

COVER PAGE

Measurement Based Resource Allocation for Interconnected WDM Rings

A. Bianco, G. Galante, E. Leonardi, F. Neri

Dipartimento di Elettronica, Politecnico di Torino

C.so Duca degli Abruzzi, 24, Torino, Italy

Phone: +39 011 5644098, Fax: +39 011 5644099

e-mail: [bianco,galante,leonardi,neri]@polito.it

Abstract

We present measurement based resource allocation scheme for interconnected WDM rings in a metropolitan area network named DAVID (Data And Voice Integration over D-WDM). The network has a two level hierarchical structure, with a backbone of optical packet routers inter-connected in a mesh, and metropolitan areas served by sets of optical rings connected to the backbone through devices called Hubs. The paper focuses on the operations of the media access protocol and on resource allocation schemes to be used in the metropolitan area network. A simple scheme for datagram (not-guaranteed) traffic is defined and its performance are examined mainly by simulation.

Keywords: Dense Wavelength Division Multiplexing, Metropolitan Area Networks, Optical Ring, MAC protocols

Correspondent author: Andrea Bianco

Dipartimento di Elettronica, Politecnico di Torino

Corso Duca degli Abruzzi 24, 10129 Torino, Italy

Phone: +39 011 5644098, Fax: +39 011 564 4099

E-mail: bianco@polito.it

Measurement Based Resource Allocation for Interconnected WDM Rings

A. Bianco, G. Galante, E. Leonardi, F. Neri *

Dipartimento di Elettronica, Politecnico di Torino, Torino, Italy. E-mail: {bianco,galante,leonardi,neri}@polito.it .

May 20, 2002

Abstract. We present measurement based resource allocation scheme for interconnected WDM rings in a metropolitan area network named DAVID (Data And Voice Integration over D-WDM). The network has a two level hierarchical structure, with a backbone of optical packet routers inter-connected in a mesh, and metropolitan areas served by sets of optical rings connected to the backbone through devices called Hubs. The paper focuses on the operations of the media access protocol and on resource allocation schemes to be used in the metropolitan area network. A simple scheme for datagram (not-guaranteed) traffic is defined and its performance are examined mainly by simulation.

Keywords: Dense Wavelength Division Multiplexing, Metropolitan Area Networks, Optical Ring, MAC protocols.

1. Introduction

The DAVID (Data And Voice Integration over D-WDM) project is part of the IST (Information Society Technology) Program sponsored by the European Community. Its aim is the design of an optical packet-switched network for the transport of IP traffic over metropolitan, national and international distances.

The DAVID network is designed to offer an optical transport format independent of the traffic type; the clients of the DAVID network are mainly IP routers and/or switches that collect traffic from legacy networks. The network is based on a hierarchical architecture consisting of several metropolitan area networks, named DAVID Metro networks, interconnected by a wide area optical backbone. We focus on the DAVID Metro network in this paper.

The DAVID Metro network consists of several uni-directional slotted optical physical rings interconnected in a star topology by a Hub. No optical buffering is required in the Metro; all the buffering is done in electronics at access nodes. The Hub functionality is ring interconnection; since the Hub is buffer-less, it behaves basically as a space switch. Ring interconnections are dynamically modified at the Hub following a scheduling algorithm. The aim of the scheduling algorithm is to provide an amount of bandwidth to ring pairs close to instantaneous (short-term) bandwidth requirements. The scheduling

* This work was supported by the E.C. under the DAVID contract.



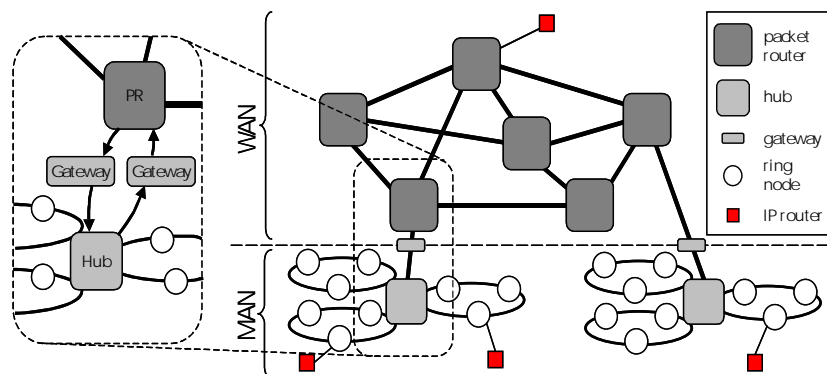


Figure 1. General overview of the DAVID network.

is based both on measurements at the Hub and on congestion signals issued by nodes. A WDMA/TDMA based MAC (Medium Access Control) protocol is defined to regulate access to shared network resources. A fairness protocol is added to guarantee throughput fairness among nodes on each ring.

The remainder of the paper is organized as follows. In Sect. 2 we give an overview of the DAVID network architecture. In Sect. 3 we focus on the Metro network, describing both the node and the Hub architecture. In Sect. 4 the MAC protocol and the scheduling algorithm at the Hub are described. In Sect. 5 we present simulation results to assess the performance of the proposed scheme.

2. Network Architecture

An overview of the two-level DAVID network architecture is shown in Fig. 1: several Metro networks are interconnected by a wide area network (WAN) backbone. Both network parts operate in packet switched mode. The backbone network consists of optical packet routers interconnected by a mesh network, while each Metro network comprises one or more rings interconnected through a Hub. Each ring collects traffic from several nodes and each Hub is connected to one or more optical packet routers in the WAN. Access points to the network are provided both in the Metro network and in the WAN, and the traffic is collected by IP routers and switches connected to local area networks (LANs). The network uses a mixed WDMA/TDMA access protocol: each fiber carries up to 32 wavelength channels at 2.5 or 10 Gbit/s, and time is divided into fixed size slots, each capable of carrying an optical packet consisting of a header and a payload.

In traditional packet switched networks, such as the Internet, buffering inside routers is needed to solve contentions arising among packets arriving

in a given node and headed to the same output port. In the DAVID WAN, optical packet routers attempt to solve contentions both exploiting wavelength diversity on optical fibers, and providing optical buffering by means of fiber delay lines.

No packet buffering in the optical domain is instead performed for packets flowing among ring nodes in the same Metro network. In a similar way, optical buffering is completely avoided along the node-to-Hub path for traffic exchanged among Metro nodes and nodes outside the Metro. Indeed, packets are buffered in ring nodes in the electrical domain, and are sent on the Metro network only when there are enough free resources on the Metro to travel from source to destination without being stored at any intermediate node. Thus, buffers are pushed towards the edge of the Metro network, and the sharing of rings resources among nodes must be regulated by a suitable MAC protocol. Instead, buffering and translation functions are confined at the interfaces between the WAN and Metro networks and are implemented in Gateways placed between optical packet routers and Hubs (see Fig. 1). Gateways participate in the MAC protocol, so that, from a logical point of view, connections from and to a Gateway appear to the Hub as additional Metro ring connections. The Hub is bufferless, as described later, and performs space switching and wavelength conversion only.

3. Metro Network

In general, a DAVID Metro network consists of several uni-directional optical physical rings interconnected in a star topology by a Hub. On each fiber, a fixed number of wavelengths is available by WDM partitioning. Logical rings can either be physically disjoint (i.e., run on different fibers), or be obtained by partitioning the optical bandwidth of one fiber into disjoint portions. Nodes belonging to the same logical ring access the same set of shared resources. In the remainder of the paper we use the term ring to identify a logical ring; any reference to physical rings will be explicit. The number of rings in a Metro network is denoted by N_{ring} .

While the number of wavelengths on each ring can be in general different, we assume that it is a multiple of the same number. In DAVID demonstrators this is dictated by technological constraints, since SOA arrays are used at each ring node to select the wavelengths from/to which packets are received/transmitted. Up to 32 wavelengths are available on each physical ring (fiber), and all wavelengths run at either 2.5 or 10 Gbit/s. We also assume that all the nodes of a ring can transmit and receive on any wavelength used in that ring. The latter is a rather essential assumption, since the access scheme would be much more complex if nodes had a limited tunability on the wavelengths of the ring they belong to. In particular, in this paper we assume

for simplicity that the same number of wavelengths (N_{chan} wavelengths) is available on each ring.

