{least}, which is exactly the same as the index of the least significant “1” bit after the operation $L_{least} = L_{least} - 1$. The instruction *popcount* can be finished in one cycle. Hence, the time complexity of the selection procedure is O(1) under the above assumption about the word length of the computer.

The time complexity of QBvN therefore scales as $O(N\eta)$ because it needs to repeat the matching-selection procedure $N\eta$ times to find η matchings.

4.1.3. Comparison of Decomposition Methods

The delay performance of difference decomposition methods is compared in this section. The network simulated is a single AAPN star (one wavelength) with 10Gbps links and 16 edge nodes ($N = 16$). The frame size is specified as 128 timeslots. Two scenarios, Metropolitan Area Network (MAN) and Wide Area Network (WAN), are studied. The MAN is defined as an AAPN network with 20km optical links. The WAN is defined as a national wide AAPN network with 1000km optical links. The offered traffic is generated according to the bursty traffic model (Section Appendix) with parameters $\alpha_{ON} = 1.2, \beta_{ON} = 1, \alpha_{OFF} = 1.4$.

For the bandwidth allocation method at the core node, the alternating projections method (see Section 3.3.3) is used to construct service matrices in terms of traffic demand matrices with the parameter $\varepsilon = 0.25$. The filling algorithm (see Section 3.3.6) is then applied to obtain valid integer-valued service matrices. Based on these valid service matrices, the different decomposition algorithms are compared and investigated.

We have identified two other algorithms, i.e. EXACT method and GLJD method, for comparison with the QBvN method. The EXACT provides an exact BvN decomposition from a service matrix. It has a high computation complexity and is therefore not very applicable in reality. The GLJD provides the least number of different switch configurations within a frame with relatively low computation complexity.

4.1.3.1. Delay Performance

The delay performance of different scheduling algorithms is shown in Figure 4.1a (corresponding to the metropolitan scenario) and Figure 4.1b (corresponding to a long-haul scenario). These figures show that QBvN performs better than GLJD and is very close to the exact decomposition method. In addition, it has the advantage of being faster to compute.

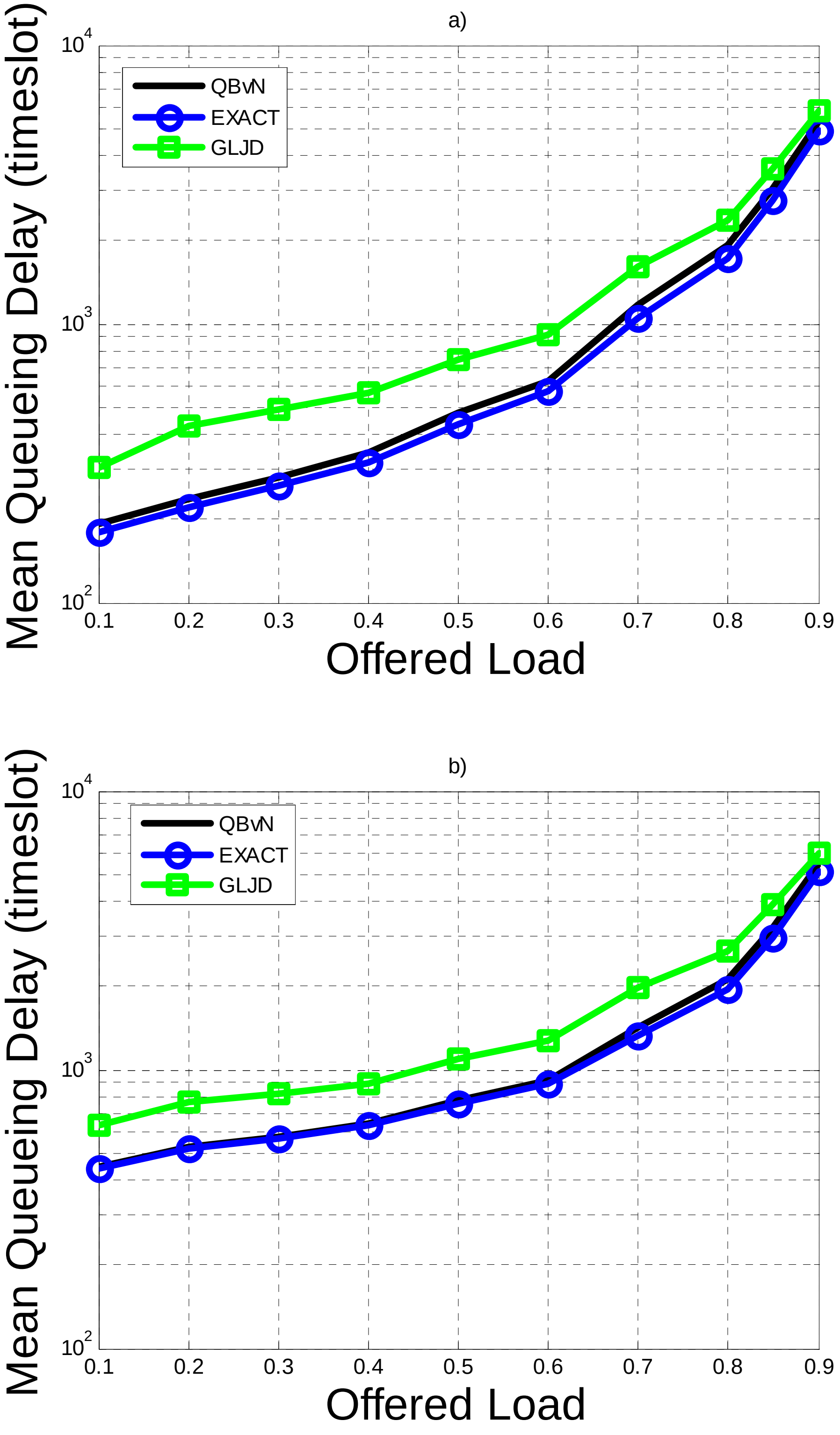


Figure 4.1: Comparison of delay performance of different decomposition methods (N=16, η =128). a) Metropolitan scenario b) Long-haul scenario

4.1.3.2. Queue Buffer Size

Figure 4.2 shows the maximum size of buffer required at the source edge node. The simulated network is a single AAPN star (one wavelength) with 10Gbps links and 16 edge nodes ($N = 16$). The traffic model is the same as used in the previous section. The duration of a timeslot is 10 microseconds. The traffic transmitted through a timeslot, i.e. the volume of a slot, is 100K bits. When the load is equal to 0.9, the buffer size needed at the source edge node is, for example, 1.04G bits.

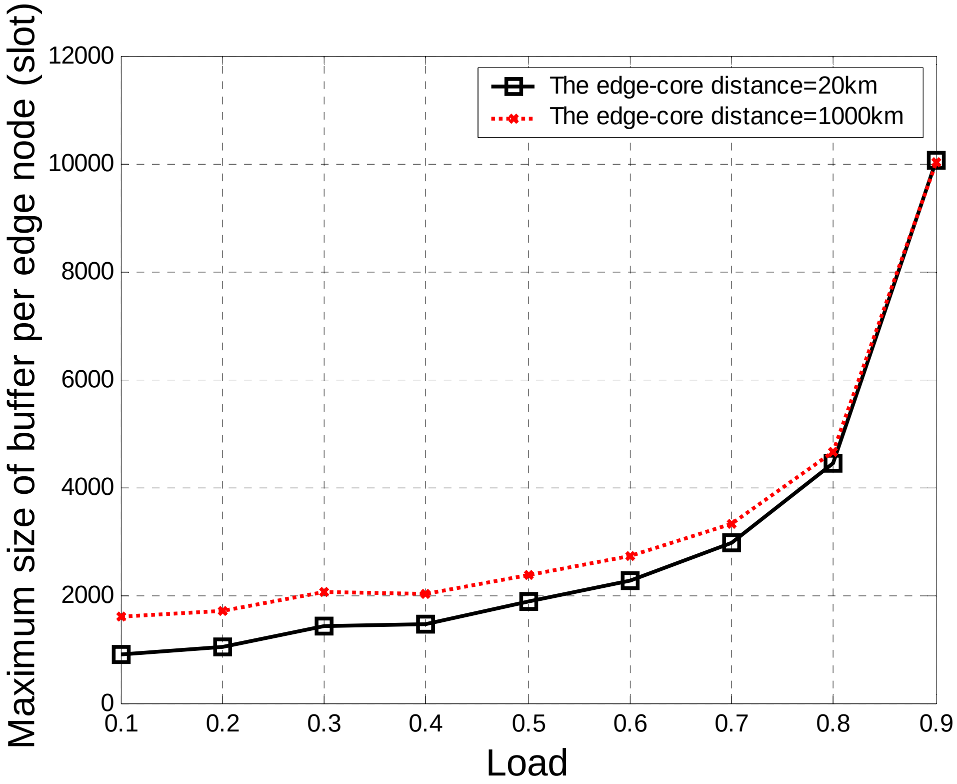


Figure 4.2: The maximum size of buffer consumed at the source edge node

4.1.3.3. Response to Fast Traffic Changes

The response time[1] of the bandwidth allocation scheme to traffic changes describes how fast the generated switch configurations adapt to sudden traffic changes, e.g. the traffic changes from uniform to non-uniform. This response time is influenced by the propagation delay (that is, signaling time between edge node and core controller), the traffic estimation and the adaptation of the switch configurations (that is, whether the switch configurations closely reflect the traffic changes).

In the simulation, the distance between the edge nodes and the core nodes is set to 20 kilometers and the change of the traffic distribution from uniform to non-uniform occurs at the 50000th timeslot. The average traffic load is set to 0.4.

Figure 4.3 shows the queuing delay of the slots when the traffic distribution changes at the 50000th timeslot. The traffic has low burstiness because it is generated according to the bursty traffic model with parameters $\alpha_{ON} = 1.8, \beta_{ON} = 1, \alpha_{OFF} = 1.4$. It is shown that the queuing delay increases to a peak and then starts to decrease to a stable value. The curve can be divided into three parts: the left part (earlier than the 50000th timeslot), the middle part (between the 50000th and the 50500th timeslot) and the right part (later than the 50500th timeslot). The left and right parts denote the situation that the scheduler is in equilibrium with traffic, respectively. For the middle part, an increase of queuing delay followed with a fall is observed. The reason is that the scheduler cannot immediately notice the traffic pattern changes (because of the signaling delay). Therefore, the generated switch configurations are not efficient enough to serve the traffic. However, when the bandwidth demand for the new traffic (e.g. traffic after the 50000th timeslot) arrives at the core node,

[1] The response time may be used to describe whether the scheme is stable against traffic changes.

the scheduler generates the switch configurations that are adaptive to the traffic. Hence the queuing delay starts to decrease.

