

Access Control Protocols for Interconnected WDM Rings in the DAVID Metro Network

A. Bianco¹, G. Galante¹, E. Leonardi¹, F. Neri¹, M. Rundo¹

¹ Dipartimento di Elettronica, Politecnico di Torino,
Corso Duca degli Abruzzi, 24 – 10129 Torino – Italy
{bianco, galante, leonardi, neri}@polito.it

Abstract. DAVID (Data And Voice Integration over D-WDM) is a research project sponsored by the European Community aimed at the design of an optical packet-switched network for the transport of IP traffic. The network has a two level hierarchical structure, with a backbone of optical packet routers interconnected in a mesh, and metropolitan areas served by sets of optical rings connected to the backbone through devices called Hubs. The paper focuses on nodes and Hubs architecture, and on the operations of the media access protocol to be used in the DAVID metropolitan area network. A simple access protocol for datagram (not-guaranteed) traffic is defined and its performance are examined by simulation.

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 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 proposed to guarantee throughput fairness among nodes on each ring.

The remainder of the paper is organized as follows. In Section 2 we give an overview of the DAVID network architecture. In Section 3 we focus on the metropolitan network describing both the node and the Hub architecture. In Section 4 the MAC protocol and the scheduling algorithm at the Hub are described. In Section 5 we present some preliminary simulation results to assess the performance of the proposed scheme. We conclude the paper in Section 6, where we describe future research directions.

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 an optical packet router 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).

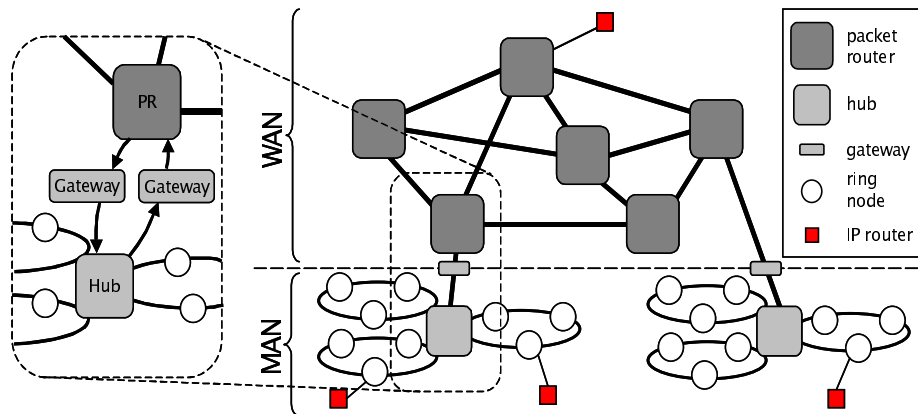


Fig. 1. General overview of the DAVID network

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 carrying an optical packet which consists of a header and a payload.

In packet switched networks 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 provide buffering in the optical domain by means of fibre 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 sharing of rings resources among nodes must be regulated by a properly designed MAC protocol.

The interfaces between WAN and Metro network are critical points where contentions involving traffic flowing between the backbone and the Metro networks might arise. This is worsened by the fact that optical packets could need either format or bit-rate translation (or, eventually, both) while travelling up and down the network hierarchy. Therefore, buffering and translation functions are implemented in Gateways placed between optical packet routers and Hubs (see Fig.1).

In DAVID, the Hub has connection points to the Metro rings and towards a WAN Optical Packet Router through a Gateway. A certain number of Hub ports (wavelengths) are devoted to connections towards the Gateway. The remaining Hub ports connect the Hub to the optical packet rings of the MAN. Since the Hub is buffer-less, as described later, it performs space switching and wavelength conversion only. Optical/electrical memories are present in the Gateway to solve contentions in the time domain for optical packets going from Metro network to WAN and vice versa. Moreover, the Gateway will participate in the MAC protocol, such that, from a logical point of view, the connections from and to the Gateway appear to the Hub as additional Metro ring connections.