Ring resources are shared by the nodes of the Metro network using a statistical time/wavelength/space division scheme. Indeed,

- each wavelength is time slotted (TDM) and the slot duration is about 500 ns,
- several slots are simultaneously transmitted through wavelength division (WDM),
- rings can be disjoint in space (SDM).

Thus, resource sharing is based on a WDMA/TDMA scheme, i.e. a combination of Wavelength Division Multiple Access and Time Division Multiple Access.

Time slots are aligned on all wavelengths of the same ring, so that a multi-slot (a slot in each wavelength) is available to each node in each time slot. Slot misalignments among different Metro rings are solved optically at the Hub; we assume for simplicity that the propagation delay on each ring is an integer multiple of the slot size. One of the wavelengths (hence a slot in each multi-slot) is devoted to management and network control purposes. We assume that this control slot can be read and written by all nodes independently of their data transmissions and receptions in other slots of the multi-slot. The control information contained in a multi-slot refers to data slot in the same multi-slot; a delay is added in each node to process information contained in the control slot.

Wavelengths are (dynamically) assigned to ring-to-ring communications by the Hub on a slot-by-slot basis. To simplify the design of the Hub, we assume that all the wavelengths in a given multi-slot are devoted to transmissions to a given destination ring, identified with a label in the control slot. Any wavelength in the multi-slot may be used by ring nodes to reach nodes in the destination ring.

Metro ring nodes are subject to collisions and receiver contentions. By collision, we mean multiple transmissions in the same time slot, the same wavelength and the same physical ring. By receiver contention we mean having in the same multi-slot and the same ring a number of packets (in different wavelengths) to be received by a given node larger than the number of receivers available at that node.

Both collisions and contentions are avoided at each source node thanks to the MAC protocol, by monitoring the state of the incoming multi-slot, and giving priority to in-transit traffic. To avoid collisions, no new packet can be transmitted on a busy channel; to avoid contentions, if the number of packets in the current multi-slot for a given destination exceeds its capacity (i.e. number of receivers), no new packet can be transmitted to that destination.

Although contentions may arise in general also at the Hub, they are avoided by defining the Hub as a space switch and by running a proper slot scheduling algorithm.

3.1. RING NODE ARCHITECTURE

We assume that the number K of transceivers at each Metro ring node is smaller than the number of WDM channels; this means that a node can only transmit and receive on at most K channels at the same time, i.e. in each multi-slot. We typically consider the case $K = 1$; thus, each node has a single tunable transceiver: tuning actions are executed before transmitting and receiving independently at the transmitter and the receiver. We also assume that all the nodes of a ring can transmit and receive on any WDM channel used in the ring they belong to. Thus, ring nodes can drop, add and erase any packet on any wavelength at each time slot, whereas switching is forbidden for in-transit traffic: no operation is allowed on data not addressed to the node.

Packets waiting to be transmitted are grouped and stored per destination ring to avoid HoL (Head of the Line) blocking (Karol et al., 1987) typical of FIFO queues. Note that this queue architecture is very similar to the VOQ (Virtual Output Queue) architecture used in IQ (Input Queued) switches (McKeown et al., 1999), where, at each input port, packets are stored in separate queues on the basis of the destination port they should reach. Since resources (multi-slots and wavelengths) in DAVID are allocated to ring-to-ring communications, queues are organized per ring destination, i.e., at each node a FIFO queue is available to store packets directed to all the nodes belonging to a given ring. This avoids HoL blocking due to collision avoidance (since multi-slots are associated with destination rings), but does not solve HoL blocking due to receiver contentions, which would require a per-destination-node queuing scheme. The considered per-destination-ring queuing is however simpler to implement and to control, and scales much better to large network configurations.

3.2. HUB ARCHITECTURE

The role of the Hub is to switch packets between Metro rings, and from Metro rings towards the WAN (and vice-versa). Being all-optical, the Hub includes only a space switching stage, a wavelength conversion stage, and a WDM synchronisation stage; 3R regeneration may be added if necessary. Note that the target switching capacity in DAVID, given that in a typical Metro network N_{ring} rings running 32 wavelengths at 10 Gbit/s are envisioned, is 1.28 Tbit/s.

In every time slot, the Hub operates a permutation from input rings to output rings, as depicted in Fig. 2 for the case of four rings. This permutation is the same for all wavelengths of each ring and is known for each time slot in each ring: we can assume that each multi-slot is labeled by the Hub with

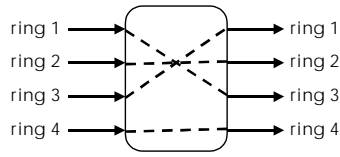


Figure 2. A ring-to-ring permutation at the Hub.

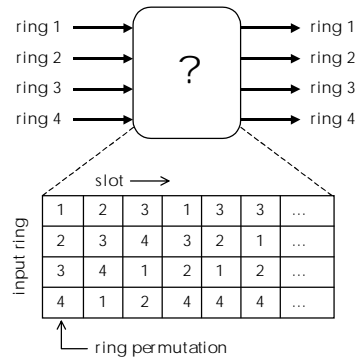


Figure 3. Scheduling at the Hub.

the identity of the ring to which packets transmitted in the multi-slot will be forwarded by the Hub.

Since we are assuming that the number of wavelengths in each ring is the same, no congestion occurs at the Hub: each incoming multi-slot can be forwarded to Hub outputs. The Hub must act as a non-blocking switch that is reconfigured in every time slot. It does not have to operate in the time domain, but it may have to perform wavelength conversion when the wavelengths used in the input ring are different from those used in the output ring (this always happens when the two rings are obtained in wavelength division on the same fiber).

The computation of the sequence of permutations operated by the Hub is a scheduling problem (Inukai, 1979; Hajek et Weller, 1997), as shown in Fig. 3. Several approaches can be envisaged to solve this problem, ranging from complex optimisations to simple heuristics, and are based onto an estimation of the ring-to-ring traffic pattern (note that the complexity of the scheduling problem depends on the number of rings, not on the number of nodes: this allows good scalability features). The scheduling algorithm is described in Sect. 4.4.

Given this Hub behaviour, each multi-slot traverses a sequence of rings, e.g. as illustrated in Fig. 4, where roman numbers indicate successive positions of the multi-slot, the upper slot is the control slot where the multi-slot

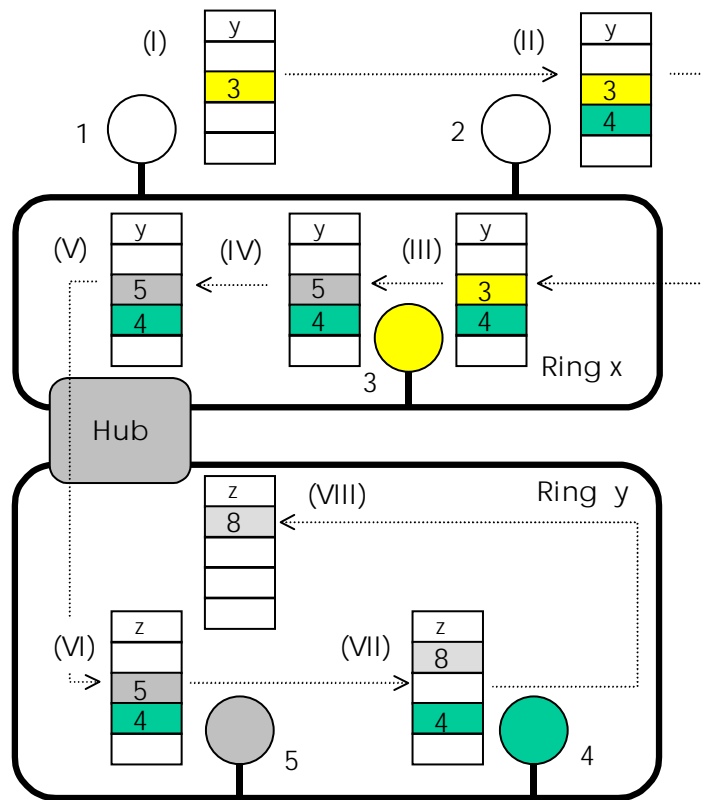


Figure 4. Multi-slot forwarding in the MAN. Numbers in slots represent packet destinations.

destination ring is written, and numbers within the multi-slot represent node destinations. Nodes of ring x transmit data to be received by nodes of ring y (Steps II to IV). Ring x can be viewed as the upstream ring, where transmissions occur, while ring y can be viewed as the downstream ring, where receptions occur. Note however that when the considered multi-slot traverses the downstream ring y (Steps VI to VIII), it gathers transmissions for the next ring, say ring z , so that the ring traversal can be viewed as a downstream path for transmissions in the previous ring, and as an upstream path for receptions in the following ring.