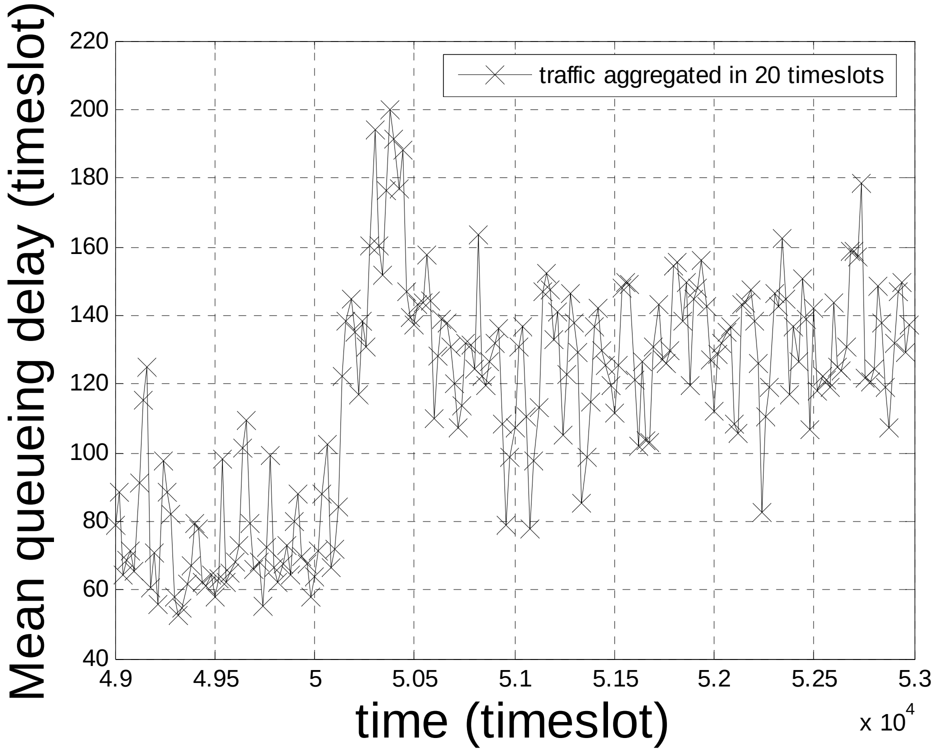


Figure 4.3: The response time to traffic.
$(\alpha_{ON} = 1.8, \alpha_{OFF} = 1.4, \beta_{ON} = 1)$

4.2. Frame-based Bandwidth Allocation Scheme with QoS Guarantee

4.2.1. QoS Service Models

Traditional IP networks, i.e. Internet, only support best-effort services. They are transforming into commercial broadband multi-service IP networks that require quality of service (QoS) support. The QoS-support is one of the key “must-have” properties of the so-called next generation networks (NGN) that seamlessly blend the public switched telephone network (PSTN) and the public switched data network (PSDN), creating a single multi-service network. To ensure QoS in the NGN, IETF (Internet Engineering Task Force) proposed two service models, namely *Integrated Service* (IntServ) and *Differentiated Service* (DiffServ).

The basic idea of the IntServ model lies in the resource reservation in terms of QoS requirements at every network element along the path before the transmission of data flows. It can thus meet the demand for end-to-end QoS, i.e. guaranteed service, over heterogeneous networks by applying the Res*ource Reservation Protocol* (RSVP) to each flow of data [RFC2210]. The guaranteed service [RFC2212] provides firm (mathematically provable) bounds on bandwidth and makes it possible to provide a service that guarantees delay. The IntServ has the advantage of being able to provide means of delivering end-to-end QoS guarantee, but it has difficulties to be implemented in reality because all flow-state information should be recorded in the network elements along the path, leading to the poor scalability for practical deployment in large networks.

To overcome the scalability issue of IntServ model, the DiffServ model has been proposed and standardized. The approach defines different classes of service (CoS) and an application may select the service class it will use. The user has to negotiate a Service Level Agreement (SLA) with the network service provider which states what maximum bandwidth will be available for each service class. The DiffServ model is flexible and scalable for managing traffic in terms of CoS but in most cases it does not provide guarantee of specific QoS parameters for any of the service classes (only relative performance is guaranteed).

4.2.2. Frame-based Bandwidth Allocation Scheme with QoS Guarantee

The QoS guaranteed frame-based bandwidth allocation scheme (QoS-FBAS) approach can be adopted to support guaranteed QoS, i.e. guaranteed bandwidth by using the IntServ model. The approach has two classes of queues at the edge node, i.e. the high priority queue and the low priority queue. The high priority queue stores the traffic that requires guaranteed bandwidth, i.e. guaranteed traffic, while the low priority queue stores the best effort traffic where there is no guarantee as to timeliness or actual delivery [Xiao1999]. The RSVP is used to reserve bandwidth for guaranteed traffic while the signaling protocol discussed in Section 3.2.2 is used for the best effort traffic (it is in the low priority queue). An admission control mechanism is equipped at the high priority queue to ensure that the total bandwidth of the guaranteed traffic that has been served is necessarily smaller than the available bandwidth.

For the guaranteed traffic, the bandwidth request is sent when the state of the high priority queue changes, i.e. a new flow of data request arrives at the queue. When the core node receives the bandwidth request (e.g. a timeslot per frame) from a given source edge node to a destination edge node, it checks out whether there exists enough available bandwidth in a future frame. If it cannot find enough bandwidth, the bandwidth request will be rejected; otherwise, the bandwidth will be granted.

For the best effort traffic, the bandwidth request is sent frame by frame. The methods discussed in Chapter 3 and 4 (i.e. traffic demand matrix construction, alternating projections method and QBvN) are used to allocate the remaining free bandwidth, i.e. the bandwidth that has not been granted to the guaranteed traffic.

4.3. Bandwidth Allocation for AAPN with Concentrators/Distributors

The scalability of the AAPN is limited by a restriction on the number of ports of the core switches. In order to improve the scalability, a device, called *concentrator/distributor*, is employed in [Maier2002] [Kamiyama2005]. For the AAPN with concentrators/distributors (Chapter 2 of Section 1.2.2), the bandwidth allocation scheme employed at the core node can be employed by two possible bandwidth allocation schemes.

The first scheme constructs a service matrix in dimension of the number of edge nodes, e.g., for a $N \times N$ AAPN core node with $M \times 1$ concentrators, the dimension of the service matrix is $N \cdot M$ which is equal to the maximum number of edge nodes.

The main issue of this scheme is the scalability due to the computation complexity. To overcome the issue, a two-step allocation is adopted in the second scheme. The first step allocates bandwidth (i.e. timeslots) between concentrators and distributors and the second step determines which edge node is allowed to transmit a slot in a given timeslot.

For a $N \times N$ AAPN core node with $M \times 1$ passive concentrators, it is defined that the k -th ($0 \le k < MN$) edge nodes connects to the $\lfloor k / M \rfloor$th concentrator. The second scheme can thus be described as follows:

(1) The traffic demand information between edge nodes is signaled to the core nodes every frame;

(2) The corresponding $MN \times MN$ traffic demand matrix T is constructed.

(3) To allocate timeslots between concentrators, the $MN \times MN$ traffic demand matrix T should be converted to a $N \times N$ traffic demand matrix between concentrators $\tilde{T}$, i.e. the entry $\tilde{t}_{ij}$ ($0 \le i, j < N$) in $\tilde{T}$ is equal to $\sum_{m=i\cdot M}^{(i+1)M-1} \sum_{n=j\cdot M}^{(j+1)M-1} t_{mn}$;

(4) Construct service matrix $\tilde{S}$ in terms of the traffic demand matrix $\tilde{T}$ (as described in Chapter 3); the value of $\tilde{s}_{ij}$ ($0 \le i, j < N$) in $\tilde{S}$ denotes the number of slots that can be transmitted from concentrator i to j during the frame.

(5) Decompose the service matrix to a number of switch configurations, one for each timeslot (the switch configuration denotes the bandwidth allocation between concentrators);

(6) These $\tilde{s}_{ij}$ slots can be allocated among the traffic demands $\{t_{mn}\},(iM \le m < (i+1)M, jM \le n < (j+1)M)$ by various methods, such as max-min fairness and so on. The Max-Min notion of fairness is based on the following premises: (a) no user should receive an allocation larger than its demand unless its demand is admissible; and (b) increasing the allocation of any entity should not result in the decrease of the allocation of another user that received an equal or smaller allocation [Bertsekas1992]. By this means, the edge node that is allowed to transmit a slot in given timeslot is determined.

4.4. Summary

In this chapter, a simple and efficient scheduling approach, called Quick Birkhoff-von Neumann decomposition method is proposed, to configure bandwidth to different edge node pairs. It is shown that the computation complexity of the method is as low as $O(NF)$ where N denotes the port count of the switch and F denotes the size of a frame. The QoS-QBvN method is also proposed that provides guaranteed services (i.e. bandwidth guarantee) for guaranteed traffic (i.e. high priority traffic). For the AAPN with concentrators/distributors, a two-step bandwidth allocation scheme is proposed for edge node pairs.

5. Routing in Multi-Core Agile All-Photonic Networks

5.1. The Multi-Core AAPN

The AAPN may contain several core nodes to form an overlaid star topology (Figure 1.1). This topology connects edge nodes together via several core nodes. As shown in [Vickers2000] and [Blouin2002], the overlaid star topology can compare favourably with mesh architectures. The overlay of several stars provides robustness in the case of link or core node failure [Boch2004]. However, data traversing the network only passes through one photonic switch, resulting in a major simplification of the control problem of ensuring that contention is rare. From the control point of view, each star can be managed independently of the others because there is no data interaction. For each star, resource allocation is concentrated at a single point in the network.

The bandwidth is shared in the time domain over each wavelength on each fiber. The minimum bandwidth unit is a timeslot. Every F timeslots are grouped together to form a fixed-size frame. The term "*call*" is used to describe a connection with a certain bandwidth guarantee (i.e. a number of timeslots reserved per frame for a period of time) during a period of time. For example, a call may be a telephone call in the telephony networks where 64kbps bandwidth is guaranteed.

For the QoS-guaranteed frame-based bandwidth allocation scheme approach (see Section 4.2.2), the traffic is classified into two categories: bandwidth-guaranteed traffic and best-effort traffic. The bandwidth-guaranteed traffic may be differentiated as various data flows (i.e. calls) between edge nodes where required bandwidth has been granted by the network. When a new call arrives at the edge node, a connection establishment request with a given bandwidth demand is sent to the core node. The scheduler at the core node may establish a connection in response to the request if there are enough free timeslots in a given frame; otherwise, the request will be rejected. When a call is finished, a connection release request is sent to the core node and the corresponding connection is released.

For the multi-core AAPN, there exist multiple cores and thus multiple paths between edge nodes. The blocking of a new arriving call depends on the available bandwidth on a chosen path (each path via different core node) to the destination. Consequently, it is important to investigate routing strategies in the AAPN so that the call blocking probability remains as low as possible.

5.2. Routing Strategies

5.2.1. Random Routing Strategies

In the *random routing strategy*, for each connection request, the source node randomly selects one among all paths to the destination in a uniform manner and then uses a signaling protocol to establish a connection. If the connection is not established successfully, the request is dropped. This strategy is simple to implement but it does not take into account link state information and thus may not be able to achieve the best performance. A variant of this strategy is called *random-with-retrying*, which introduces retrying in path selection. In the event of an unsuccessful establishment, the source node randomly selects another path among the remaining paths, which would significantly reduce the blocking probability of the network. The request will be dropped if all paths are tried with no success.

5.2.2. Least-Congested-Path Routing Strategy

5.2.2.1. Overview

In the least-congested-path routing strategy, path selection is based on the current timeslot usage on the two links of a path. For each connection request, the source node selects the least congested path among all paths to the destination, making the timeslot usage on each link more balanced and the network performance improved. The congestion of a path is defined as the number of timeslots available on the most congested link of the path. The congestion of a link is measured in terms of the number of timeslots available on the link. The fewer the number of available timeslots, the more congested the link.

To support this strategy, each edge node must maintain the state (i.e. the free bandwidth available on each path) information for all paths. This information should be advertised and updated by each core node periodically using a signaling protocol, typically one time each frame. The link state information should contain the timeslot usage during the frame. Note that the link state information used to make a path selection by the source may be outdated because of the propagation delay on each link, which would affect network performance.

Compared with the random strategy, this strategy is more complex to implement because it requires the core nodes to advertise and update the link state information periodically and the source nodes to compute a path based on the link state information it maintains.