We will focus on the DAVID Metro network in the remainder of the paper.

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 fibre, a fixed number of wavelengths is available by WDM partitioning. Logical rings can either be physically disjoint (i.e., run on different fibres), or be obtained by partitioning the optical bandwidth of one fibre into disjoint portions. Nodes belonging to the same logical ring access the same set of shared resources. Recall that one logical ring may represent the WAN/MAN gateway functionality. 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 dic-

tated 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 (fibre), 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 could have 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 ($N_{\text{chan}}=4$ wavelengths) is available on any 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 alignment among different Metro rings is dealt with 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; thus, 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 time-slot basis: all the wavelengths in the 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 a ring node to reach any node 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.

It is important to observe that contentions may arise also at the Hub; contentions 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.

The board of a ring node is basically composed of two parts: an optical part and an electronic one. For the optical part, the ring node can drop, add and erase any packet on any wavelength at each time slot; switching is forbidden for in-transit traffic, in the sense that no operation is allowed on data not addressed to the node. Data are taken off the ring when they arrive at their destination.

The electronic part is composed of the following portions: a segmentation (reassembly) stage to create fixed size data units from variable size packets (viceversa), a queuing stage, in which packets are grouped and stored per destination ring to avoid HoL (Head of the Line) blocking [1], and a load balancing stage, to distribute the packets evenly over the available wavelengths. The HoL blocking is typical of FIFO queues: a packet at the head of the FIFO queue that cannot be transmitted to avoid collisions or contentions on the ring may prevent a successful transmission of another packet following in the FIFO order. Note that this queue architecture is very similar to the VOQ (Virtual Output Queue) architecture used in IQ (Input Queued) switches [2], 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 communication, 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.

The electronic interface is used also to solve the contention problem by running the MAC protocol and to drive the packet insertion on the ring in a free slot.

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 $N_{\text{ring}}=4$ 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 the identity of the ring to which packets transmitted in the multi-slot will be forwarded by the Hub.

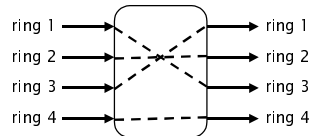


Fig. 2. A ring-to-ring permutation at the Hub

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 re-configured 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 fibre).

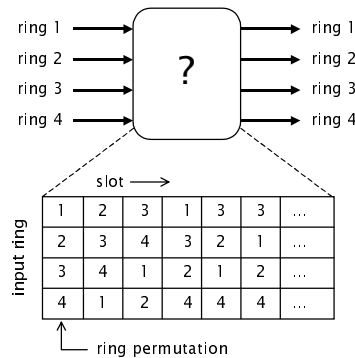


Fig. 3. Scheduling at the Hub

The computation of the sequence of permutations operated by the Hub is a scheduling problem [3, 4], 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 Section 4.4.

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

slot, the upper slot is the control slot where the multi-slot 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 traversal of a ring can be viewed as a downstream path for transmissions done 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 simulation experiments, and force all traffic to pass through the Hub before being removed from the ring.

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 very 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.