Space reuse of slots is possible in the DAVID Metro: a node receiving a packet leaves free the corresponding slot, which can be reused in the same ring, possibly by the same receiving node, for another transmission (see the transmission from node 1 to node 3 in Step I of Fig. 4). This also means that, in the example above, transmissions on upstream ring x can also be directed to other nodes of ring x (in addition to transmissions to nodes of downstream ring y). Note that transmissions to destinations belonging to the same ring of

the source node must go through the Hub when the destination precedes the source in the ring, hence Hub permutations in which the input and the output ring are the same are possible and required.

We inhibit these slot reuse capabilities in our simulations, and force all traffic to pass through the Hub before being removed from the ring by the destination node.

4. MAC Protocol and Scheduling at the Hub

In this section, we first describe the contention and collision problem in a DAVID Metro. Then, a simple access control scheme is proposed, and a fairness control is introduced to overcome the unfair behaviour of ring architectures. Finally, the scheduling algorithm at the Hub is discussed in detail.

4.1. CONTENTION AND COLLISION RESOLUTION

Receiver contentions are not recoverable (packets would be lost), unless complex receiver architectures are used. The proposed approach to solve contentions and collisions avoids packet losses in the path from the source node to the destination node, and is presented in the sequel. It is mainly achieved by the nodes, so that the operations and the implementation of the Hub are drastically simplified. In particular, no packet buffering, nor packet switching in the time domain, is required at the Hub.

4.2. THE ACCESS CONTROL SCHEME

In the description of the access control scheme, we assume for simplicity that the number of wavelengths supported on each ring is the same.

The choice of a ring for the DAVID Metro network significantly impacts the underlying framework in which the MAC protocol operates. Although the generic solutions befitting switches with VOQ architecture can be adapted to the ring topology, the nature of the ring, where the optical signals pass through all nodes, taking a round trip time for the collection of reservations, and for the distribution of state information, makes token based solutions more advantageous for this environment.

The state of each slot of the multi-slot is reflected in suitable fields of the control slot. Each node that has packets to send must monitor the control wavelength seeking an empty slot in any λ of a multi-slot that will be forwarded by the Hub to the corresponding destination ring. The node grabs the slot by setting the corresponding slot state field in the control slot, and by adding the destination address in the relevant field. The node must check before grabbing the slot that the intended destination does not already appear

in as many other λ s as the number of available tunable receivers ($K = 1$ in this paper), in which case it refrains from getting this slot and waits for the next opportunity.

Ring nodes also monitor the control wavelength looking for any instance of their address, in which case they tune to the indicated λ to receive the data contained in the corresponding slot. As mentioned above, we assume that each node introduces a traversal delay larger than the time required to process the control slot plus the time required to tune the receiver to the proper λ .

In summary, receiver contentions are solved assuming that the source node knows how many receivers are available at the destination node: transmission of a packet is forbidden if the number of packets sent by upstream nodes in the current multi-slot to the destination exceeds the reception capacity. To avoid collisions, an empty-slot protocol is used: incoming slots are inspected, and transmission is permitted in a slot on a given wavelength only if it is free, i.e., no upstream node transmitted in that slot and that wavelength. Note that this gives some advantage to upstream nodes, i.e., to nodes preceding others along the signal propagation direction: a given node can be completely starved by continuous transmissions of upstream nodes. This raises fairness issues, so that a protocol that provides fairness control is needed.

4.3. FAIRNESS CONTROL

As noted above, the proposed empty-slot operation can exhibit fairness problems under unbalanced traffic; this is particularly true in the ring topology, in which upstream nodes have generally better access chances than downstream nodes.

Credit-based schemes, such as the Multi-MetaRing (Ajmone Marsan et al., 1999) previously studied in a single ring context can enforce throughput fairness. MetaRing (Cidon and Ofek, 1993) was proposed by Y. Ofek for ring-based, electronic metropolitan area networks. It is basically a generalisation of the token-ring technique: a control signal or message, called SAT, is circulated in store-and-forward mode from node to node along the ring. A node forwarding the SAT is granted a transmission quota: the node can transmit up to Q packets before the next SAT reception. When a node receives the SAT, it immediately forwards the SAT to the next node on the ring if it is satisfied (hence the name SAT), i.e. if no packets are waiting for transmission on the ring, or Q packets were transmitted since the previous SAT reception. If the node is not satisfied, the SAT is kept at the node until one of the two conditions above is met. Thus, the SAT is delayed by nodes suffering throughput limitations, and the SAT rotation time increases with the network load. To be able to provide the full bandwidth to a single node, the quota Q must be at least equal to the number of data slots contained in the ring, i.e., proportional

to the ring latency (propagation delay) measured in slot times. In overload, each node sends exactly Q packets per SAT rotation time.

In the case of the DAVID MAN, several rings exist, and multi-slots traverse pairs of rings. We therefore need a SAT for each ring pair (upstream ring, downstream ring). SAT signals can be carried in the multi-slot control wavelength. The Hub must be able to store N_{ring}^2 SATs, where N_{ring} is the number of rings attached to the Hub. Since SATs do not carry any information, N_{ring}^2 boolean variables $\text{SAT}_{i,j}$ do the job; $\text{SAT}_{i,j}$ is TRUE when the SAT regulating transmissions from ring i to ring j is at the Hub. When the Hub issues on ring i a multi-slot that will be switched, upon return to the Hub, to ring j , if the SAT $i \rightarrow j$ is currently at the Hub (i.e., if $\text{SAT}_{i,j} = \text{TRUE}$), the SAT is loaded in the control slot of the multi-slot, by setting a suitable bit, and by setting $\text{SAT}_{i,j}$ to FALSE at the Hub.

Each node inspects the control slot of incoming multi-slots, and operates on SATs as described above for the single ring case. Recall that each queue is regulated by a different SAT and transmission opportunities are regulated by a MetaRing quota Q that may be different for each queue; however, the quota Q must be greater or equal to the ring latency to allow a single node to grab all the available bandwidth; thus, since in this paper we assume that all ring latencies are equal, we use the same value of the quota Q for all queues.

SATs are also used to trigger congestion notification signals from ring nodes to the Hub. This information is used by the Hub to determine the scheduling in successive frames as described later.

4.4. A SIMPLE SCHEDULING ALGORITHM

We describe the approach followed to compute the scheduling at the Hub; the algorithm is run in a centralised fashion at the Hub. Multi-slots are labelled at the Hub according to the outcome of the scheduling algorithm, using the control slot to identify the ring to which the multi-slot will be forwarded upon return to the Hub. Only unicast transmissions are considered, i.e. multicast transmissions are considered as multiple unicast transmissions.

The Hub scheduler is driven by an $N_{\text{ring}} \times N_{\text{ring}}$ request matrix \mathbf{R} . Each element $\mathbf{R}_{i,o}$ contains the number of multi-slots that must be transmitted from input ring i to output ring o , i.e., the number of multi-slots labelled with o in the control channel that the Hub must send on ring i , and, upon arrival at the Hub, switch to ring o . This request matrix is obtained by mixing periodic measurements and congestion signals issued by nodes as described below.

According to combinatorial theory (Hall, 1969), \mathbf{R} can be scheduled in at most F time slots, where the frame length F is equal to:

$$F = \max \left\{ \max_o \sum_i \mathbf{R}_{i,o}, \max_i \sum_o \mathbf{R}_{i,o} \right\}$$

by using a sequence of F switching matrices $\mathbf{P}(i)$, $i \in \{1, 2, \dots, F\}$, of size $N_{\text{ring}} \times N_{\text{ring}}$. A switching matrix is a binary matrix whose element $\mathbf{P}_{i,o}$ is 1 when input ring i is connected to output ring o , and 0 otherwise. The resulting scheduling is then repeated until a new value for \mathbf{R} becomes available and a new matrix decomposition can be computed. Traffic from ring i to ring o is thus served with a rate proportional to $\mathbf{R}_{i,o}/F$.