5.2.2.2. Analytical Model for Blocking Probability

In this section, the blocking probability of the *least-congested-path routing strategy* is studied and an analytical model is proposed. In the literature, a variety of analytical models have been proposed to compute the call blocking probability with different network constraints, traffic models, and routing and wavelength assignment algorithms. For example, Kovacevic et al. proposed an approximate analytical model in [Kovacevic1996] to compute the blocking probability of an all-optical network both with and without wavelength conversion. This model, however, only considered static routing and did not consider the load correlation between successive links of a path. In [Barry1996], Barry et al. proposed an analytical model to compute the blocking probability of a multi-hop path in all-optical networks, taking wavelength correlation into account. However, this model makes more simplistic traffic assumptions and does not take into account the dynamic nature of network traffic. The model proposed by Birman [Birman1996] uses a reduced load approximation method with state-dependent arrival rates to compute the blocking probabilities with fixed routing and least loaded routing. It is good for small networks, and is applicable to arbitrary topologies and traffic patterns. However, it is computationally intensive as the complexity increases exponentially with the number of hops. In [Subramaniam1996], Subramaniam et al. proposed an analytical model that takes both dynamic traffic and link-load correlation into account and has a moderate complexity. It has been shown to be more accurate than the other models for a variety of network topologies. In [Li1999], Li et al. proposed an approximate analytical model to analyze the blocking performance of fixed-path least congestion routing and dynamic routing using neighborhood information with link-load correlation considered.

In the context of all-optical TDM networks, less work has been done for computing blocking performance. In [Yates1999], Yates et al. analyzed the blocking performance of multi-wavelength TDM networks. The importance of wavelength conversion and timeslot interchange in improving the blocking performance was investigated based on both analytical and simulation results. In [Sivakumar2004], Sivakumar et al. investigated the effects of wavelength conversion and timeslot interchange on the blocking performance of TDM wavelength routed networks based on simulation experiments. No analytical model was presented. In [Wen2002], Wen et al. studied the blocking performance of a family of wavelength and timeslot assignment algorithms proposed for TDM wavelength-routed networks. The blocking performance was analyzed based on simulation results without using any analytical model.

The analytical model for blocking probability is developed in this Section which is based on Sivakumar's [Sivakumar2004] and Girard's work [Girard1990]. To develop the analytical model, the following assumptions are made.

- Connection requests arrive at each edge node according to a Poisson process with rate λ.
- The destination of each connection request is uniformly distributed among all edge nodes except the source.

- The holding time of each connection is exponentially distributed with mean (1/μ).
- The bandwidth demand of each connection request is one timeslot.
- The number of timeslots in a frame is identical on each fiber link and is equal to F.
- A timeslot is randomly selected from a set of free timeslots on the selected path to be allocated to a connection. Note that a free timeslot on a path is defined as one that is not used on all the links of the path.

In addition to the above assumptions, it is also assumed that the number of overlaid cores in the network M is two, which is a reasonable deployment in practice. In this case, there are only two fixed alternate paths between a pair of edge nodes, each having two hops. The two paths are referred as the first path and the second path, respectively. Note that all these assumptions are also used in the simulation model.

It is assumed that the timeslots cannot be interchanged in the core node (i.e. no buffering). This means that the link load or the use of individual timeslots on consecutive links is correlated. To ensure accuracy, it is important to capture this correlation in the analytical model. At the same time, this should not make the model too complicated to be tractable. For this purpose, the correlation model is used which has been presented in [Subramaniam1996] and adapted to the network scenario considered in this work. In the adapted model, a timeslot is analogous to a wavelength in the original model in [Subramaniam1996]. It is assumed that the link loads in the network have Markovian spatial correlation, i.e., the availability of a timeslot on a particular link of a path depend only on the availability on the previous one link of the path. It is possible to further extend the correlation effects but at the expense of a more complicated model. Due to the limitation of space, only the results that are directly related to our analytical model are given. The readers may refer to [Subramaniam1996] for other details on the correlation model.

The notations used in the correlation model are defined as follows.

- $Q(w)$: the probability that w timeslots are free on a link;
- $S(y|x_p)$: the conditional probability that y timeslots are free on a link of a path, given x_p timeslots are free on the previous link of the path;
- $U(z_c|y,x_p)$: the conditional probability that z_c connections (timeslots) continue to the current link from the previous link of a path, given x_p timeslots are free on the previous link and y timeslots are free on the current link;
- $R(n|x_f,y,z_c)$: the conditional probability that n timeslots are free on a two-hop path, given x_f timeslots are free on the first hop, y timeslots are free on the second hop, and z_c connections continue from the first hop to the second hop;
- $T^{(l)}(n,y)$: the probability that n timeslots are free on an l -hop path and y timeslots are free on hop l ;

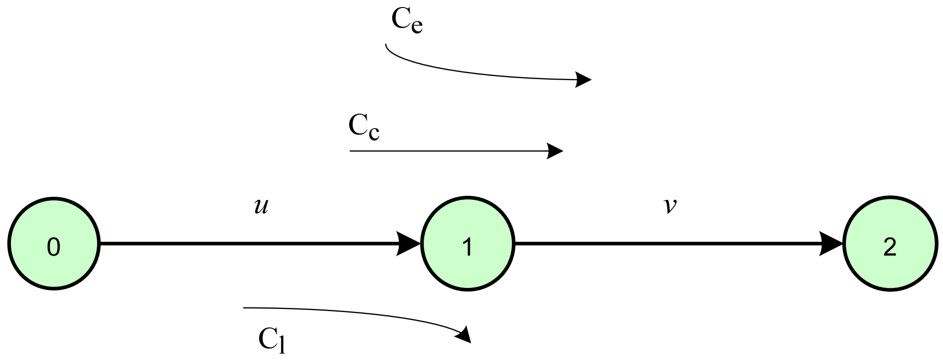


Figure 5.1: Call arriving and leaving on a two-hop path

Consider a two-hop path consisting of link u (the first link) and link v (the second link), as shown in Figure 5.1. Let C_l be the number of calls that enter link u at node 0 but leave link u at node 1, C_c be the number of calls that enter link u at node 0 and continue on to link v at node 1, and C_e be the number of calls that enter link v at node 1. Accordingly, the number of calls that use the first link is $C_l + C_c$ and the number of the calls that use the second link is $C_c + C_e$. Since the number of calls on a link cannot exceed the total number of available timeslots (*F*) in a frame, we have $C_l + C_c \le F$ and $C_c + C_e \le F$. Also, let λ_l be the arrival rate of calls that enter link u at node 0 but leave link v at node 1, λ_c be the arrival rate of calls that enter link u at node 0 and continue on to link v at node 1, and λ_e be the arrival rate of calls that enter link v at node 1. The corresponding Erlang load is denoted by $\rho_l = \lambda_l / \mu$, $\rho_c = \lambda_c / \mu$, and $\rho_e = \lambda_e / \mu$. Therefore, C_l, C_c, and C_e can be characterized by a three-dimensional Markov chain, with each state denoted by an integer triplet (c_l, c_c, c_e). The steady-state probability of state (c_l, c_c, c_e) as in [Bertsekas1992] is given by

$$\pi(c_l, c_c, c_e) = \frac{\dfrac{\rho_l^{c_l}}{c_l!}\dfrac{\rho_c^{c_c}}{c_c!}\dfrac{\rho_e^{c_e}}{c_e!}}{\sum_{j=0}^{L}\sum_{i=0}^{L-j}\sum_{k=0}^{L-j}\dfrac{\rho_l^{i}}{i!}\dfrac{\rho_c^{j}}{j!}\dfrac{\rho_e^{k}}{k!}}, \quad \begin{array}{l} 0 \le c_l + c_c \le L \\ 0 \le c_c + c_e \le L \end{array}$$

Hence, the conditional probabilities defined earlier can be derived as follows, which is similar to [Subramaniam1996].

$$R(n \mid x_f, y, z_c) = \frac{\begin{pmatrix} x_f \\ n \end{pmatrix}\begin{pmatrix} F - x_f - z_c \\ y - n \end{pmatrix}}{\begin{pmatrix} F - z_c \\ y \end{pmatrix}} \tag{5.1}$$

for $\min(x_f, y) \ge n \ge \max(0, x_f + y + z_c - F)$, and is 0 otherwise.

$$\begin{aligned} & U(z_c \mid y, x_p) \\ & = P(C_c = z_c \mid C_c + C_e = F - y, C_l + C_c = F - x_p) \\ & = \frac{\pi(F - x_p - z_c, z_c, F - y - z_c)}{\sum_{x_c=0}^{\min(F - x_p, F - y)} \pi(F - x_p - x_c, x_c, F - y - x_c)} \end{aligned} \tag{5.2}$$

$$\begin{aligned} & S(y \mid x_p) \\ & = P(C_c + C_e = F - y \mid C_l + C_c = F - x_p) \\ & = \frac{\sum_{x_c=0}^{\min(F - x_p, F - y)} \pi(F - x_p - x_c, x_c, F - y - x_c)}{\sum_{x_c=0}^{F - x_p} \sum_{x_e=0}^{F - x_c} \pi(F - x_p - x_c, x_c, x_e)} \end{aligned} \tag{5.3}$$

and

$$
\begin{aligned}
Q(w) &= P(C_l + C_c = F - w) \\
&= \sum_{x_c=0}^{F-w} \sum_{x=0}^{F-x_c} \pi(F - w - x_c, x_c, x)
\end{aligned}
\tag{5.4}
$$

The steady-state probability that n timeslots are free on an l-hop path and y timeslots are free on hop l can be recursively calculated as

$$
T^{(l)}(n,y) = \sum_{x_p=0}^{F} \sum_{x_f=0}^{F} \sum_{z_c=0}^{F'} R(n \big| x_f, y, z_c) U(z_c \big| y, x_p) \times S(y \big| x_p) T^{(l-1)}(x_f, x_p) \tag{5.5}
$$

$$
F' = \min(F - x_p, F - y)
$$

Note that the starting point of the above recursion is $T^{(1)}(n,y)$, which is given by

$$
T^{(1)}(n,y) = \begin{cases} 0 & n \neq y \\ Q(n) & n = y \end{cases}
$$

Therefore, the probability that n timeslots are free on an l-hop path is given by

$$
Q^{(l)}(n) = Q_p(n) = \sum_{y=0}^{F} T^{(l)}(n,y) \tag{5.6}
$$

A fundamental assumption made in the correlation model is that the path used by a connection does not depend on the link state on the path. For the fixed shortest-path routing, it is possible to assume that the effect of the blocking probability on the carried traffic load is negligible and the arrival rate on each link is the same in order to make the analysis simple. However, these assumptions become invalid for the least-congested-path routing. In this case, the path for a call is selected based on the current network state. As a result, the arrival rate on each link is dynamically changing. To address this problem, we use a method based on the Erlang fixed-point method for alternate routing [Girard1990]. To describe this method, the following notations need to be defined.

- $R_u^{(1)}$ ($R_v^{(1)}$): the set of all first-paths that pass through link u (link v);

- $R_u^{(2)} / R_v^{(2)}$: the set of all second-paths that pass through link u /link v;
- $R_{u,v}^{(1)}$: the set of all first-paths that pass through link u and link v ;
- $R_{u,v}^{(2)}$: the set of all second-paths that pass through link u and link v ;
- $P_1(p_{ij}^1)$: the probability that a connection between node i and node j is established on the first path, p_{ij}^1 ;
- $P_2(p_{ij}^2)$: the probability that a connection between node i and node j is established on the second path, p_{ij}^2 .