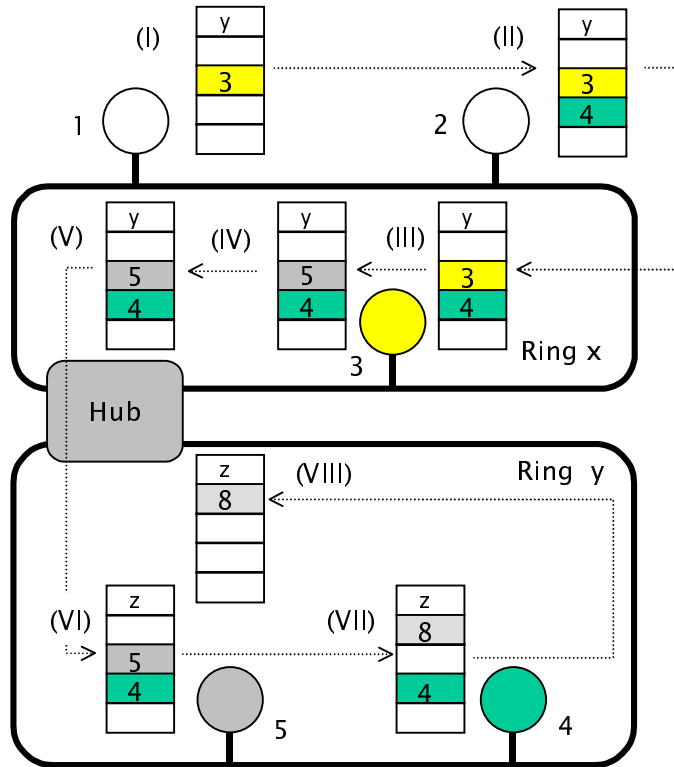


Fig. 4. Multi-slot forwarding in the MAN. Number in slots represent packet destinations

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 signal has to pass through all nodes taking a round trip time for the collection of reservations, makes token based solutions more advantageous for this environment.

The status 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 status field also 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. Again we assume at each node a delay in processing multi-slots larger than the tuning time required to set 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 only if the slot in some wavelength 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 [5] previously studied in the context of single ring can enforce throughput fairness. MetaRing [6] 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 are met. Thus, SAT are delayed by nodes suffering throughput limitations, and SAT rotation times increase 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 $\mathbf{SAT}_{i,j}$ do the job; $\mathbf{SAT}_{i,j}$ 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 $\mathbf{SAT}_{i,j} = \text{TRUE}$), the SAT is loaded in the control slot of the multi-slot, by setting a suitable bit, and by setting $\mathbf{SAT}_{i,j}$ to FALSE.

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.

SAT 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 transmission 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 $\mathbf{R}_{i,o}$ in \mathbf{R} 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 in Section 4.4.1.

According to combinatorial theory [7], \mathbf{R} can be scheduled in at most F time slots, where the frame length F is equal to:

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

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 in F is then repeated an integer number of times, until a new value for \mathbf{R} becomes available and a new matrix decomposition can be computed. Traffic flows from ring i to ring o are served with a rate proportional to $\mathbf{R}_{i,o}/F$.

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. Thus, the sum of each row and column in \mathbf{P} is either equal to 0 or 1. Each switching matrix represents the Hub switching configuration in a given time slot; recall that we need to obtain a set of F ring permutations as the outcome of the Hub scheduling algorithm. Thus, 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 matrixes, i.e. matrixes \mathbf{P} such that

$$\sum_i \mathbf{P}_{i,o} = 1, \quad \forall o \quad \sum_o \mathbf{P}_{i,o} = 1, \quad \forall i$$

The outcome of the scheduling algorithm is a sequence of F doubly stochastic switching matrices; this scheduling satisfies a 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. [8].

The matrix \mathbf{R} may be associated with a bipartite graph G having $2 N_{\text{ring}}$ nodes. Each node represents either one input or one output of the switch, and input node i is connected to output node j 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, and a matching is said to be *maximum* when it has maximum size.

We may apply a maximum size algorithm [9] on \mathbf{R}^F to obtain the Hub scheduling, i.e., a sequence of doubly stochastic $\mathbf{P}(i)$, $i \in \{1, 2, \dots, F\}$.

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. The request matrix \mathbf{R} is decomposed into F switching matrices through iterated application of the critical maximum matching algorithm [4]. A *critical maximum matching* is a maximum matching which covers all the critical input and output nodes.

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 to uniformly distribute ring to ring pairs on the F time slots of the frame, to reduce traffic burstiness.