Each switching matrix represents the Hub switching configuration in a given time slot. Since each input ring can be connected to at most one output ring and each output ring can be connected to at most one input ring in each time slot, a switching matrix always contains at most one non-null element in each row and in each column. The outcome of the Hub scheduling algorithm is a set of F ring permutations; hence, in each time slot, one and only one element from each row and one and only one element from each column must be equal to 1 in \mathbf{P} . In other words, we are interested in doubly stochastic switching matrices, i.e. matrices \mathbf{P} such that $F = \sum_i \mathbf{P}_{i,o} = 1, \forall o$, and $\sum_o \mathbf{P}_{i,o} = 1, \forall i$. A scheduling algorithm generating a sequence of F doubly stochastic switching matrices accommodates a request matrix \mathbf{R}^F where each row and each column sum to F , a condition that in general does not hold for \mathbf{R} . We artificially add integer quantities, representing ring-to-ring multi-slot requests, to some elements in the original matrix \mathbf{R} , to obtain the matrix \mathbf{R}^F to be scheduled. Any algorithm can be used to obtain a matrix \mathbf{R}^F satisfying this condition; see e.g. (Chang et al., 2000).

The matrix \mathbf{R}^F may be associated with a bipartite graph G having $2N_{\text{ring}}$ nodes. Each node represents either one input or one output of the switch, and input node i is connected to output node o by one edge only if $\mathbf{R}_{i,o} \neq 0$. A *matching* on G is a subset E of the edges in G such that, each node in G is incident to at most one edge in E . The number of edges in E is the *size* of the matching. A matching is said to be *maximum* when it has maximum size (Tarjan, 1983).

We may apply an iterated maximum size algorithm on \mathbf{R}^F to obtain the Hub scheduling, i.e., a sequence of doubly stochastic $\mathbf{P}(i)$, $i \in \{1, 2, \dots, F\}$. At step i , the decomposition algorithm computes the switching matrix $\mathbf{P}(i)$ as a maximum size matching on \mathbf{R}^F . Then, $\mathbf{P}(i)$ is subtracted from \mathbf{R}^F , and a new iteration is started. At the end of iteration F , the obtained sequence of matrices $\mathbf{P}(i)$ is randomly shuffled to uniformly distribute ring-to-ring pairs in the frame, to reduce traffic burstiness.

Another possible algorithm that may be used is a critical maximum matching on \mathbf{R} . Any input i for which $\sum_o \mathbf{R}_{i,o} = F$, and any output o for which $\sum_i \mathbf{R}_{i,o} = F$ is said to be *critical*, since it must be served in every time slot if \mathbf{R} must be scheduled in F slots. A *critical maximum matching* is a maximum matching which covers all the critical input and output nodes. The request matrix \mathbf{R} is decomposed into F switching matrices through iterated application of the critical maximum matching algorithm (Hajek et Weller,

1997) as done above for the maximum size matching. The only difference stems from the fact that, occasionally, the size of a critical maximum matching may be lower than N_{ring} . In this case, matrix $\mathbf{P}(i)$ must be completed so that all input rings are always connected to all output rings. At step i , the decomposition algorithm computes the switching matrix $\mathbf{P}(i)$ as a critical maximum matching on \mathbf{R} . When the matching has size lower than N_{ring} , matrix $\mathbf{P}(i)$ is completed so that all input rings are always connected to all output rings. Finally, $\mathbf{P}(i)$ is subtracted from \mathbf{R} , and a new iteration is started. At the end, the matrices $\mathbf{P}(i)$ are randomly shuffled as above to reduce traffic burstiness.

Traffic Measurement

The request matrix \mathbf{R} used by the scheduling algorithm is estimated on the basis of traffic measurements performed at the Hub during consecutive observation windows (OW); the duration of each OW is fixed and roughly equal to 10 ring propagation times.

The key idea of the algorithm is that, as long as the network is not overloaded, the throughput is a good estimator of the offered load. When one or more traffic relations among different ring pairs become overloaded, congestion control mechanisms are introduced to modify the bandwidth allocation in the network. Note that overloading conditions depend on the scheduling at the Hub. If the scheduling determined at the Hub is not matched to the traffic distribution, some ring experience overloading conditions until the scheduling is not modified, since the scheduling determines bandwidth allocation among ring-to-ring pairs.

The matrix \mathbf{R} that must be scheduled is computed, at the end of each OW, as the weighted sum of three contributions (consisting of $N_{\text{ring}} \times N_{\text{ring}}$ matrices):

$$\mathbf{R} = [\mathbf{SM} + \beta \mathbf{IC} + \gamma \mathbf{EC}]$$

with β and γ positive constants where:

- **SM** (smoothed measure) is a measure of the (long term) average number of multi-slots transmitted among ring pairs; each element is a real number ranging between 0 and OW; this is an absolute throughput measure;
- **IC** (implicit congestion) is the percentage of filled slots; each element is represented as a real number between 0 and 1; this is a relative throughput measure;
- **EC** (explicit congestion) takes into account explicit congestion signals sent by ring nodes, where each element is either 0 or 1.

The Hub stores in each element $M_{i,o}$ of matrix \mathbf{M} (measured) the number of packets flowing from ring i to ring o during each OW. The matrix \mathbf{M} is

then passed through an exponential filter to smooth out measurement errors, obtaining matrix \mathbf{SM} . Thus, a new value for \mathbf{SM} is computed at the end of each OW as a function of the last measured matrix \mathbf{M} and of the values assumed by \mathbf{SM} at the end of the previous OW:

$$\mathbf{SM}_{\text{new}} = \alpha \mathbf{SM}_{\text{old}} + (1 - \alpha) \mathbf{M}/N_{\text{chan}}$$

where $\alpha \in [0, 1]$ is a constant, N_{chan} is the number of wavelengths channels available on a logical ring, which we assume to be equal for all Metro rings; matrix \mathbf{M} is divided by N_{chan} to convert number of packets in number of multi-slots. Therefore, element $\mathbf{SM}_{i,o}$ of \mathbf{SM} is the average number of multi-slots transmitted from ring i to ring o during one OW, roughly averaged over the last $1/\alpha$ observation windows.

Matrix \mathbf{IC} gives the ring-to-ring throughput measured at the Hub, i.e., the occupation of scheduled slots. Each element $\mathbf{IC}_{i,o}$ is the ratio of the number of packets sent from ring i to ring o over the number of slots available for transmission on the same traffic relation in one OW. If $\mathbf{IC}_{i,o}$ is close to 1, this is a signal of potential congestion between i and o .

Matrix \mathbf{EC} is a binary matrix which provides information on the ring congestion level on the basis of nodes queue length. Congestion signals are triggered at nodes by SAT transmissions. Each node on ring i , when releasing $\mathbf{SAT}_{i,o}$, checks the length of the queue toward ring o ; if the queue contains more than $L_{\text{thr}} = Q$ packets (where Q is the MetaRing quota), the node sends, on the control channel, a congestion signal to the Hub. Note that we use the control channel to send congestion signals to the Hub instead of SAT messages, since SAT messages may be delayed by downstream nodes experiencing difficulties in channel access. Each element $\mathbf{EC}_{i,o}$ in \mathbf{EC} is set to 1 at the Hub, if the Hub has received at least one congestion signal toward ring o from a node on ring i during the last OW. The value of L_{thr} is related (equal in our simulations) to the MetaRing quota Q ; the rationale is that the quota represents, for each node, transmission opportunities toward a given ring between two consecutive SAT receptions. We assume congestion if the number of packets still in the queue when releasing the SAT is greater than the MetaRing quota, since the node will not be able to transmit all queued packets in the following SAT rotation time.

The two congestion signals operate on two different time scales: the first indication, stored in \mathbf{IC} , is related to the observation window, which is fixed; the second indication, stored in \mathbf{EC} is triggered by SAT arrivals, and depends on the SAT rotation time, which in turn depends on the number of nodes in the network. Moreover, the implicit congestion signal can be used as an early congestion signal indication, to trigger an increase in slot allocation to a given ring pair without waiting until the queue size in a node exceeds the threshold.