Using the least-congested-path routing, a connection is established on the path with more free timeslots. Hence, if the first path has more free timeslots than the second path, it is selected for the connection. Otherwise, the second path is selected if there is at least one free timeslot on the path. Therefore, we have

$$P_1(p_{ij}^1) = \sum_{\alpha=1}^{L} Q_{p_{ij}^1}(\alpha) \sum_{\beta=0}^{\alpha} Q_{p_{ij}^2}(\beta) \tag{5.7}$$

$$P_2(p_{ij}^2) = \sum_{\alpha=1}^{L} Q_{p_{ij}^2}(\alpha) \sum_{\beta=0}^{\alpha-1} Q_{p_{ij}^1}(\beta) \tag{5.8}$$

The arrival rate of calls that enter link u and continue to link v is given by

$$\rho_c(u,v) = \begin{cases} \lambda P_1(p_{ij}^1) & \mathit{if}\ \ u = e_{i1},\ v = e_{1j} \\ \lambda P_2(p_{ij}^2) & \mathit{if}\ \ u = e_{i2},\ v = e_{2j} \\ 0 & \mathit{otherwise} \end{cases} \tag{5.9}$$

The arrival rate of the calls that leave link u, which include the calls that use link u in the first or second path but do not continue to link v, is given by

$$\rho_l(u,v)=\begin{cases}\sum_{p_{ij}^1\in R_u^1}\lambda P_1(p_{ij}^1)-\rho_c(u,v) & if\ u=e_{i1}\\ \sum_{p_{ij}^2\in R_u^2}\lambda P_2(p_{ij}^2)-\rho_c(u,v) & if\ u=e_{i2}\\ 0 & otherwise\end{cases}\tag{5.10}$$

The arrival rate of the calls that enter link v, which include the calls that use link v in the first or second path but do not continue from link u to link v, is given by

$$\rho_e(u,v)=\begin{cases}\sum_{p_{ij}^1\in R_v^1}\lambda P_1(p_{ij}^1)-\rho_c(u,v) & if\ v=e_{1j}\\ \sum_{p_{ij}^2\in R_v^2}\lambda P_2(p_{ij}^2)-\rho_c(u,v) & if\ v=e_{2j}\\ 0 & otherwise\end{cases}\tag{5.11}$$

Given the arrival rates to each link, the blocking probability between edge node i and edge node j can be calculated as follows.

$$P_b(i,j)=Q_{p_{ij}^1}(0)\times Q_{p_{ij}^2}(0)\tag{5.12}$$

An iterative algorithm is developed to compute the blocking probability on each path. In the algorithm, a small positive number ε is set as a convergence criterion. The main steps of the algorithm can be described as follows.

1. For each pair of source and destination nodes, initialize $P_b^{'}(i,j)=0$, $i,j=1,2,\cdots,N$. For all links, initialize $\rho_l(u,v)$, $\rho_c(u,v)$, and $\rho_e(u,v)$ arbitrarily, $u,v\in E$;
2. Calculate $Q_p(n)$ for each path between each pair of source and destination nodes using equations (5.5) and (5.6);
3. Calculate the blocking probability $P_b(i,j)$ for each pair of source and destination nodes using equation (5.12). If $\max\left|P_b(i,j)-P_b^{'}(i,j)\right|<\varepsilon$, terminate the computation. Otherwise, let $P_b^{'}(i,j)=P_b(i,j)$ and go to the next step.

4. Calculate $\rho_l(u,v)$, $\rho_c(u,v)$, and $\rho_e(u,v)$ for each link using equations (5.9), (5.10), and (5.11), and then go back to step 2).

5.2.2.3. Results

In this section, the accuracy of the proposed analytical model is verified by performing a simulation study and comparing the simulation results with the analytical results obtained using the model. Without loss of generality, the AAPN with 4 edge nodes, 2 core nodes and eight timeslots (F =8) in each frame is considered. In the simulation, the call arrivals follow Poisson distribution and the duration of the calls follows exponential distribution. The results are obtained with the mean holding time of calls equal to 100 frames. The bandwidth allocation algorithm is the same as described in Section 1 of this chapter.

Figure 5.2 compares the analytical results and the simulations results in terms of call blocking probability in the network. In computing the analytical results, the convergence factor ε is set to 10^{-10} to ensure accuracy; multiple iterations were performed. In the simulation, each point was obtained with a simulation time of 10^6 timeslots. It is observed that the analytical results are close to the simulation results. It is noted that the simulation shows that the network has nearly no blocking when the traffic load is less than 0.5, which is confirmed by the analytical results, because the network has enough bandwidth to handle the traffic. However, the network blocking probability increases with the increases of the traffic load. When the traffic load is high, there exists a difference between the simulation and analytical results. This is because of the signaling delay, i.e. the state information sent from the core nodes becomes out-of-date when it arrives at the edge nodes. (Note that the analytical model assumes that there is no signaling delay between core and edge nodes).

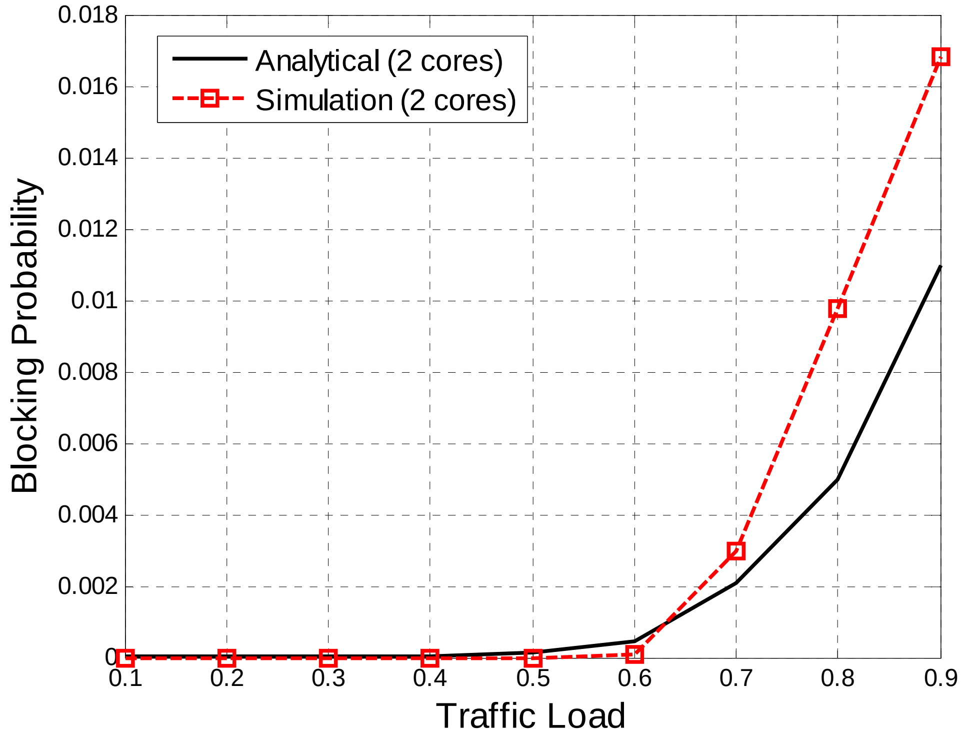


Figure 5.2: Blocking probability versus traffic load

5.2.3. Hybrid Routing Strategy

The least-congested-path routing strategy can effectively balance the traffic load on each link and thus decrease the blocking probability of the network. For a particular call, however, it may select a longer path instead of an available shorter path, which would increase the end-to-end delay of the connection. Actually, load balancing is unnecessary under low traffic load because there is no congestion in the network in this context (see Figure 5.2). The source node may therefore select the shortest available path for each connection request, which may improve the delay performance of the network while not affecting the blocking probability. Based on this argument, a *hybrid routing strategy* is proposed that takes into account both load balancing and end-to-end delay in path selection.

The strategy selects a path according to either the end-to-end distance or the congestion of a path, depending on the traffic load ρ. More precisely, the shortest available path is selected for the connection request when the traffic load is less than a certain threshold, i.e. $\rho \le \rho_{threshold}$; while the least congested path is selected when otherwise. The selection of the threshold value $\rho_{threshold}$ is discussed in the following.

For the AAPN with M core nodes, an observation, when applying the least-congested-path routing strategy, is that there is nearly no blocking if the traffic load is less than 0.5. When applying the hybrid routing strategy, the source edge node selects the core node that resides in the shortest path if its traffic load $\rho \le \rho_{threshold}$. In this context, the traffic load of that path may increase to as maximum as $M \cdot \rho_{threshold}$ (the traffic that, if the least congested path routing strategy is applied, should be shared by M paths is aggregated to the shortest path). Now we want to maintain the network blocking probability the same as when the least-congested-path routing strategy is applied. To achieve this goal, the traffic load of the shortest path can not exceed 0.5[1]. Then we have

$$M \cdot \rho_{threshold} = 0.5 \tag{5.13}$$

Consequently, the traffic load threshold $\rho_{threshold}$ is

$$\rho_{threshold} = 0.5 / M \tag{5.14}$$

5.3. Performance Evaluation

In this section, we evaluate the performance of the routing strategies discussed in the previous section through simulation. We consider a network with eight edge nodes (N=8)

[1] Note that a small increase of the traffic load offered to the network may cause a dramatic increase of the load in the shortest path because the traffic distributed to different cores aggregates to the same link. Consequently, when traffic load is greater than 0.5, a big increase of the blocking probability may occur if the shortest path is still applied.

and two core nodes (*M*=2). The node layout and link distances (in kilometers) are given in Table 5.1. The call arrival process is Poisson and the duration of each call is exponentially distributed.

Table 5.1 Node layout and link distance

Cores Distance Edges	Toronto	Montreal
Quebec	792	248
Montreal	564	0
Ottawa	396	195
Toronto	0	546
Waterloo	105	651
London	186	731
Hamilton	68	610
Windsor	366	910

Figure 5.3, Figure 5.4 and Figure 5.5 show the average blocking probability, end-to-end delay, and set-up time, respectively, with different routing strategies. The results are obtained with a frame size equal to 100 timeslots and the mean holding time of each call equal to 100 frames.

In Figure 5.3, one can observe that all the strategies achieve good blocking performance when traffic load is low. This means that the hybrid routing strategy can achieve zero blocking probability even when the shortest-path routing is used under low traffic load. The least-congested-path routing strategy and the hybrid routing strategy have the same blocking performance. The random routing strategy shows the worst performance because it does not take into account current link state information. The random-with-retrying strategy shows the best performance because it uses retrying. Note that retrying can also be used with the other strategies in order to further reduce the blocking probability.

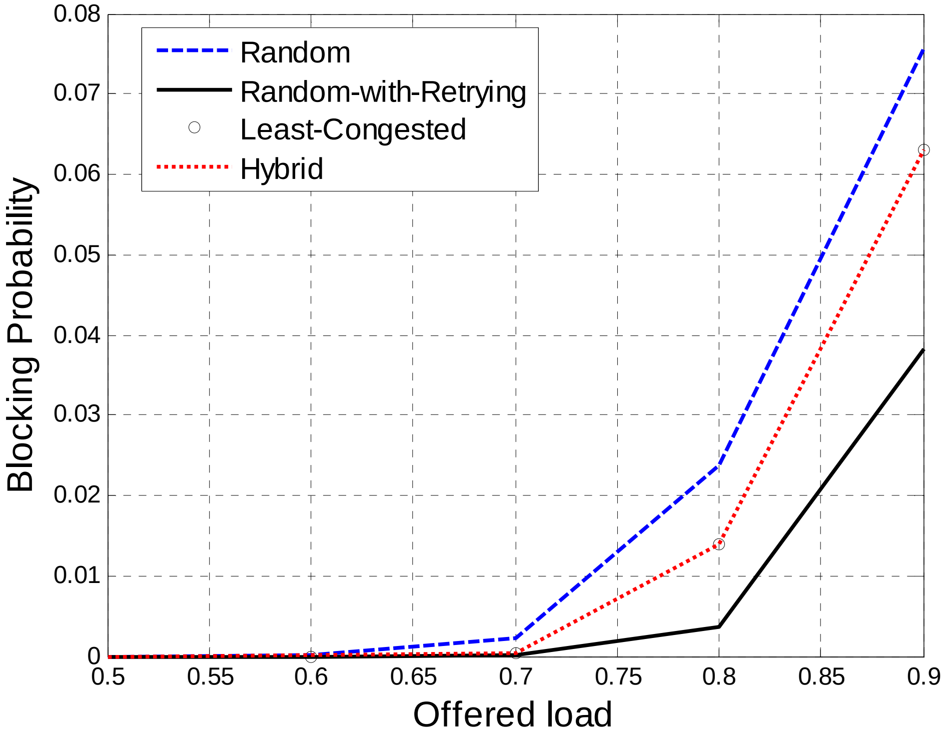


Figure 5.3: Blocking probability versus traffic load.