4.4.1 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 sum of 3 contributions $N_{\text{ring}} \times N_{\text{ring}}$ matrices):

$$\mathbf{R} = \lceil \mathbf{SM} + \beta \mathbf{IC} + \gamma \mathbf{EC} \rceil$$

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, where 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, where 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 $\mathbf{M}_{i,o}$ 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 connections 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 exceed a given threshold, i.e. it contains more than $L_{\text{thr}} \geq Q$ 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 signal 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 simulation experiments) to the MetaRing quota Q ; the rationale is that the quota represents, for each node, transmission opportunities toward a given ring between two consecutive node SAT reception. We assume congestion if the number of packets already in the queue when releasing the SAT is greater than the MetaRing quota, since the node will not be able to transmit all the packets in the queue in the following SAT rotation time.

Note that 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.

The presented algorithm has some important properties that we want to highlight. 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 exist 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 Section 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 each multi-slot on ring x labelled with destination 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 comprise $N_{\text{ring}}=4$ rings, with 10 nodes on each ring. For each ring-to-ring communication $N_{\text{chan}}=4$ data channels are available; thus, each multi-slot comprise 5 slots, 4 for data traffic and 1 for control and management. Each nodes store 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 values used in our simulation experiments for the parameters defined in the measurement algorithm are the following: $\beta=1$, $\gamma=3$, $\alpha=0.9$. The ring round trip time is assumed 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.

We consider two traffic patterns: a uniform traffic pattern and an unbalanced traffic pattern. Define the weight matrix \mathbf{W} , of size $N_{\text{ring}} \times N_{\text{ring}}$, where the value assumed by 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,

$$\sum_i \mathbf{W}_{i,o} \leq 1, \quad \forall o \qquad \sum_o \mathbf{W}_{i,o} \leq 1, \quad \forall i$$

In the uniform traffic pattern $\mathbf{W}_{i,o}=1/N_{\text{ring}} \quad \forall i,o$. 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.

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 ring symmetries. Nodes on the same ring do not exhibit throughput unfairness thanks to the MetaRing algorithm.

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 offered load under uniform traffic. Each destination ring is treated fairly and the total network utilisation is close to 0.95. Note that we significantly overload the network, since ρ ranges from 0.1 to 3, but the algorithm behaves well even under this extreme condition.

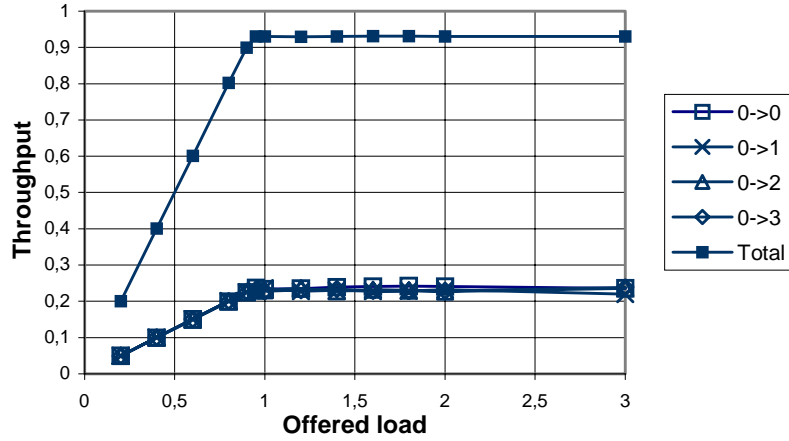


Fig. 5. Throughput under uniform traffic

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 with a combinatorial analysis.

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 $OW \times \rho$, once this value is reached, the frame length changes slowly following traffic fluctuations. The convergence speed is determined by the value of the 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 like fairness criteria [10]; 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.

