Available at www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

ELSEVIER

Computer Networks 41 (2003) 143–160

COMPUTER NETWORKS

www.elsevier.com/locate/comnet

A simulation study of optical burst switching and access protocols for WDM ring networks

Lisong Xu, Harry G. Perros *, George N. Rouskas

Department of Computer Science, College of Engineering, North Carolina State University, P.O. Box 8206, Raleigh, NC 27695-7534, USA

Received 14 January 2002; received in revised form 21 June 2002; accepted 16 September 2002

Responsible Editor: A. Fumagalli

Abstract

We consider a wavelength division multiplexing metro ring architecture with optical burst switching. The ring consists of N nodes, and each node owns a home wavelength on which it transmits its bursts. The ring operates under the fixed transmitter tunable receiver scheme. Control information is transmitted on a separate control channel. Five different burst switching access protocols are proposed, and their performance and fairness is evaluated by simulation.

Keywords: Optical burst switching; Ring; MAN; Access protocols

1. Introduction

Wavelength division multiplexing (WDM) appears to be the solution of choice for providing a faster networking infrastructure that can meet the explosive growth of the data traffic. Because of the bursty nature of the data traffic, wavelength-routed optical networks [3] which employ circuit switching may not be the most appropriate for the emerging optical Internet. Optical packet switching [9,11] is an alternative technology that appears to be the optimum choice. However, at this moment the technology is not mature enough to provide a viable solution. Optical burst switching (OBS) [4] is a switching technique that occupies the middle of the spectrum between the well-known circuit switching and packet switching paradigms, borrowing ideas from both to deliver a completely new functionality. The unit of transmission is a *burst*, which may consist of several IP packets, ATM cells, frame relay frames, or other types of data, such as HDTV traffic, and sensor traffic. The transmission of each burst is preceded by the transmission of a *burst header packet*, which usually takes place on a separate signaling channel. Unlike circuit switching, a source node does not wait for confirmation that a path with available resources has been setup; instead, it starts transmitting the data burst soon after the transmission

* Corresponding author. Tel.: +1-919-515-2858; fax: +1-919-515-7925.

E-mail addresses: lxu2@csc.ncsu.edu (L. Xu), hp@csc.ncsu.edu (H.G. Perros), rouskas@csc.ncsu.edu (G.N. Rouskas).

1389-1286/03/$ - see front matter
PII: S1389-1286(02)00371-7

of the burst header packet. We will refer to the interval of time between the transmission by the source node of the first bit of the burst header packet and the transmission of the first bit of the data burst as the *offset*. The burst header packet carries information about the burst, including the offset value, the length of the burst, its priority, etc. The purpose of the burst header packet is to inform each intermediate node of the upcoming data burst, so that it can configure its switch fabric appropriately so that to switch the burst to the appropriate output port. However, in case of congestion or output port conflicts, an intermediate node may drop a burst. Also, consecutive bursts between a given source-destination pair may be routed independently of each other.

There are several variants of burst switching, mainly differing on the length of the offset. In the burst switching scheme called Tell And Go (TAG) [7], the burst is transmitted immediately after the burst header packet. That is, the offset is only the transmission time of the burst header packet. This scheme is practical only when the switch configuration time and the switch processing time of a burst header packet are very short. At the other extreme, the Tell and Wait (TAW) [7] scheme requires the offset to be at least equal to the time required to receive an acknowledgement from the destination. TAW is equivalent to circuit switching in that it incurs a round-trip delay to setup the transmission, and since the burst header packet reserves resources, delivery of the burst is guaranteed. Another advantage of TAW is that it eliminates receiver collisions, since a node returns an acknowledgment only for bursts it is prepared to accept.