In Figure 5.4, it is observed that the hybrid routing strategy achieves better end-to-end delay performance than the other strategies when traffic load is lower than 0.25[1] because the shortest paths are selected instead of the least-congested paths. The improvement depends on the propagation delay of the shortest paths.

[1] Note that two core nodes (M=2) are used in this simulation. Therefore, the value of $\rho_{threshold}$ is equal to 0.25.

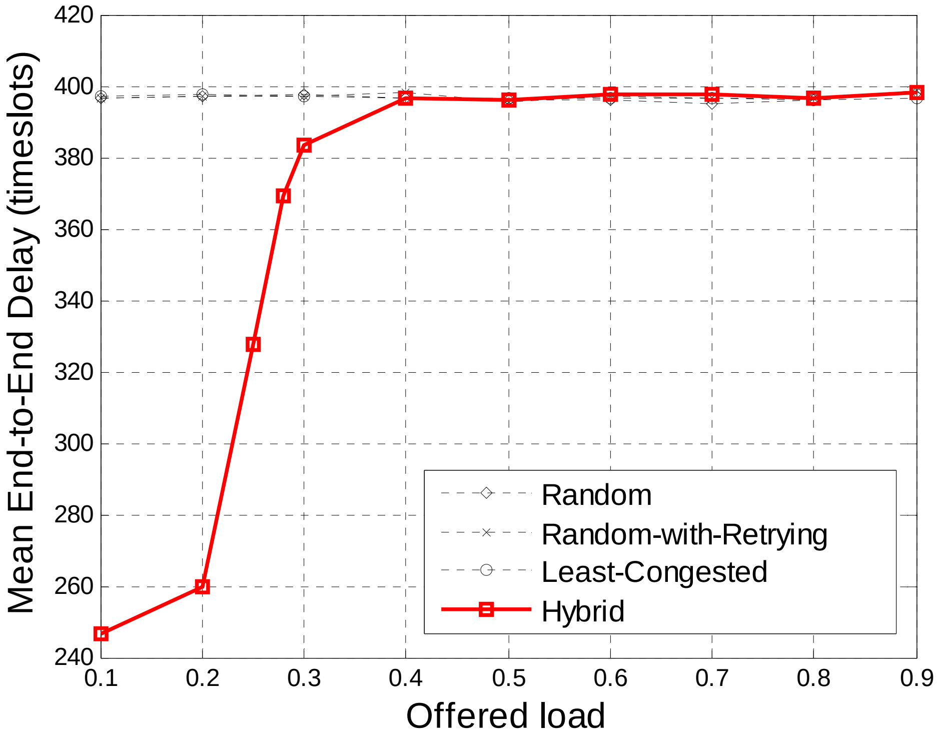


Figure 5.4: Mean end-to-end delay versus traffic load.

In Figure 5.5, it is observed that the hybrid routing strategy achieves better set-up delay performance than the other strategies when traffic load is lower than 0.25. This is because the shortest paths are selected and it takes the signaling protocol[1] less time to establish a connection. While the random-with-retrying strategy improves the blocking performance when traffic load is larger than 0.7, it needs a larger connection set-up time because of the use of retrying.

[1] Note that the in-band signalling is assumed.

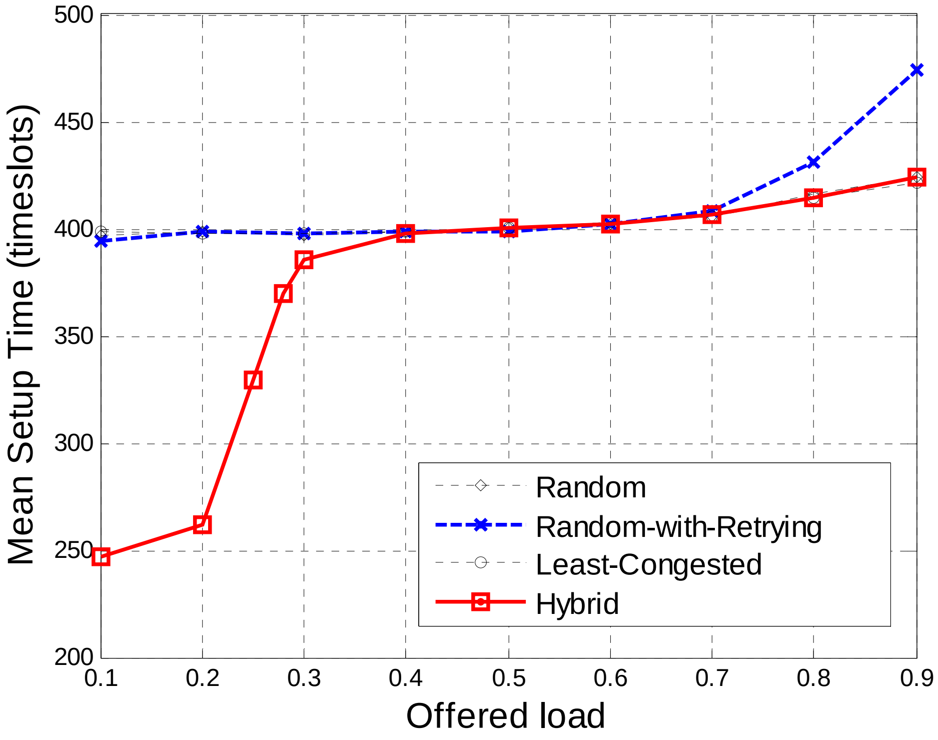


Figure 5.5: Set-up time versus traffic load.

5.4. Summary

In this Chapter, routing strategies are investigated for load balancing in all-optical overlaid-star TDM networks. The random routing strategy and the hybrid strategy are first presented, and the hybrid routing strategy is then proposed to improve the delay performance under low traffic load. The simulation results show that the proposed hybrid routing strategy can significantly improve the delay performance under low traffic load while maintaining the same blocking probability as that of the other routing strategies. The introduction of retrying leads to the best blocking performance but at the cost of a larger connection set-up time.

6. Conclusions

6.1. Summary

This thesis focuses on the bandwidth allocation for the agile all-photonic networks (AAPN). The AAPN can be deployed as a long-haul network or a metropolitan network. In this context, two problems related to the bandwidth allocation must to be solved:

i. What kind of bandwidth allocation scheme can be employed at the core node? – It should be both efficient and simple.

ii. What routing strategy should be employed in the context of the AAPN?

Both theoretical analysis and simulations confirm that the combination of the alternating projections method with the quick Birkhoff-von Neumann decomposition method provides a simple and efficient bandwidth allocation scheme for the AAPN. The routing method proposed in this work provides a trade-off between the delay and the blocking probability, i.e. the method improves the delay performance of the network dramatically under the low load while not affecting the blocking performance at high load.

6.2. Thesis Contributions

6.2.1. Bandwidth Allocation Scheme

The main contribution of this thesis work lies in the frame-based bandwidth allocation scheme for the AAPN. As outlined in the summary above, the combination of the alternating projection method[1] and the quick Birkhoff-von Neumann decomposition method[2] provides a simple and efficient frame-based bandwidth allocation scheme for the AAPN. For the alternating projections method (see footnote 1), I compared the similarity of the method with others (see Section 3.3.4, 3.3.5 and 3.3.7) and showed through simulation that the projections method provides a better delay performance.

The proposed bandwidth allocation scheme can be extended to the context of the AAPN with concentrators. In this case, a two-step allocation scheme is proposed in this thesis to overcome the scalability issue. This scheme can be used in the AAPN with either passive or active concentrators.

Another extension of the proposed bandwidth allocation scheme is in the context of several levels of QoS. This new scheme, called the QoS guaranteed frame-based bandwidth allocation scheme and described in Section 4.3.2, can be adopted to support guaranteed QoS.

I also proposed two analytical models[3], called first-fit model and first-fit with random model, to analyze the delay performance for the AAPN and made the simulation to verify the model. It is shown that, if a bandwidth allocation algorithm keeps allocating free bandwidth (i.e. the bandwidth that is not allocated to any requests), the allocation algorithm may achieve a good delay performance especially in long-haul networks.

[1] Cheng Peng, Sofia A. Paredes, Trevor J. Hall and Gregor v. Bochmann, “Constructing service matrices for agile all-photonic network cores”, in proceedings of 11th IEEE Symposium on Computers and Communications (ISCC'06), Pula-Cagliari, Sardinia, Italy, 26-29 June, 2006, pp. 967-973. (the Best Student Paper Award)

[2] Cheng Peng, Gregor v. Bochmann and Trevor J. Hall, “Quick Birkhoff-von Neumann decomposition algorithms for agile all-photonic network cores”, in proceedings of 2006 IEEE International Conference on Communications (ICC 2006), Istanbul, Turkey, 11-15 June, 2006, pp. 2593-2598

[3] Cheng Peng, Peng He, Gregor v. Bochmann and Trevor J. Hall, “Delay performance analysis for an agile all-photonic star network”, 5th International IFIP-TC6 Networking Conference, Coimbra, Portugal, May 15-19, 2006, Proceedings. Lecture Notes in Computer Science 3976 Springer 2006, pp. 368-378

6.2.2. Routing

In the area of routing for AAPNs, an analytical model[1] for the AAPN is proposed (see footnote 2) to investigate the blocking probability when the well-known least-congested path routing is employed. In this work, I investigated the analytical model and verified it through simulation.

Furthermore, a routing model[2] is proposed (see footnote 3) by which the source edge node can select the shortest available path under low traffic load and balance the traffic when the traffic load is high. The method improves the delay performance of the network dramatically at low traffic load while not affecting the blocking performance at high load. In this work, I proposed the routing model and studied the performance by simulation.

6.3. Contributions related to the AAPN Project

6.3.1. Fault Restoration

In the area of fault restoration in AAPNs, a fault detection and restoration method[3] has been proposed and the performance was simulated. In this work, Dr. Jun Zheng and I

[1] Jun Zheng, Cheng Peng, and Gregor v. Bochmann, "Blocking model for all-optical overlaid-star TDM networks", in proceedings of 2006 IEEE GLOBECOM conference, San Francisco, USA, pp. 1-6

[2] Jun Zheng, Cheng Peng, Gregor v. Bochmann and Trevor J. Hall, "Load balancing in all-optical overlaid-star TDM networks", in proceedings of 2006 IEEE Sarnoff Symposium, Princeton, NJ, USA, 27-28 March, 2006, .

[3] Jun Zheng, Cheng Peng, and Gregor v. Bochmann, "An Effective Fault Detection and Localization Scheme for All-Optical Overlaid-Star TDM Networks", in proceeding of 2006 ChinaCom, Beijing, China

proposed an effective fault localization technique to facilitate the provisioning of a restoration scheme in the event of a network failure. Furthermore, Dr. Jun Zheng proposed a fault advertisement protocol and I analyzed the service recovery time with the proposed fault detection and localization technique.

6.3.2. AAPN Prototype Implementation

The AAPN prototype will be implemented in software and hardware. The control protocols has been designed between edge nodes and the core switch controller, allowing the integration of various bandwidth allocation algorithms and IP-MPLS transmission of traffic coming from certain applications and artificial traffic sources.

Mr. Yong Deng, Dr. Yiming Zhang and I are in charge of the software development of the AAPN prototype. Specifically, I implemented the frame-based bandwidth allocation scheme (see Chapter 3 and Chapter 4). As compliant with the technical specification of AAPN, the execution time of the scheme should be less than 1 millisecond for 64×64 switches (the frame size is 100 timeslots and the duration of each timeslot is 10 microseconds). I tested the execution time of the scheme in Linux with Intel's Pentium(R) 4 CPU 3.20GHz using a traffic demand trace generated through a bursty traffic model (see Section 4.2 of Chapter 2) with parameters $\alpha_{ON} = 1.2, \beta_{ON} = 1, \alpha_{OFF} = 1.4$ and $\rho = 0.7$. The trace length is 1,000,000 timeslots. In the case of switches with 64 ports, the implementation finishes execution in around 0.777 milliseconds; in the case of a switch with 32 ports, the implementation finishes execution in 0.338 milliseconds on average[1].