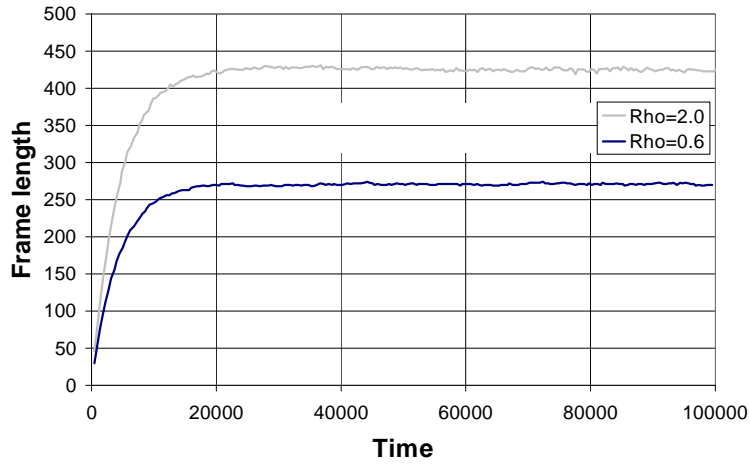


Fig. 6. Frame length as a function of time under uniform traffic

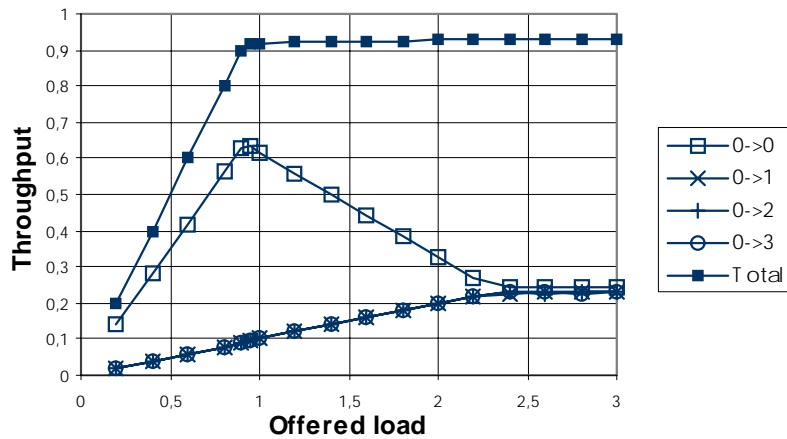


Fig. 7. Throughput under unbalanced traffic

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 interval 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. Then, the allocation is 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 heavily loaded conditions.

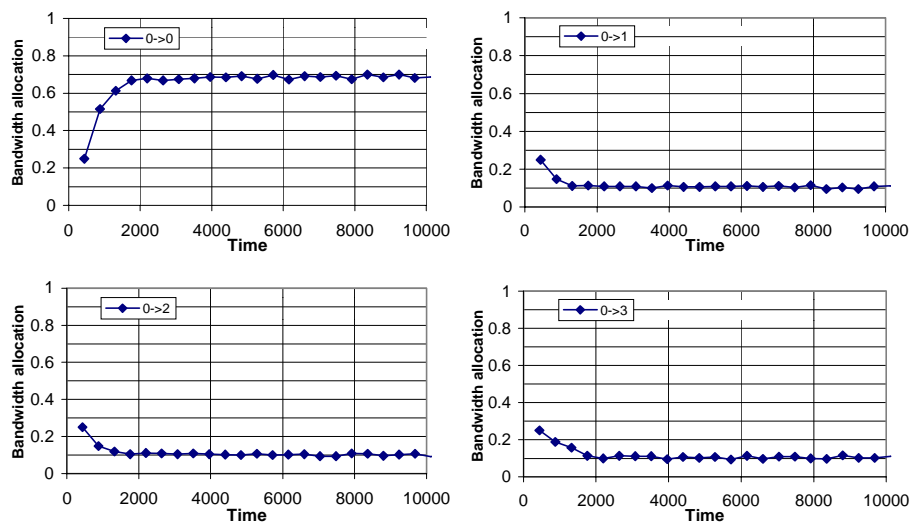


Fig. 8. Bandwidth allocation for unbalanced traffic with $\rho=0.6$

Finally, in Fig. 9 we show the queue occupancy (in packets) in overload ($\rho=2.0$) for a given node on ring 0 (all other nodes show similar queue length behaviours), sampled every 100 slot times. Whereas the queue length for intra-ring traffic saturates since this connection is overloaded, all other queues show oscillating behaviours, since each inter-ring connection becomes congested only when the scheduling does not allocate enough slots to this connection. 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 ring propagation time, the time constant under which any bandwidth control cannot be achieved in this network.