We want to highlight some properties of the presented algorithm. Suppose that the network is not overloaded, since the scheduling algorithm at the Hub

provides enough slots (bandwidth) to each ring-to-ring traffic relation. This means that the scheduling determines a slot allocation matched to the offered traffic, i.e. a slot allocation that satisfies all traffic relations, which are never congested. This is the solution we would like to obtain with our algorithm under stationary traffic conditions. Congestion signals are never issued, since nodes do not experience congestion. Thus, the frame length is determined by the scheduling on matrix $\mathbf{R} = \mathbf{SM}$; the measured average slot occupation is proportional to OW via the network load ρ . All the simulation results show that if the network is not overloaded, the frame length is close to this value. This feature is obtained because the measurement interval is fixed. If we had a variable measurement interval proportional to the frame length, we would have obtained a shrinking frame length, since each measurement would create a matrix \mathbf{R} where each element is on average reduced by a factor ρ with respect to the value assumed in the previous interval. On the other hand, a fixed measurement interval raises the problem of deciding a value for such interval, which indirectly decides also the granularity in bandwidth allocation and control. Recall that we chose the measurement interval to be equal to 10 ring latencies in our simulation experiments.

Finally, we must ensure that the scheduling provides at least a multi-slot for each ring-to-ring pair, i.e. at least a set of covering permutations must be scheduled in the frame, so that at least one multi-slot is available in each ring to send packets to any other ring. Otherwise, if no traffic exists on a given ring-to-ring pair, it is not possible to measure any slot occupation, the SAT cannot be sent and explicit congestion signals cannot be raised by nodes and no implicit congestion signal may be measured at the node. We enforce the scheduling to provide this set of N_{ring} covering permutations in each frame.

5. Simulation Results

We present some simulation results to assess the performance of the proposed access scheme. We do not exploit the space reuse capability described at the end of Sect. 3.2: if a multi-slot on ring x is labelled with destination ring z , it is used only to send traffic to nodes in ring z . Moreover, we force all packets sent on ring x and addressed to ring x (inter-ring traffic) to pass through the Hub; this is required to allow the Hub to perform traffic measurement for all ring pairs.

In our simulation experiments the Metro network comprises $N_{\text{ring}} = 4$ rings, with $N_{\text{node}} = 10$ nodes on each ring. For each ring-to-ring communication $N_{\text{chan}} = 4$ data channels are available; thus, each multi-slot comprises 5 slots, 4 for data traffic and 1 for control and management. Each node stores packets in 4 queues, one for each destination ring. Each queue is 1000 packets long, and the packet size is matched to the slot size. The

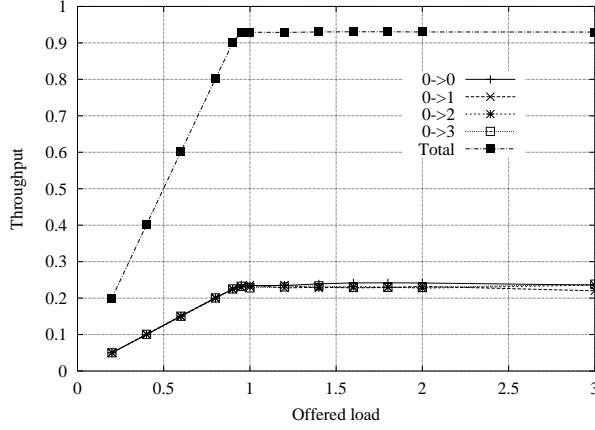


Figure 5. Throughput under uniform traffic.

ring round trip time is assumed to be equal to 44 time slots, the MetaRing quota is $Q = 44$, and the threshold $L_{\text{thr}} = Q$. The observation window is $OW = 440$ time slots. The “standard” values used in our simulation experiments for the parameters defined in the measurement algorithm are the following: $\beta = 10$, $\gamma = 3$, $\alpha = 0.1$. No optimization has been attempted on these values. We will show later that these values may be non optimal, but also that the scheme is very robust to parameters variations.

We consider three traffic patterns: a uniform traffic pattern, an unbalanced traffic pattern and a randomly generated time-variant traffic pattern. We define the weight matrix \mathbf{W} , of size $N_{\text{ring}} \times N_{\text{ring}}$, where each element $\mathbf{W}_{i,o}$ is a real number ranging between 0 and 1, representing the percentage of traffic generated on ring i toward ring o with respect to the total network load ρ . Clearly, $\sum_i \mathbf{W}_{i,o} \leq 1, \forall o$, and $\sum_o \mathbf{W}_{i,o} \leq 1, \forall i$. In the uniform traffic pattern $\mathbf{W}_{i,o} = 1/N_{\text{ring}} \forall i, o$. For the unbalanced traffic pattern $\mathbf{W}_{i,o} = 0.7$ when $i = o$, and $\mathbf{W}_{i,o} = 0.1$ otherwise; in other words, the ratio among intra-ring traffic and inter-ring traffic is 7. In the random pattern, the values of the elements $\mathbf{W}_{i,o}$ are time-variant. Every 200,000 time slots, each element in the matrix is randomly selected; the only constraint we impose is that the sum of the elements in each row and in each column is equal to 1.

Packets are generated at ring nodes according to a Bernoulli distribution whose average is derived from the weight matrix described above.

We first plot the throughput (ratio between used and allocated slots) for each destination ring on ring 0; this is a steady-state value obtained using statistically significant measures by simulation. Note that, although we plot the throughput for a single ring, the same behaviour holds for all other rings due to traffic symmetries. Nodes on the same ring do not exhibit throughput unfairness thanks to the MetaRing algorithm.

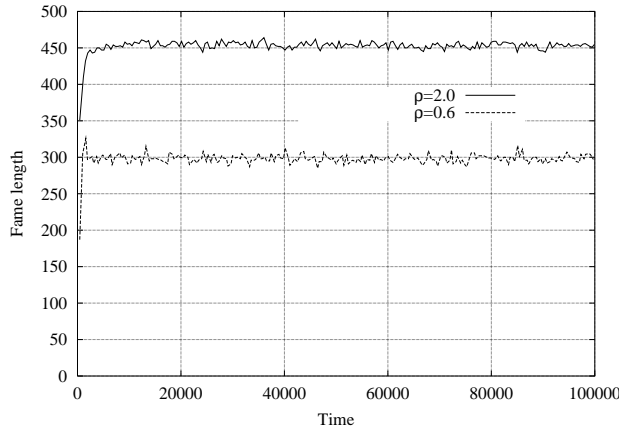


Figure 6. Frame length as a function of time under uniform traffic.

In Fig. 5 we report the throughput for each destination ring on ring 0, and the overall network throughput (black square markers) as a function of the offered load under uniform traffic. Each destination ring is treated fairly and the total network utilisation is close to 0.95. Although we significantly overload the network, since ρ ranges from 0.1 to 3, the algorithm behaves well even under these extreme conditions. The 5% utilisation loss is small, given the complexity of the system, and it can be shown to be mainly due to receiver contentions, which can be analytically evaluated (see the Appendix).

Fig. 6 shows the frame length as a function of time. We start with a uniform scheduling with a frame equal to N_{ring} slots. As expected, the system converges to a frame length roughly equal to $OW \times \min(\rho, 1)$; once this value is reached, the frame length changes slowly following traffic fluctuations. The convergence speed is mainly determined by the value of parameter α .

In Fig. 7 we report the throughput for each destination ring on ring 0, and the overall ring throughput (black square markers) as a function of offered load under unbalanced traffic. For values of ρ ranging from 0.1 to 1, the throughput is proportional to the weight matrix defined for the unbalanced traffic scenario. As soon as the offered load ρ increases to values that create congestion, the scheduling algorithm treats all ring-to-ring connections fairly according to a max-min-fairness like criterion (Bertsekas and Gallager, 1987); the intra-ring throughput decreases steadily until it reaches the same throughput obtained by inter-ring connections. Also in this scenario each destination ring is treated fairly and the total network utilisation is close to 0.95.