[1] The code was optimized according to the Sun application tuning seminar, "application performance optimization on Sun systems", that was held by HPCVL (high-performance computing virtual laboratory) http://www.hpcvl.org/

6.4. Future Work

The above work in routing and fault restoration is restricted to the framework of AAPN. In the future, the work can be extended to a context where AAPNs and legacy networks are interconnected. In this context, the network consists of several domains (e.g. the AAPN itself may be viewed as a domain). It is therefore worth studying the global optimal routing when AAPNs are deployed to the backbone domain, which includes dynamic routing algorithms to create paths that span multiple domains and efficient mechanism of routing information exchange within and between domains.

In the fault restoration aspect, it is worth studying and designing novel methods/mechanisms and algorithms for MPLS flows over AAPNs in multi-layer (e.g. IP/MPLS layer and optical layer), multi-service (e.g. services with differential protection and restoration requirements) and multi-domain network scenario. In general, these techniques and their combinations should be usable in different parts/segments of the networks to protect an inter-domain path. In addition, to improve the efficiency of the resource utilization, the capability to share bandwidth among inter-domain backup paths protecting independent facilities must be considered [He2006].

Appendix: Internet Traffic and Modeling

A. Internet Traffic Characteristics

Network devices put packets on an Internet link and multiplex the packets from different active connections. Empirical and theoretical studies of Internet traffic conclude that an increasing number of simultaneous active connections cause a dramatic change in the statistical properties of packet traffic on an Internet link [Cao2002]. Starting at low connection loads on an uncongested link, packet arrivals are long-range dependent, creating burstiness[1]. As the connection load increases, the packet arrivals tend to Poisson and the burstiness begins to reduce due to the multiplexing gain. When the load is heavy, the waiting time for service will increase exponentially as the load on a queue approaches 100% [Kingman1962].

These results have important implications for the AAPN. For optical metropolitan networks, an edge node connects with a small number of user hosts. Hence connection loads may not be very large. The traffic characteristics tend to be bursty. For optical long haul networks, an optical link (e.g. wavelength) may carry large numbers of connections. The traffic characteristics tend to be smooth.

[1] It means large variation in the traffic bit rate.

B. Bursty Traffic Models

The main findings that traffic exhibits self-similar or fractal behavior[1] was first highlighted in [Leland1993]. Self-similar traffic exhibits long-range dependence (i.e., hyperbolic decay of autocorrelations with increasing time separation). This is in contrast to classical traffic models, such as Poisson, which exhibit short-range dependence (i.e., exponentially decaying autocorrelations).

There are two common families of self-similar traffic generators: fractional Gaussian noise and fractional ARIMA processes[2]; in the teletraffic literature the former is more prevalent [Reeve2003]. Fractional Gaussian noise, as shown in [Wang2003], is produced when a number of on-off sources are multiplexed together. Each source is either sending traffic at a constant rate (in the **on** state) or sending no traffic at all (in the **off** state). The distribution of the time spent in each state is heavy tailed. The Pareto distribution, with finite mean and infinite variance, may be used to model the life time in each state. The aggregation of these sources yields a process with long-range dependence due to the heavy-tail nature of the Pareto distribution, and hence a bursty traffic trace [Leland1994] [Paxson1995a].

The probability density function for the Pareto distribution is [Ash1993]:

$$f(x) = \alpha\beta^{\alpha} x^{-\alpha-1}, x \geq \beta$$

and the cumulative distribution function is:

$$F(x) = 1 - (\beta / x)^{\alpha}$$

[1] These two terms mean that the traffic is bursty at all time scales and there is no natural length of a burst; in effect at any instant of time it is impossible to predict if a burst will occur, and if it does, how long that burst will last.

[2] ARIMA is an acronym for **A**uto**R**egressive, **I**ntegrated, **M**oving **A**verage.

with mean

$$E(x) = \frac{\alpha\beta}{\alpha - 1}$$

Where β is the minimum value of x, $\beta > 0$; and α is the tail index, $\alpha > 0$. When $\alpha \le 2$, the variance of the distribution is infinite. When $\alpha \le 1$, the mean value becomes infinite as well. For self-similar traffic, the distribution is asked to have infinite variance, but finite mean. Therefore, α should be $1 < \alpha < 2$. As shown in Figure A-1, longer tails in the distribution are obtained with $\alpha \to 1$.

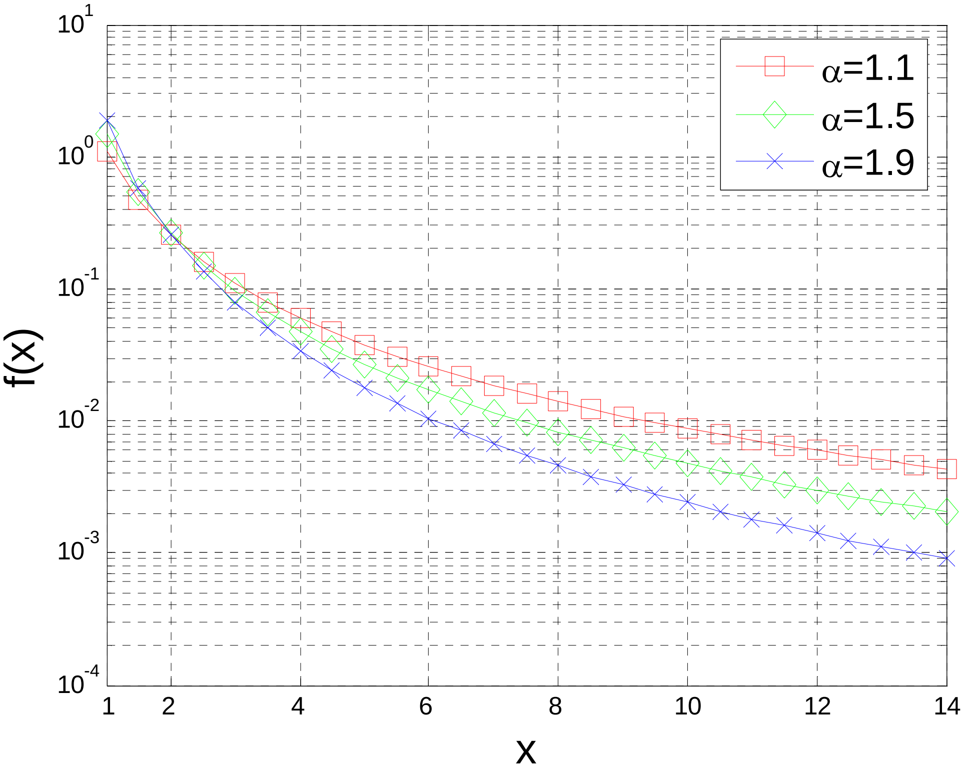


Figure A-1: Probability density functions for Pareto distribution with different α ($\beta = 1$)

Let the cumulative distribution function $F(x)$ be denoted as u, i.e. $u = F(x)$, $(0 \le u < 1)$. The inverse function of $F(x)$ is

$$x = \frac{\beta}{(1-u)^{1/\alpha}}$$

If u is a random variable uniformly distributed between 0 and 1, then x is a Pareto distributed random variable. The formula implies that a uniform distributed random variable u that can be generated by computers can be mapped to a Pareto distributed random variable x.

The Pareto distributed random variable x may be used to describe both the sending time in the **on** state, x_{ON}, and the silent time in the **off** state, x_{OFF} (In this case, both the sending time and the silent time are Pareto distribution). For the AAPN, both are counted by the number of timeslots. A method to generate bursty traffic is to move between the **on** state and the **off** state alternately with the given parameters α_{ON}, β_{ON}, α_{OFF}, ρ (average load)[1].

Given a uniform distributed random variable u, the sending time can be determined by

$$x_{ON} = \frac{\beta_{ON}}{(1-u)^{1/\alpha_{ON}}}$$

Now, in order to determine the silent time x_{OFF}, we need to know the value of the parameter β_{OFF}. This parameter can be determined by the following arguments.

Define

$$E_{OFF} = E(x_{OFF})$$

$$E_{ON} = E(x_{ON})$$

[1] Here the parameter β_{OFF} is missing so the silent time x_{OFF} cannot be determined. Therefore, we need to calculate the parameter by these givens.

Given a specified average load ρ, the average silent time E_{OFF} can be calculated by

$$\rho = \frac{E_{ON}}{E_{ON} + E_{OFF}}$$
$$\Rightarrow$$
$$E_{OFF} = (\frac{1}{\rho} - 1)E_{ON}$$

where the average traffic sending time E_{ON} can be determined by

$$E_{ON} = \frac{\alpha_{ON}\beta_{ON}}{\alpha_{ON} - 1}$$

Finally, the parameter β_{OFF} can be determined

$$E_{OFF} = \frac{\alpha_{OFF}\beta_{OFF}}{\alpha_{OFF} - 1}$$
$$\Rightarrow$$
$$\beta_{OFF} = \frac{\alpha_{OFF} - 1}{\alpha_{OFF}} E_{OFF}$$

Now we can calculate the silent time given the uniform distributed random variable u

$$x_{OFF} = \frac{\beta_{OFF}}{(1-u)^{1/\alpha_{OFF}}}$$

C. Poisson Traffic Model

The Poisson traffic model provides a simple and trivially tractable model for reasoning about telecommunications traffic. However, it underestimates the burstiness of aggregated network traffic [Paxson1995b]. The Poisson traffic model uses the Poisson distribution to model the number of events (e.g. packet arrivals, telephone calls, etc.) occurring within a given time interval. The interval between consecutive events follows an exponential distribution.

The probability density function for the exponential distribution is:

$$f(x) = \lambda e^{-\lambda x}$$

where λ is the rate parameter[1] and the cumulative distribution function is:

$$F(x) = 1 - e^{-\lambda x}$$

with the mean

$$E(x) = \frac{1}{\lambda}$$

Let the cumulative distribution function $F(x)$ be denoted as u, i.e. $u = F(x)$, $(0 \le u < 1)$. The inverse function of $F(x)$ is

$$x = -\frac{1}{\lambda}\log(1-u)$$

If u is a random variable uniformly distributed between 0 and 1, then x is a exponential distributed random variable that may be used to describe the interval between consecutive arrivals. The relationship between the arrival time and the interval is shown in Figure A-2.

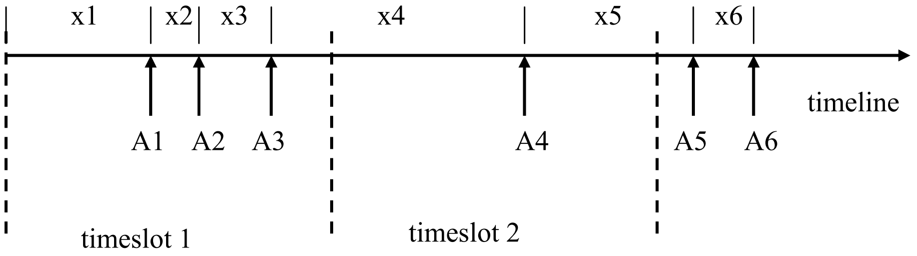


Figure A-2: Relationship between the arrival time and the interval time

[1] It is usually regarded as arrival rate of the calls in telephony networks.

The $A_i, i = 1,2,3,...$ denote the packet arrivals and the $x_i, i = 1,2,3,...$ denotes the intervals between consecutive arrivals. Thus the arrival time T_i of A_i may be calculated by

$$T_i = \sum_{j=1}^{i} x_j$$

For each packet arriving at the time T_i, the length of the packet may be calculated through the Poisson distribution as well. Note that since packets have duration and the inter-arrival time is arbitrarily short, Poisson distribution is not physical on short time scales as it allows packets to overlap.