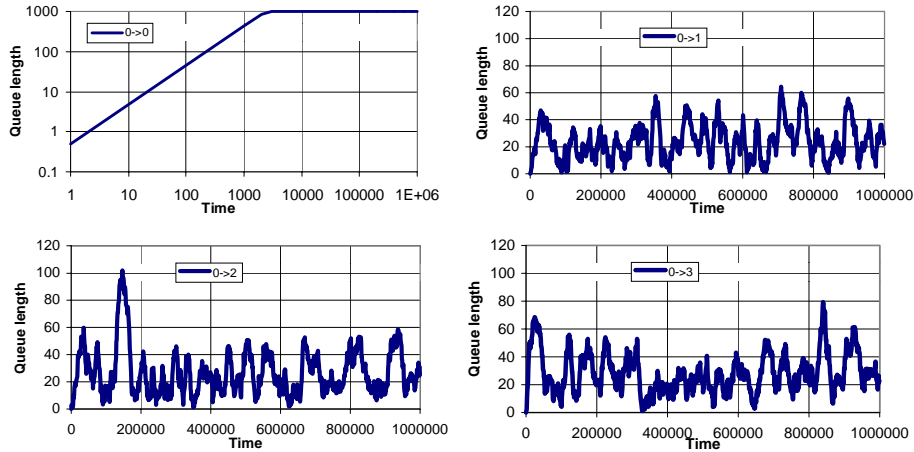


Fig. 9. Queue length for unbalanced traffic with $\rho=2.0$

6 Conclusions and Future Work

Although the presented simulation results are encouraging and the algorithm shows good performance, several issues remain to be addressed.

First, other traffic scenarios should be studied to prove the algorithm's 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.

Then, an accurate analysis of the effect of the parameter setting must be provided, to obtain a set of values that provides good performance under different conditions. Transient behaviours must be 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 for Internet.

References

1. M. Karol, M. Hluchyj, S. Morgan, "Input Versus Output Queuing on a Space Division Switch", IEEE Transactions on Communications, Vol.35, No.12, December 1987, pp.1347-1356
2. N. McKeown, A. Mekkittikul, V. Anantharam, J. Walrand, "Achieving 100% throughput in an input-queued switch", IEEE/ACM Transactions on Communications, Vol. 47, No. 8, August 1999

3. T. Inukai, "An efficient SS/TDMA time slot assignment algorithm", IEEE Transactions on Communications, Vol. 27, pp. 1449-1455, October 1979
4. B. Hajek, T. Weller, "Scheduling Non-Uniform Traffic in a Packet-Switching System with Small Propagation Delay", IEEE/ACM Transactions on Networking, Vol.5, No.6, December 1997, pp. 813-823
5. M. Ajmone Marsan, A. Bianco, E. Leonardi, A. Morabito, F. Neri, "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, pp.58-66, February 1999
6. I. Cidon, Y. Ofek, "MetaRing - a Full-Duplex Ring with Fairness and Spatial Reuse", IEEE Transactions on Communications, Vol.41, No.1, January 1993, pp.110-120
7. M. Hall, Jr., Combinatorial Theory, Waltham, MA, Blaisdell, 1969
8. C.S. Chang, W.J. Chen, H.Y. Huang, "Birkhoff-von Neumann Input Buffered Crossbar Switches", IEEE Conference on Computer Communications (INFOCOM 2000), Tel Aviv, Israel, pp. 1614-1623, March 2000
9. R.E. Tarjan, *Data Structures and Network Algorithms*, Society for Industrial and Applied Mathematics, Pennsylvania, November 1983
10. D. Bertsekas, R. Gallager, *Data networks*, Prentice-Hall, 1987