We examine in Fig. 8 the bandwidth allocation determined by the scheduling algorithm for ring 0 under unbalanced traffic for $\rho = 0.6$ (similar curves are observed for other values of ρ). The allocation is sampled at intervals

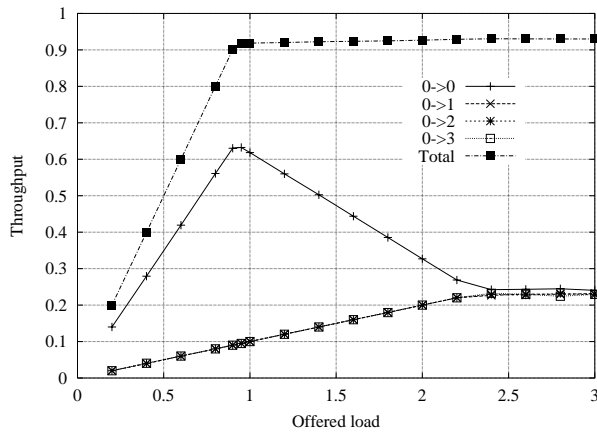


Figure 7. Throughput under unbalanced traffic.

lasting OW, the observation window. The ideal scheduling algorithm would allocate steadily bandwidth equal to 0.7 for the connection from ring 0 to ring 0, and 0.1 to all other inter-ring connections. In our experiment, the initial scheduling algorithm is matched to a uniform traffic pattern, which is clearly not optimal for unbalanced traffic. We can observe a transient behaviour of less than 2000 slot times (roughly 4 observation windows); this value depends on the choice made for the parameters defined in the measurement algorithm. After the initial transient, the allocation becomes close to the optimal one, with some small variations of few % around the ideal value; these differences are due to traffic fluctuations, to which the scheduler tries to adapt the bandwidth allocation, and to inaccuracies in the traffic measurement process. The choice of the parameters should be optimised to control these fluctuations under all traffic conditions. We observed that the algorithm does not exhibit any drift from the optimal values also under heavy load conditions.

In Fig. 9 we show the queue occupancy (in packets) in sustained overload ($\rho = 2.0$) for a given node on ring 0, sampled every 1000 slot times. All other nodes show similar queue length behaviours. Whereas the queue length for intra-ring traffic saturates since this connection is overloaded, all other queues show oscillating behaviours, since inter-ring connections become congested only when the scheduling does not allocate enough slots to them. Remarkably, although the algorithm aims only at fair bandwidth allocation, the queue occupancy level is fairly well controlled, at values smaller than 100 packets, a value not far from the observation window OW, which is the time constant under which no bandwidth control can be achieved in this network.

In the same scenario, by tuning the parameter values, we were able to obtain a more controlled queue behavior, as shown in Figs. 10 and 11, where we set $\gamma = 0$, and $\beta = 10$ and $\beta = 20$ respectively. However, note that we

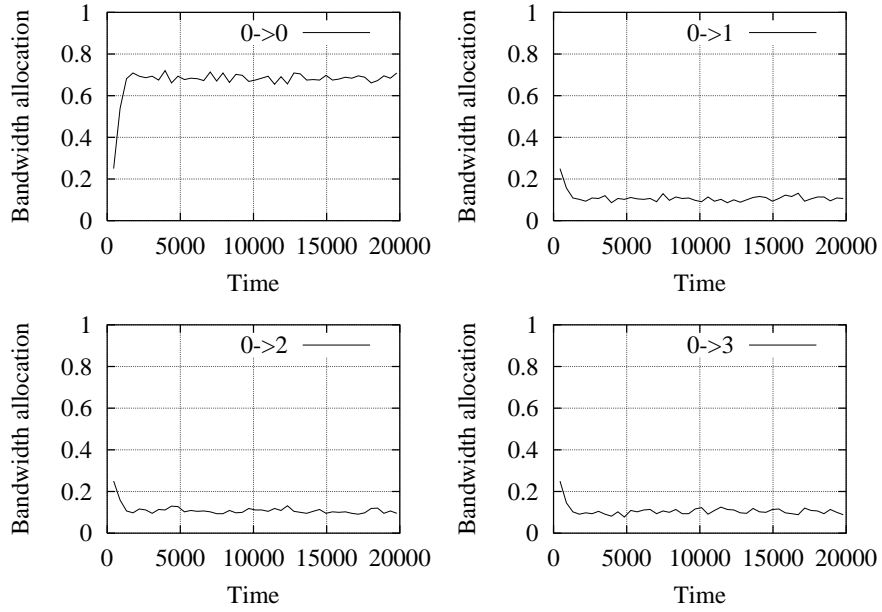


Figure 8. Bandwidth allocation under unbalanced traffic when $\rho = 0.6$.

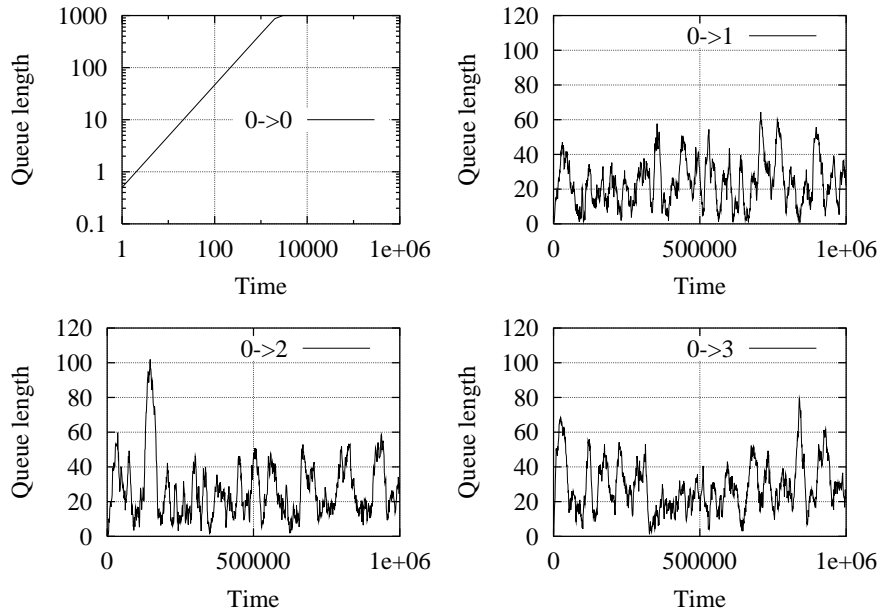


Figure 9. Queue length for unbalanced traffic with $\rho = 2.0$, $\alpha = 0.1$, $\beta = 1$, $\gamma = 3$.

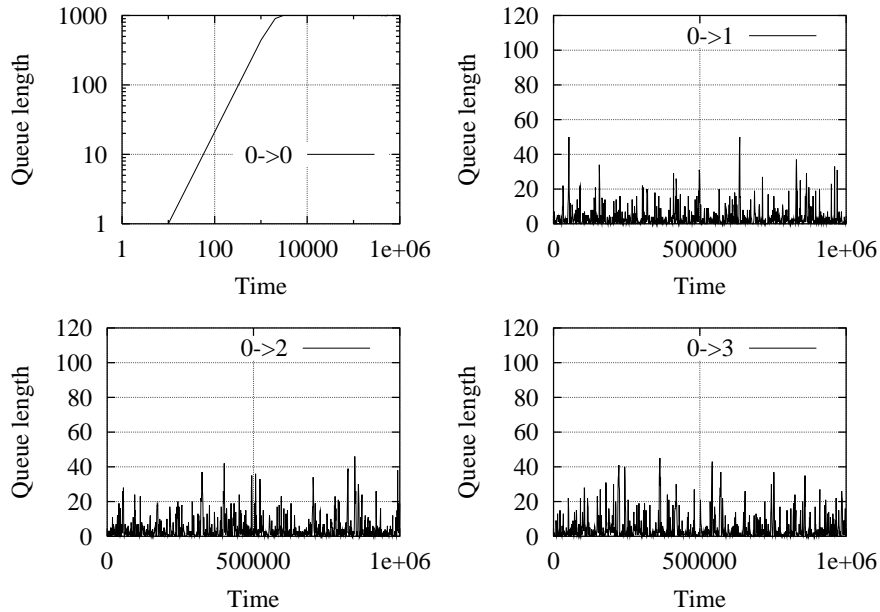


Figure 10. Queue length for unbalanced traffic with $\rho = 2.0$, $\alpha = 0.1$, $\beta = 10$, $\gamma = 0$.

are looking at queue lengths, which are a very difficult parameter to control. The scheme aims at good and fair bandwidth allocation and no significant differences were observed in terms of throughput by varying these parameters in this scenario. Moreover, even with our standard parameter values, queue lengths are controlled at values close to the ring round trip time, which is a remarkable result.