An intermediate burst switching scheme, known as Just Enough Time (JET) [4], selects the offset in a manner that takes into account the processing delays of the burst header packet at the intermediate switches. Let $T_{\mathrm{i}}^{(\mathrm{p})}$ denote the processing delay of a burst header packet at an intermediate switch, $T_{\mathrm{d}}^{(\mathrm{p})}$ denote the processing delay of a burst header packet at the destination switch, and $T_{\mathrm{d}}^{(\mathrm{s})}$ denote the time to setup (configure) the destination switch. Then, the offset value for JET is

$$\mathrm{offset}_{\mathrm{JET}} = \left(\sum_i T_{\mathrm{i}}^{(\mathrm{p})} \right) + T_{\mathrm{d}}^{(\mathrm{p})} + T_{\mathrm{d}}^{(\mathrm{s})}. \tag{1}$$

The offset calculation for the JET protocol is illustrated in Fig. 1 for a path that includes two

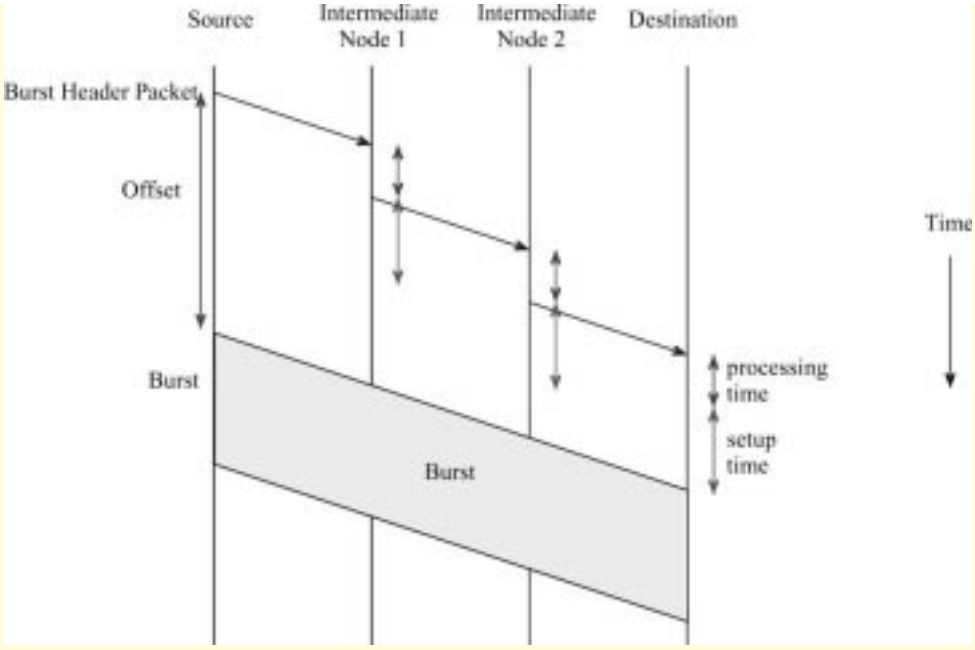

Fig. 1. Offset calculation in the JET OBS protocol.

intermediate switching nodes between the source and destination of the burst. As can be seen, the offset needs to be long enough to account for the processing time of the burst header packet at the two intermediate nodes and the destination plus the switch setup time at the destination. If the offset time is less than that, then there is a possibility that the burst may arrive at a node before the node is ready to switch the burst.

One issue that arises in computing the offset under JET is determining the number of intermediate switching nodes (hops) between the source and destination. In OBS networks, information about the number of hops in a path may not, in general, be readily available; even when such information is somehow known, because of the effects of routing changes, it is not guaranteed to be valid when used. Thus, it is desirable to use an offset value that does not depend on the path used and does not require the exchange of information among network nodes.

As we can see from expression (1), the part of the offset value that depends on the path between the source and destination is the sum of the processing times at intermediate nodes. Given the recent advances in hardware implementation of communication protocols, it is reasonable to assume that the processing time $T_i^{(p)}$ in (1) will be very short for most common functions of the signaling protocol (i.e., no exception conditions). In this case, fiber delay lines of reasonable length may be used at intermediate nodes to delay each incoming burst by an amount of time equal to $T_i^{(p)}$. Given such fiber delays, the first term in the right hand side of (1) can be omitted when computing the offset. We call this new scheme the Only Destination Delay (ODD) protocol, and its offset is given by

$$\text{offset}_{\text{ODD}} = T_d^{(p)} + T_d^{(s)}. \quad (2)$$

Furthermore, instead of using destination-specific values for the processing and switching delays in (2), one may use a constant offset value by taking the maximum of these values over all destinations. A constant offset that does not depend on the path (number of hops) to the destination significantly simplifies the design and implementation of signaling protocols and optical switches for burst switching networks.