References

[Agusti2005] A. Agusti-Torra, G. v. Bochmann and C. Cervello-Pastor, “Retransmission schemes for optical burst switching over star networks”, in Proceedings of. the 2nd IFIP Intern. Conf. on Wireless and Optical Communications Networks (WOCN), March 2005, Dubai, United Arab Emirates, pp. 126-130.

[**Anderson1993**] T. E. Anderson, S. S. Owicki, J. B. Saxe, C. P. Thacker, “High-speed switch scheduling for local-area networks”. ACM Transactions on Computer Systems, Vol. 11, No. 4, November 1993, pp. 319-352.

[Ash1993] Carole Ash, “The probability tutoring book: an intuitive course for engineers and scientists (and everyone else!)”. IEEE Press, New York, USA, 1993.

[Barry1996] R. A. Barry and P. A. Humlet, “Models of blocking probability in all-optical networks with and without wavelength changers,” IEEE Journal on Selected Areas in Communications, Vol. 14, No. 5, Jun. 1996, pp. 858-867.

[Bauschke2002] H.H. Bauschke, P.L. Combettes, D.R. Lake, “Phase retrieval, error reduction algorithm, and Fienup variants: a view from convex optimization”, Journal of the Optical Society of America, Vol. 19, No. 7, July 2002, pp.1334-1345.

[Bertsekas1992] D. Bertsekas and R. Gallager, “Data Networks”, Prentice Hall, 1992.

[Bianco2000] A. Bianco, E. Leonardi, M. Mellia, and F. Neri, "Network Controller Design for SONATA - A Large-Scale All-Optical Passive Network," IEEE Journal on Selected Areas in Communications, Vol. 18, No. 10, Oct. 2000, pp. 2017-2028.

[**Birkhoff1946**] G. Birkhoff, “Tres observaciones sobre el algebra lineal”, Univ. Nac. Tucuman, Rev. Ser. A 5, 1946, pp. 147-151.

[Birman1996] A. Birman, “Computing approximate blocking probabilities for a class of all-optical networks,” IEEE Journal on Selected Areas in Communications, Vol. 14, No. 5, June 1996, pp. 852-857.

[Blouin2002] F.J. Blouin, A.W. Lee, A.J.M. Lee, and M. Beshai, “Comparison of two optical-core networks,” Journal of Optical Networking, Vol. 1, No. 1, Jan. 2002, pp. 56-65.

[**Boch2004**] G.v. Bochmann, M.J. Coates, T. Hall, L. Mason, R. Vickers and O. Yang, “The Agile All-Photonic Network: An architectural outline”, in Proceedings of the. Queen's University Biennial Symposium on Communications, 2004, pp. 217-218.

[Boyle1986] 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.

[Brunato2003] M. Brunato, R. Battiti, and E. Salvadori, “Dynamic load balancing in WDM networks,” Optical Networks Magazine, Vol. 4, No. 5, 2003, pp. 7-20.

[Cao2002] J. Cao, W. Cleveland, D. Lin, and D. Sun, "Internet traffic tends toward Poisson and independent as the load increases," in Nonlinear Estimation and Classification, D. Denison, M. Hansen, C. Holmes, B. Mallick, and B. Yu, Eds. New York, NY: Springer Verlag, Dec. 2002, pp. 83-109.

[Chang2000] C.S. Chang, W.J. Chen and H.Y. Huang, “Birkhoff-von neumann input buffered crossbar switches”, Proceedings of IEEE INFOCOM, 2000, pp. 1614-1623.

[Chang2001] Cheng-Shang Chang, Wen-Jyh Chen, Hsiang-Yi Huang, "Birkhoff-von Neumann input-buffered crossbar switches for guaranteed-rate services", IEEE Transactions on Communications, Vol. 49, No. 7, July 2001, pp. 1145-1147.

[**Chao2000**] H. J. Chao, “Satrun: a terabit packet switch using dual round-robin”, IEEE Communication Magazine, Vol. 38, No. 12, December 2000, pp. 78-79.

[**Chen1994**] M. Chen and N. D. Georganas, “A fast algorithm for multi-channel/port traffic scheduling,” in Proceedings of. IEEE Supercom/ICC‘94, pp. 96–100.

[Chen2003] Y. Chen, H. Yu, D. Xu, C. Qiao. “Performance analysis of optical burst switched node with deflection routing.” In Proceedings of IEEE ICC, vol.2, 2003, pp.1355-1359

[**Chuang1999**] Shang-Tse Chuang, Ashish Goel, Nick McKeown, “Matching output queueing with a combined input/output-queued switch”. IEEE Journal on Selected Areas in Communications, Vol. 17, No. 6, June 1999, pp. 1030-1039.

[Cole1982] R. Cole and J. Hopcroft, “On edge coloring bipartite graphs”, SIAM Journal of Computing,Vol. 11, 1982, pp.540-546.

[**Cole2001**] R. Cole, K. Ost and S. Schirra, “Edge coloring bipartite multigraphs in O(E logD) time”, Combinatorica, Vol. 21, No. 1, 2001, pp. 5-12.

[**Cormen2001**] T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, “Introduction to algorithms”, the MIT Press, 2001, pp.664-669

[**Dai2000**] J.G. Dai and B. Prabhakar, “The throughput of data switches with and without speedup”, In Proceedings of the IEEE INFOCOM, Tel Aviv, Israel, 2000, pp.556-564.

[**Gale1962**] D. Gale, L.S. Shapley, “College admission and the stability of marriage”, American Mathematical Monthly, Vol. 69, 1962, pp.9-15.

[Ghani2000] N. Ghani, S. Dixit, and T. Wang, “On IP-over-WDM integration,” IEEE Communications Magazine, vol. 38, no. 3, Mar. 2000, pp. 72-84.

[**Giaccone2002**] P.Giaccone, B.Prabhakar, D. Shah, "Towards simple, high-performance schedulers for high-aggregate bandwidth switches", in Proceedings of IEEE INFOCOM '02, New York, NY, Jun.2002, pp.1160-1169.

[Girard1990] A. Girard, “Routing and dimensioning in circuit-switched networks”, Addison-Wesley, 1990.

[Grover2003] W. D. Grover, “Mesh-based survivable transport networks: options and strategies for optical, MPLS, SONET and ATM networking”, Prentice Hall PTR, Upper Saddle River, New Jersey, Aug. 2003.

[Hall2005] Trevor J. Hall, Sofia A. Paredes, Gregor v. Bochmann. "An agile all-photonic network", in Proceedings of the International Conference on Optical Communications and Networks, ICOCN 2005; Bangkok, Thailand, 14-16 December 2005, pp. 365-368.

[He2006] P. He and G. v. Bochmann, “Routing of MPLS flows over an agile all-photonic star network”, in Proceedings of IASTED International Conference on Communication Systems and Applications (CSA 2006), July, 2006, pp. 138-144.

[Hoffman1953] A.J. Hoffman and H.W. Wielandt, “The variation of the spectrum of a normal matrix”, Duke Math. Journal, Vol. 20, 1953, pp.37-39.

[**Hluchyj1988**] M. Hluchyj and M. Karol, "Queueing in high performance switching", IEEE Journal on Selected Areas in Communications, Vol. 6, No. 9, December 1988, pp. 1587-1597.

[**Hopcroft1973**] J. E. Hopcroft, R.M. Carp, “An $n^{5/2}$ algorithm for maximum matching in bipartite graphs”, SIAM Journal on Computing, 1973, pp. 225-231.

[Hsu2001] Ching-Fang Hsu, Te-Lung Liu, and Nen-Fu Huang, “Performance of adaptive routing strategies for wavelength-routed networks,” in Proceedings of 2001 IEEE International Conference on Performance, Computing, and Communications, Apr. 2001, pp. 163-170.

[Johnson1960] D.M. Johnson, A.L. Dulmage, and N.S. Mendelsohn, “On an algorithm of G. Birkhoff concerning doubly stochastic matrices”, Canad. Math. Bull., Vol. 3, 1960, pp.237-242.

[Kam1999] A.C. Kam, K.Y. Siu, “Linear-complexity algorithms for QOS support in input-queued switches with no speedup”, IEEE Journal on Selected Areas in Communications, Vol. 17, No. 6, 1999,pp.1040-1056.

[**Kamiyama2005**] N. Kamiyama, "A Large-Scale AWG-Based Single-Hop WDM Network Using Couplers With Collision Avoidance," IEEE/OSA JLT, Vol. 23, No. 7, July 2005, pp. 2194-2205.

[**Karol1987**] M. J. Karol and M. G. Hluchyj, Samuel P. Morgan, “Input versus output queueing on a space-division packet switch”. IEEE Transactions on Communications, Vol. 35, No. 12, December 1987, pp. 1347-1356.

[**Karol1988**] M. Karol, M. Hluchyj, and S. Morgan, “Input versus output queuing on a space division switch”, IEEE Journal on Selected Areas in Communications, Vol. 35, 1998, pp.1347-1356.

[Karp1990] R. M. Karp, U. V. Vazirani, V. V. Vazirani, “An optimal algorithm for on-line bipartite matching”, in Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing, April 1990, pp. 352-358.

[**Keslassy2003**] I. Keslassy, M. Kodialam, T.V. Lakshman and D. Stiliadis, "On guaranteed smooth scheduling for input-queued switches", in Proceedings of IEEE INFOCOM, 2003, pp. 1384-1394.

[Kingman1962] J. F. C. Kingman, “On queues in heavy traffic”, Journal of the Royal Statistical Society, Series B, No. 24, 1962, pp. 383-392.

[Knopp1967] P Knopp and R Sinkhorn, “Concerning nonnegative matrices and doubly stochastic matrices”, Pacific Journal of. Mathematics, Vol. 21, No. 2, 1967, pp. 343-348.

[Kovacevic1996] M. Kovacevic and A. Acampora, “Benefits of wavelength translation in all-optical clear-channel networks,” IEEE Journal on Selected Areas in Communications, Vol. 14, No. 5, Jun. 1996, pp. 868-880.

[**Krishna1999**] P. Krishna, N.S. Patel, A. Charny and R.J. Simcoe, “On the speedup required for work-conserving crossbar switches”, IEEE Journal on Selected Areas in Communications, Vol.17, No. 6, 1999, pp.1057-1066

[**Kumar2004**] Neha Kumar, Rong Pan, Devavrat Shah, “Fair Scheduling in Input-Queued Switches under Inadmissible Traffic”, in Proceedings of Globecom, Dallas, 2004, pp. 1713-1717.

[Leland1993] Will E. Leland, Murad S. Taqq, Walter Willinger, and Daniel V. Wilson., “On the self-similar nature of Ethernet traffic”, In Proceedings of ACM SIGCOMM’93, San Francisco, USA, 1993, pp183-193.

[Leland1994] Will E. Leland, Murad S. Taqqu, Walter Willinger, Daniel V. Wilson, “On the self-similar nature of Ethernet traffic (Extended Version)”, IEEE/ACM Transactions on Networking, Vol. 2, No. 1, February 1994, pp. 1-15.

[**Leonardi2001**] E. Leonardi, M. Mellia, F. Neri, M.A. Marsan, “On the stability of input-queued switches with speedup”, IEEE/ACM Transactions on Networking, Vol. 9, No. 1, 2001, pp.104-118.

[**Leonardi2001A**] E. Leonardi, M. Mellia, F. Neri, M.A. Marsan, “Bounds on average delays and queue size averages and variances in input-queued cell-based switches”, In Proceedings of the IEEE INFOCOM’2001, Anchorage, 2001, pp.1095-1103.

[Li1999] L. Li and A. K. Somani, “Dynamic wavelength routing using congestion and neighborhood information,” IEEE/ACM Transactions on Networking, Vol. 7, No. 5, Oct. 1999, pp. 779-786.