Finally, we present results for randomly generated traffic, varied every 200,000 slots. For network load up to 0.85, the system shows very good performance, as shown in Figs. 12, 13, and 14, for queue lengths, ring-to-ring throughput, and total network throughput. Transients are well controlled and the network utilization is very high.

In overload conditions, with $\rho = 0.95$, the system shows some throughput losses; packet losses are experienced (Fig. 15) since the system is not fast enough in following traffic fluctuations, as it can also be seen observing ring-to-ring throughputs in Fig. 16.

6. Conclusions and Future Work

The paper presented a measurement-based resource allocation scheme designed for empty-slot WDM rings interconnected by a Hub acting as a ring-

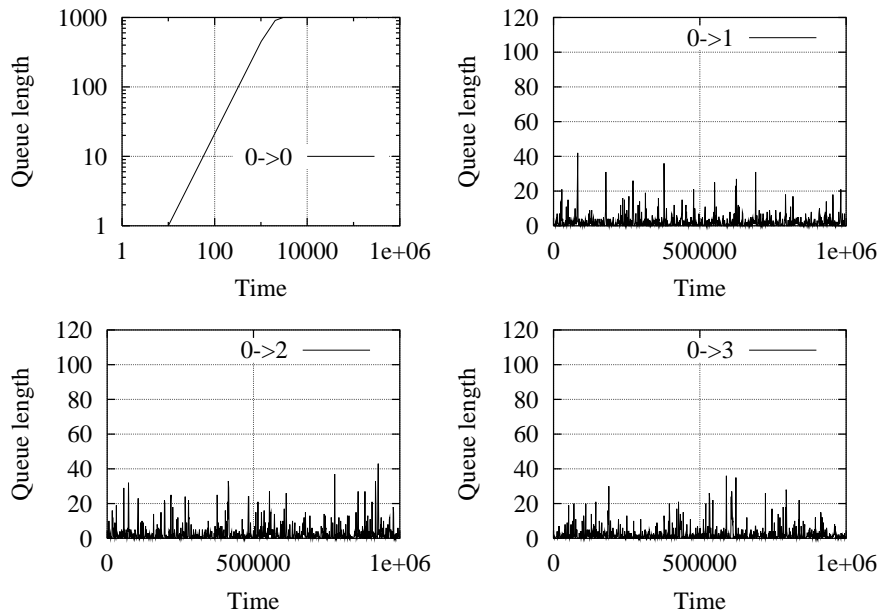


Figure 11. Queue length for unbalanced traffic with $\rho = 2.0$, $\alpha = 0.1$, $\beta = 20$, $\gamma = 0$.

to-ring permutator. The work was done in the framework of the European project DAVID, under Alcatel leadership.

Although the presented simulation results are encouraging and the algorithm shows good performance and robustness to the parameter setting, several issues remain to be addressed. Other traffic scenarios should be studied to prove the algorithm robustness to different environments. Different traffic patterns should be examined, and traffic generation should be extended from Bernoulli to on-off and/or heavy-tailed traffic models. Transient behaviours must be more carefully analysed to test the ability of the algorithm to follow short-term traffic fluctuations. Finally, we want to extend the proposal to deal with multiple classes of traffic, to provide QoS guarantees similar to those of the DiffServ environment defined by the IETF.

References

- Ajmone Marsan M., Bianco A., Leonardi E., Morabito A., Neri F. All-Optical WDM Multi-Rings with Differentiated QoS. *IEEE Communications Magazine, Feature topic on Optical Networks, Communication Systems and Devices*, M. Atiquzzaman, M. Karim (eds.), Vol. 37, No. 2, Feb. 1999, pp. 58–66.
- Bertsekas D., Gallager R. *Data networks*. Prentice-Hall, 1987.

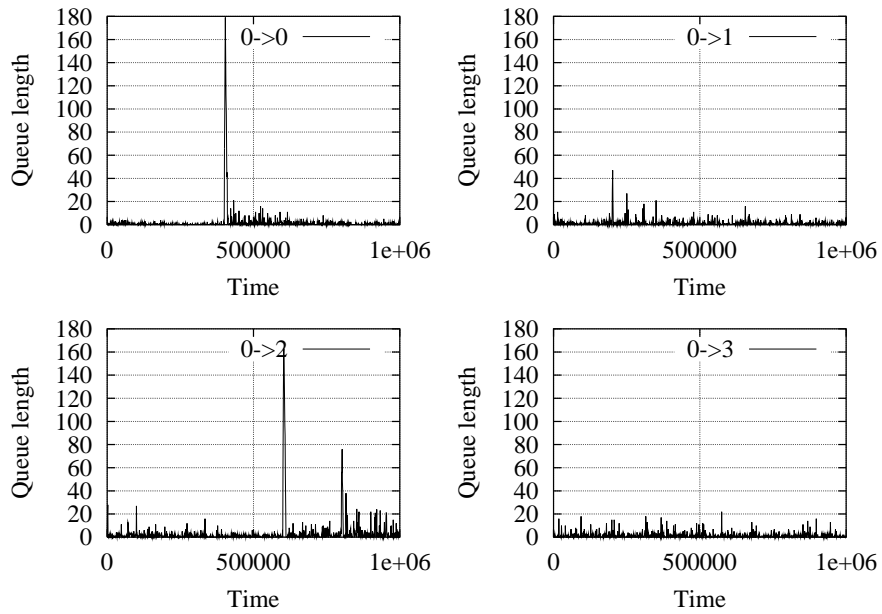


Figure 12. Queue length for random traffic with $\rho = 0.85$, $\alpha = 0.1$, $\beta = 10$, $\gamma = 3$.

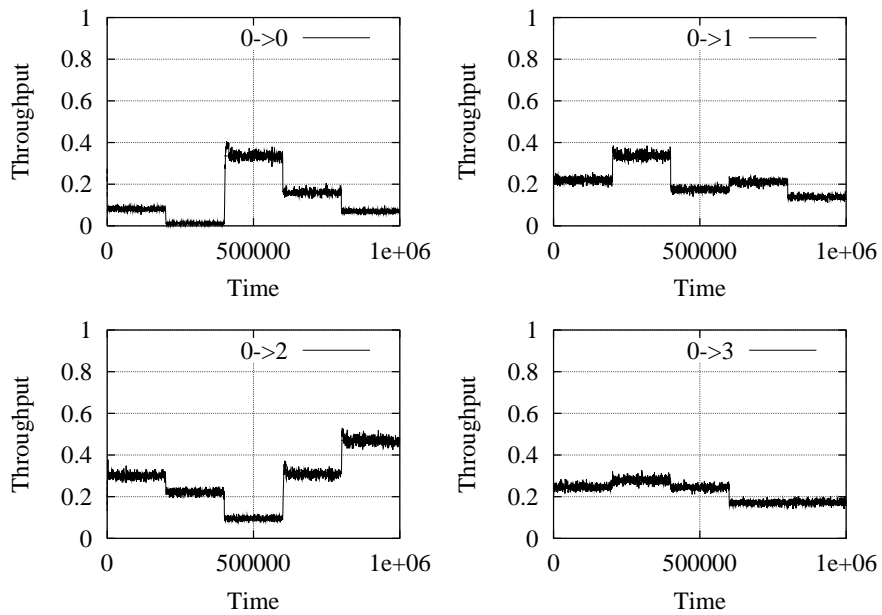


Figure 13. Throughput for random traffic with $\rho = 0.85$, $\alpha = 0.1$, $\beta = 10$, $\gamma = 3$.

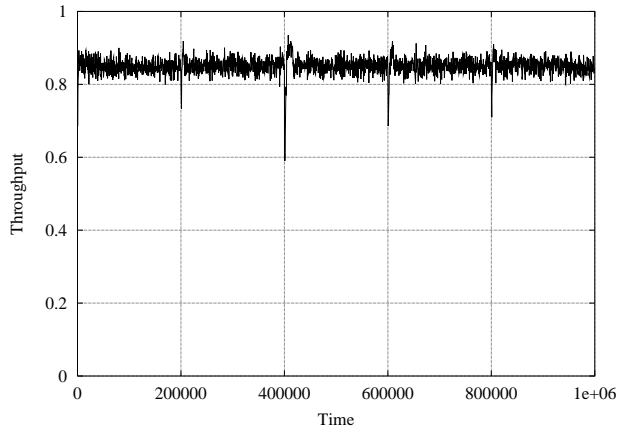


Figure 14. Total throughput for random traffic with $\rho = 0.85$, $\alpha = 0.1$, $\beta = 10$, $\gamma = 3$.