OBS has been studied in the context of wide area networks with a mesh topology [4–6,8]. In this paper we study burst switching access protocols for WDM ring networks. Our focus on ring topologies is motivated by the wide deployment of SONET/SDH rings. These networks represent a significant investment on the part of carriers, and are currently being upgraded to support WDM. To the best of our knowledge, this is the first study of burst switching protocols specifically for ring networks. Our vision of the OBS ring is that it will be used to transport different types of traffic, such as IP, ATM, and frame relay traffic, and also HDTV and sensor traffic that may not be transported over IP, ATM or frame relay. The objective of this paper is to analyze the performance and fairness of five different OBS access protocols. How these protocols can be used to provide different classes of services to different applications is beyond the scope of this paper.

This paper is organized as follows. Section 2 describes the ring network we consider and the basic operation of burst switching in such an environment. Section 3 provides a detailed description of the various burst switching access protocols studied in this paper. Section 4 presents the simulation results on the performance of these burst switching access protocols, and finally Section 5 provides some concluding remarks.

2. The network under study

2.1. Ring and node architecture

We consider N OBS nodes organized in a unidirectional ring, as shown in Fig. 2. The ring can be a metropolitan area network (MAN) serving as the backbone that interconnects a number of access networks, and transporting multiple types of traffic from users, such as IP traffic, ATM traffic, frame relay traffic, HDTV traffic, and sensor traffic. Each fiber link between two consecutive OBS nodes in the ring can support $N+1$ wavelengths. Of these, N wavelengths are used to

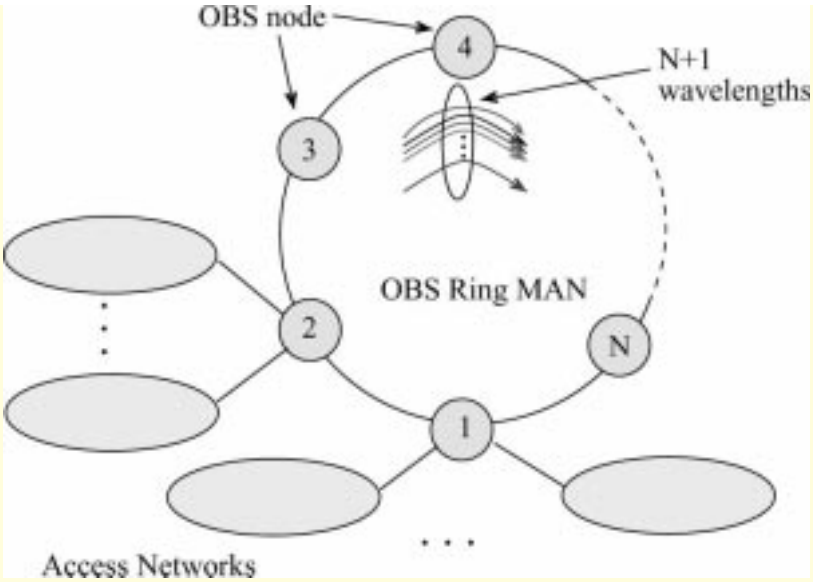

Fig. 2. OBS ring MAN.

transmit bursts, and the $(N+1)$th wavelength is used as the control channel.

Each OBS node is attached to one or more access networks. In the direction from the access networks to the ring, the OBS node acts as a concentrator. It collects and buffers electronically data, transmitted by users over the access networks, which need to be transported over the ring. Buffered data are subsequently grouped together and transmitted in a burst to the destination OBS node. A burst can be of any size between a minimum and maximum value. Bursts travel as optical signals along the ring, without undergoing any electro-optic conversion at intermediate nodes. In the other direction from the ring to the access networks, an OBS node terminates optical bursts destined to itself, electronically processes the data contained therein, and delivers them to users in its attached access networks.