[**Li2001**] J. Li, N. Ansari, “Enhanced Birkhoff-von Neumann decomposition algorithm for input queued switches”, in IEE Proceedings of Communications, Vol. 148, No. 6, 2001, pp. 339-342.

[**Li2003**] Y. Li, S. Panwar and H. J. Chao, “Frame-based matching algorithms for optical switches”, in Proceedings of Workshop on High Performance Switching and Routing, Torino, Italy, Jun., 2003, pp.97-102.

[Liu2005a]. X. Liu, A. Vinokurov, L.G. Mason, "Performance comparison of OTDM and OBS scheduling for agile all-photonic network", in Proceedings of IFIP Metropolitan Are Network conference, April 2005, Vietnam

[Liu2005b]. X. Liu, N. Saberi, M.J. Coates and L.G. Mason, “A comparison between time slot scheduling approaches for all-photonic networks”, in Proceedings of. IEEE International Conference on Information, Communications and Signal Processing, Bangkok, Thailand, Dec. 2005, pp. 1197-1201.

[Maach2002] A. Maach, G. v. Bochmann. “Segmented burst switching: enhancement of optical burst switching to decrease loss rate and support quality of service” in Proceedings of the 6th IFIP Working Conference on Optical Network design and modeling, Torino, Italy, February 2002, pp. 69-84.

[**Maier2002**] M. Maier, M. Scheutzow, M. Reisslein, and A. Wolisz, "Wavelength reuse for efficient transport of variable-size packets in a metro WDM network," in Proceedings of IEEE INFOCOM’2002, vol. 3, June 2002, pp. 1432-1441.

[Mas1999] C. Mas, P. Thiran, and J. L. Boudec, “Fault localization at the WDM layer,” Kluwer Photonic Network Communications, Vol. 1, No. 3, 1999, pp. 235-255.

[Mas2000] C. Mas and P. Thiran, “An efficient algorithm for locating soft and hard failures in WDM networks,” IEEE Journal on Selected Areas in Communications, Vol. 18, No. 10, Oct. 2000, pp. 1900-1911.

[Mas2001] C. Mas and P. Thiran, “A review on fault location methods and their application to optical networks,” SPIE Optical Networks Magazine, Vol. 2, No. 4, Jul./Aug. 2001, pp. 73-87.

[**Mason2006**] L.G. Mason, A. Vinokurov, N. Zhao and D. Plant, "Topological design and dimensioning of agile all photonic networks”, Elsevier Computer Networks Journal, Vol 50, ,No. 2, 2006, pp.268-287

[**Mekkittikul1998**] A. Mekkittikul and N. McKeown, “A practical scheduling algorithm for achieving 100% throughput in input-queued switches,” in Proceedings of INFOCOM ‘98, San Francisco, CA, Vol. 2, pp. 792–799.

[**McKeown1993**] N. McKeown, P. Varaiya and J. Warland, “Scheduling cells in an input-queued switch”, IEEE Electron. Letter, December 1993, pp.2174-2175.

[**McKeown1995**] N. McKeown, “Scheduling algorithms for input-queued switches”, Ph.D. thesis, University of California at Berkeley, 1995

[**McKeown1996**] N. McKeown, V. Anantharam, and J. Walrand, “Achieving 100% throughput in an input-queued switch,” in Proc. IEEE INFOCOM, San Francisco, CA, 1996, pp. 296–302.

[**McKeown1999**] N. McKeown, “The iSLIP Scheduling Algorithm for Input-Queued Switches”, IEEE/ACM Tran. On Networking, Vol. 7, No. 2, April, 1999, pp.188-201.

[**McKeown1999A**] N. Mckeown, A. Mekkittikui, V. Anantharam and J. Walrand, “Achieving 100% throughput in an input-queued switch”, IEEE Transactions on Communications, Vol. 47, No. 8, 1999, pp.1260-1267.

[**Mneimneh2003**] S. Mneimneh and K. Siu, “On achieving throughput in an input-queued switch”, IEEE/ACM Transactions on Networking, Vol. 11, No. 5, Oct., 2003, pp.858-867.

[Morris2000] Robert Morris and Dong Lin, “Variance of aggregated web traffic”, in Proceedings of INFOCOM’2000, 2000, pp 360-366.

[Narula2000] A. Narula-Tam and E. Modiano, “Dynamic load balancing in WDM packet networks with and without wavelength constraints,” IEEE Journal on Selected Areas in Communications, Vol. 18, No. 10, Oct. 2000, pp. 1972-1979.

[Neumann1953] J. von Neumann, “A certain zero-sum two-person game equivalent to the optimal assignment problem,” Contributions to the Theory of Games, Vol. 2, Princeton University Press, Princeton, New Jersey, 1953, pp. 5-12.

[**Nong1999**] G. Nong, J.K. Muppala and M. Hamdi, “Analysis of nonblocking ATM switches with multiple input queues”, IEEE/ACM Transactions on Networking, Vol. 7, No. 1, Feb. 1999, pp.60-74.

[**Oki2001**] E. Oki, R. Rojas-Cessa and H. J. Chao, “A pipeline-based approach for maximal-sized matching scheduling in input-buffered switches”, IEEE Communications Letters, Vol. 4, No. 6, Jun., 2001, pp.263-265.

[**Paredes2004**] S. A. Paredes-Zamorano, “Flexible bandwidth provision and scheduling in a packet switch with an optical core”, Ph.D. thesis, King’s College, University of London, London, UK 2004.

[Paredes2005] S.A. Paredes, T.J. Hall, "Flexible bandwidth provision and scheduling in a packet switch with an optical core". OSA Journal of Optical Networking, Vol. 4, No. 5, May 2005, pp. 260-270.

[Paxson1995a] Vern Paxson, "Fast approximation of self-similar network traffic", Technical Report, Lawrence Berkeley Laboratory and EECS Division; University of California, Berkeley. April 20, 1995.

[Paxson1995b] Vern Paxson and Sally Floyd, "Wide area traffic: the failure of Poisson modeling", IEEEACM Transactions on Networking, Vol. 3, No. 3, 1995, pp. 226-244.

[Peng2006a] Cheng Peng, Sofia A. Paredes, Trevor J. Hall and Gregor v. Bochmann, "Constructing Service Matrices for Agile All-Photonic Network Cores", in proceedings of 11th IEEE Symposium on Computers and Communications (ISCC'06), Pula-Cagliari, Sardinia, Italy, 26-29 June, 2006, pp. 967-973.

[Peng2006b] Cheng Peng, Gregor v. Bochmann and Trevor J. Hall, "Quick Birkhoff-von Neumann Decomposition Algorithms for Agile All-Photonic Network Cores", in proceedings of 2006 IEEE International Conference on Communications (ICC 2006), Istanbul, Turkey, 11-15 June, 2006, pp. 2593-2598.

[Reeve2003] David C. Reeve, "A New Blueprint for Network QoS", PhD thesis, University of Kent, Canterbury, UK, Augest 2003

[RFC2210] J. Wroclawski, "The use of RSVP with IETF integrated services", http://www.ietf.org/rfc/rfc2210.txt, Last accessed on April 10, 2007

[RFC2210] S. Shenker, C. Partridge and R. Guerin, "Specification of guaranteed quality of service", http://www.ietf.org/rfc/rfc2212.txt, Last accessed on April 10, 2007

[Saberi2006a] N. Saberi and M.J. Coates, "Fair matching algorithm: fixed-length frame scheduling in all-photonic networks", in Proceedings of. IASTED International Conference of Optical Communications Systems and Networks, Banff, AB, Canada, July 2006, pp. 213-218.

[Saberi2006b] N. Saberi and M.J. Coates, “Minimum rejection scheduling in all-photonic networks”, in Proceedings of IEEE BROADNETS, San Jose, CA, Oct. 2006.

[**Serpanos2000**] D.N. Serpanos and P.I. Antoniadis, “FIRM: a class of distributed scheduling algorithms for high-speed ATM switches with multiple input queues”, in Proceedings of the IEEE INFOCOM’2000, Tel-Aviv, Israel, 2000, pp. 548-555

[**Shah2002**] D. Shah and M. Kopikare, “Delay bounds for the approximate maximum weight matching algorithm for input queued switches”, in Proceedings of the IEEE INFOCOM’2002, New York, 2002, pp.1024-1031.

[Sinkhorn1964] R. Sinkhorn, “A relationship between arbitrary Positive matrices and doubly stochastic matrices”, *The Annals of Mathematical Statistics*, Vol. 35, No. 2, 1964, pp. 876-879.

[Sivakumar2004] M. Sivakumar and S. Subramaniam, “A performance evaluation of time switching in TDM wavelength routing networks,” in Proceedings of the 1st International Conference on Broadband Networks (Broadnets’04), San Jose, CA, Oct. 2004, pp. 212-221.

[**Stoica1998**] I. Stoica and H. Zhang, "Exact emulation of an output queueing switch by a combined input output queueing switch", in Proceedings of the 6th IEEE/IFIP IWQoS'98, Napa Valley, CA, May 1998, pp. 218-224.

[Subramaniam1996] S. Subramaniam, M. Azizoglu, and A. K. Somani, “All-optical networks with sparse wavelength conversion,” IEEE/ACM Transactions on Networking, Vol. 4, No. 4, Aug. 1996, pp. 544-557.

[**Tamir1988**] Y. Tamir and G. Frazier, “High performance multi-queue buffers for VLSI communication switches,” in Proceedings of the 15th Ann. Symp. Computer Architecture, June 1988, pp. 343–354.

[**Tassiulas1998**] L. Tassiulas, "Linear complexity algorithms for maximum throughput in radio networks and input queued switches", in Proceedings of IEEE INFOCOM’98, Vol. 2, 1998, pp. 533-539.

[**Towles2003**] B. Towles and W. J. Dally, "Guaranteed scheduling for switches with configuration overhead," IEEE/ACM Transactions on Networking, Vol. 11, No. 5, October, 2003, pp. 835-847.

[Turner1999] J. Turner. “Terabit burst switching”, International Journal of High Speed Networks, Vol. 8, No. 1, 1999, pp. 3-16.

[Vickers2000] R. Vickers and M. Beshai, “PetaWeb architecture,” in Proceedings of Networks 2000 Symposium, Toronto, Canada, 2000.

[Vinokurov2005] A. Vinokurov, X. Liu, and L. Mason, “Resource sharing for QoS in Agile All Photonic Networks”, in Proceedings of OPNETWORK 2005, Washington D.C., August 2005.

[Wang2003] S. Wang, D. Xuan, R. Bettati, and W. Zhao, “A study of providing statistical QoS in a differentiated services network”, in Proceedings of the 2nd IEEE International Symposium on Network Computing and Applications, 2003, pp. 297-304.

[Wen2002] B. Wen and K. M. Sivalingam, “Routing, wavelength and time-slot assignment in time division multiplexed wavelength-routed optical WDM networks,” in Proceedings of IEEE INFOCOM’02, New York, Jun. 2002, pp. 1142-1150.

[Xiao1999] X. Xiao and L. M. Ni., “Internet QoS: a big picture”, IEEE Network, Vol. 13, No. 2, March/April 1999, pp. 8-18.

[Yates1999] J. Yates, J. Lacey, and D. Everitt, "Blocking in multiwavelength TDM networks," Telecommunications Systems Journal, Vol. 12, 1999, pp.1-19.

[**Yim2004**] R Yim, N. Devroye, V. Tarokh and H.T Kung, "Achieving Fairness in Two-Dimensional Generalized Processor Sharing", in Proceedings of the 22nd Biennial Symposium on Communications, Queen's University, Kingston, Ontario, Canada, April, 2004, pp.185-187.

[Yoo1999] M. Yoo, C. Qiao. "Optical burst switching [OBS] – a new paradigm for an optical internet." International Journal of High Speed Networks, Vol. 8, No. 1, 1999, pp. 69-84.

[Zheng2004] J. Zheng and H. T. Mouftah, "Optical WDM networks: concepts and design principles", Wiley-IEEE Press, New Jersey, Jul. 2004.