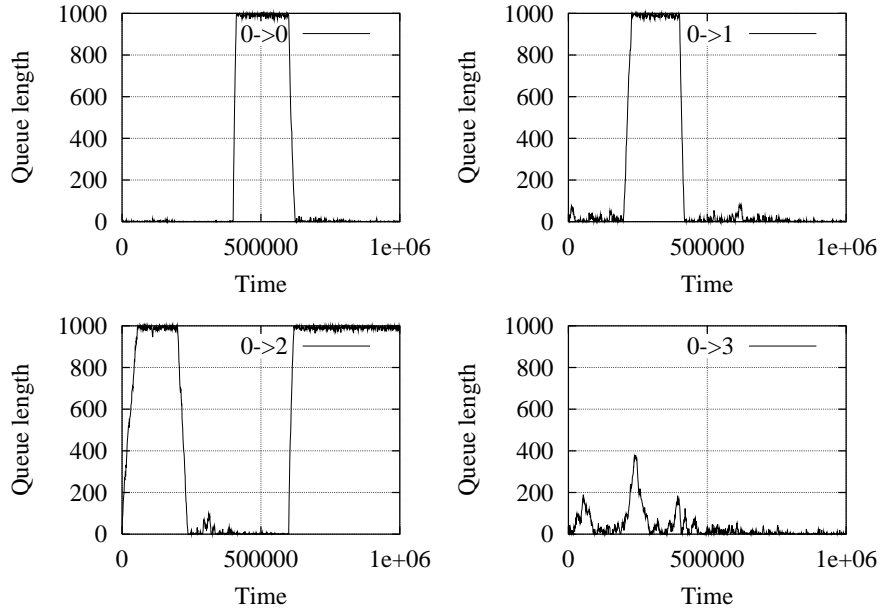


Figure 15. Queue length for random traffic with $\rho = 0.95$, $\alpha = 0.1$, $\beta = 10$, $\gamma = 3$.

Chang C.S., Chen W.J., Huang H.Y. Birkhoff-von Neumann Input Buffered Crossbar Switches. *IEEE Conference on Computer Communications (INFOCOM 2000)*, Tel Aviv, Israel, March 2000, pp. 1614–1623.

Cidon I., Ofek Y. MetaRing — a Full-Duplex Ring with Fairness and Spatial Reuse. *IEEE Transactions on Communications*, Vol. 41, No. 1, Jan. 1993, pp. 110–120.

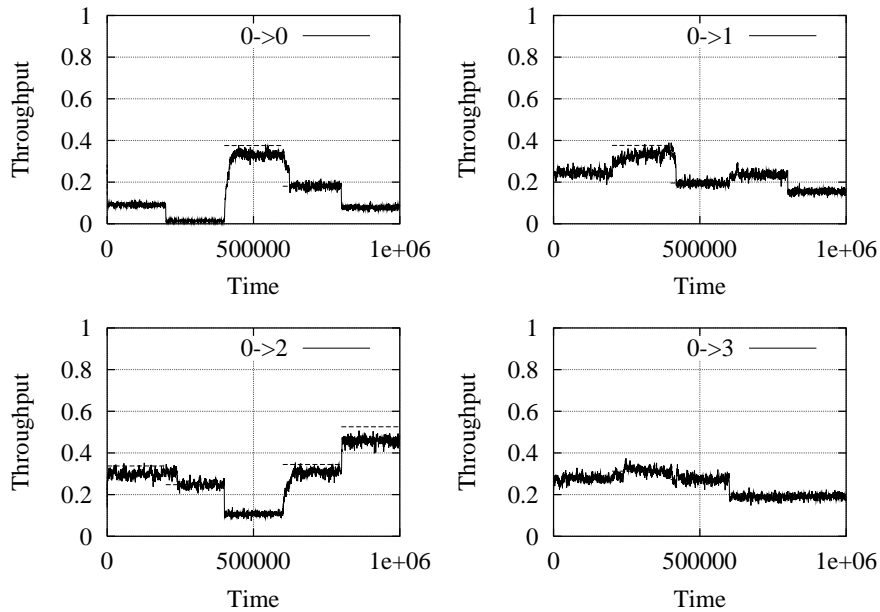


Figure 16. Throughput for random traffic with $\rho = 0.95$, $\alpha = 0.1$, $\beta = 10$, $\gamma = 3$.

Hajek B., Weller T., Scheduling Non-Uniform Traffic in a Packet-Switching System with Small Propagation Delay. *IEEE Transactions on Networking*, Vol. 5, No. 6, Dec. 1997, pp. 813–823.

Hall M. Jr. *Combinatorial Theory*. Waltham, MA, Blaisdell, 1969.

Inukai T. An efficient SS/TDMA time slot assignment algorithm. *IEEE Transactions on Communications*, Vol. 27, Oct. 1979, pp. 1449–1455.

Karol M., Hluchyj M., Morgan S., Input Versus Output Queuing on a Space Division Switch. *IEEE Transactions on Communications*, Vol. 35, No. 12, Dec. 1987, pp. 1347–1356.

McKeown N., Mekkittikul A., Anantharam V., Walrand J. Achieving 100% throughput in an input-queued switch. *IEEE Transactions on Communications*, Vol. 47, No. 8, Aug. 1999.

Tarjan R.E. *Data Structures and Network Algorithms*. Society for Industrial and Applied Mathematics. Pennsylvania, November 1983.

Appendix

We present an analytical Markovian model for the evaluation of the maximum throughput achievable in the DAVID Metro network, assuming uniform traffic, and $K = 1$, i.e., a single transceiver per node.

Let us consider a multi-slot during its journey along ring i . We suppose that the considered multi-slot is used by nodes on ring i to transmit packets to nodes on ring j . The multi-slot state at a given instant of time can be represented by an ordered pair of integer indices (n_t, n_r) , which represent

respectively the number of packets directed to ring j that have been transmitted by nodes on ring i already visited by the multi-slot, and the number of packets directed to nodes on ring i that have not been received yet. Node k on ring i can both transmit a new packet directed to ring j , and extract from the ring a packet directed to it. Given the multi-slot state $M^k = (n_t^k, n_r^k)$ just before visiting node k , the transmission and reception probabilities of node k , P_t^k and P_r^k , can be easily computed as follows:

$$P_t^k = \begin{cases} 0 & \text{if } n_t^k + n_r^k = N_{\text{chan}} \\ 1 - n_t^k / N_{\text{node}} & \text{if } n_t^k + n_r^k < N_{\text{chan}} \end{cases}$$

and $P_r^k = n_r^k / (N_{\text{node}} - k + 1)$. Assuming that P_r^k and P_t^k are independent, it is possible to represent the evolution of the multi-slot state along its journey on the ring by means of a periodic discrete-time Markov chain whose states space is given by triples of integers (n_t, n_r, m) , where m represents the last visited node by the multi-slot along ring i .

Note that the equilibrium distribution of packets transmitted in the multi-slot during its journey on ring i must be equal to the distribution of packets that were on the multi-slot just before starting its journey on the ring.

The average number of packets transmitted by nodes of ring j , normalized to the number of wavelengths N_{chan} , is the maximum throughput achievable by the DAVID Metro network under uniform traffic.

Table I shows results obtained with this model. Note that with $N_{\text{node}} = 10$ and $N_{\text{chan}} = 4$ the limit throughput is around 0.95, which is in good agreement with Fig. 5.

Table I. Maximum throughput under uniform traffic

N_{node}	N_{chan}	max throughput
10	1	1.000000
10	2	.9891437
10	3	.9742157
10	4	.9536157
10	5	.9258112
10	7	.8448739
10	10	.6513216
20	2	.9998781
20	4	.9904285
20	8	.9636848
20	10	.9386381
20	20	.6415141