The architecture of an OBS node is shown in Fig. 3. Each node is equipped with one optical add–drop multiplexer (OADM), and two pairs of optical transceivers. The first pair consists of a receiver and transmitter fixed tuned to the control wavelength, and are part of the control module in Fig. 3. The control wavelength is dropped by the OADM at each node, and added back after the control module has read the control information and (possibly) has inserted new information (the following subsection provides more details on the operation of the control wavelength).

The second pair of transceivers consists of a transmitter that is fixed tuned to the node's *home wavelength*, and an agile receiver (or a receiver array) that can receive from all N wavelengths that transmit bursts. Each OBS node has a dedicated home wavelength on which it transmits its bursts. The OADM at each node removes the optical signal from the node's home wavelength by dropping the corresponding wavelength, as Fig. 3 illustrates. The OADM also drops the optical signal on other burst wavelengths, whenever they contain bursts for this node. In the case where multiple bursts arrive, each on a different wavelength, at an OBS node, the receive module in Fig. 3 employs a collision resolution strategy to determine which burst will be accepted.

To support ODD, an extra fiber delay line (not shown in Fig. 3) is added into the node to delay outgoing bursts on all wavelengths except the control wavelength and the node's home wavelength.

Data waiting for transmission is organized into (logical) transmit queues according to their destination. The data buffer at each OBS node is shared by $N-1$ transmit queues, each corresponding to

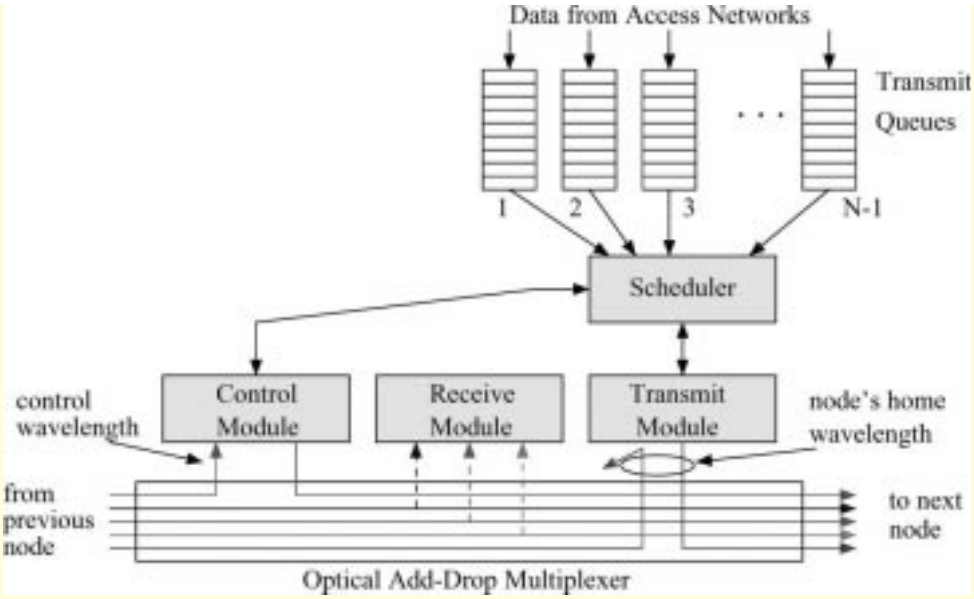

Fig. 3. OBS node architecture (delay lines are not shown).

one of the $N-1$ destination nodes. The order in which transmit queues are served is determined by the scheduler module in Fig. 3. In this paper, the transmit queues are served in a round-robin manner.

2.2. Control wavelength operation

The control wavelength is used for the transmission of control slots. In a ring with N nodes, N control slots, one for each node, are grouped together in a *control frame* which continuously circulates around the ring. Depending on the length of the circumference of the ring, there may be several control frames circulating simultaneously. In this case, control frames are transmitted back-to-back on the control wavelength.

Each node is the owner of one control slot in each control frame. Each control slot contains several fields, as Fig. 4 illustrates. The format and type of the fields depend on the OBS protocol used (for more details, refer to the description of the protocols in Section 3). In general, however, each control slot includes fields for the destination address, the offset, and the burst size. Other fields, such as a token field, may be included for some of the protocols, as necessary.

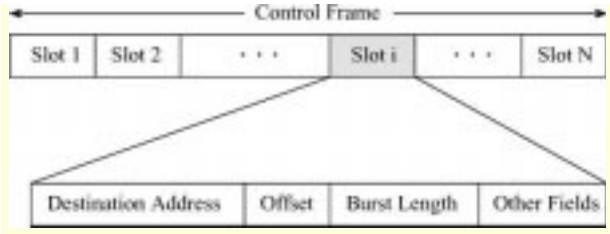

Fig. 4. Structure of a control frame.

When acting as a source, a node waits for the next control frame and writes the burst information (destination address, burst length, and, if applicable, the offset) in its own control slot. If it has nothing to transmit, it just clears all the fields in its control slot. At each node, the entire control frame is read first to determine whether any control slots indicate a burst transmission to this node. If so, and assuming that the node is not in the process of receiving another burst, it instructs its tunable receiver to tune to the appropriate wavelength to receive the burst; in other words, preemption is not allowed. In case of a receiver collision (i.e., when the address of this node is specified in multiple control slots, which may give rise to overlapping transmissions), the destination node selects one of the bursts to receive. In an acknowledgment-based protocol, the node also

modifies the appropriate field as an indication to the source node to transmit its burst.

We note that each node in the ring acts as a source node (inserting bursts in its home wavelength), as an intermediate node (passing through bursts traveling to downstream nodes), and as a destination node (terminating bursts sent to it). As a result, each node must read each control frame in its entirety before determining what action to take (i.e., whether to write in its own control slot to indicate its intention to transmit a burst, and/or whether to acknowledge the request of a burst transmission). Therefore, in a ring network the time to process a control frame is the same for intermediate and destination nodes (i.e., $T_{\mathrm{i}}^{(\mathrm{p})} = T_{\mathrm{d}}^{(\mathrm{p})}$). The control frame is delayed by this amount of time as it passes through each node. This delay is the sum of the control frame transmission time plus the time to process the control frame, and it can be kept short by employing a simple protocol implemented in hardware. A number of OBS protocols having these features are described in the next section.

3. OBS protocols

Since each OBS node is assigned a unique home wavelength, bursts may be lost due to receiver collisions. This occurs when two or more source nodes transmit (each on its home wavelength) bursts to the same destination node, and the burst transmissions overlap in time. In this paper, we proposed a number of different access protocols that differ mainly in the way that receiver conflicts are resolved. These protocols can be classified in the following three classes, depending on who is responsible to resolve receiver collisions.

- *Source node:* In this class of protocols, a source node resolves receiver collisions using the information transmitted on the control wavelength.
- *Destination node:* In this class of protocols, a source node must get permission from the destination node, before it can send its burst. The destination node schedules all incoming requests so that to avoid collisions.
- *Other:* In this class of protocols, neither the source node nor the destination node are responsible for receiver collision resolution. For example, a common method in ring networks is to use tokens to resolve receiver collisions.

Our emphasis is on protocols that use few rules, are simple to implement in hardware (i.e., they can operate at wire speeds) and are distributed in nature (i.e., each node locally executes an identical copy of the protocol and makes transmit decisions by its local knowledge). We have deliberately avoided protocols that are centralized in nature, or they require the collection of transmit queue sizes, or they require network-wide synchronization (e.g., TDM-based schemes).

In this paper, we propose five access protocols, namely round-robin with random selection (RR/R), round-robin with persistent service (RR/P), round-robin with non-persistent service (RR/NP), round-robin with acknowledgement (RR/ACK), and round-robin with tokens (RR/Token). The first three protocols belong to the "source node" class, RR/ACK belongs to the "destination node" class, and the last one belongs to "other" class. Before we proceed with the description of the five protocols, a short discussion on the assumptions we make is necessary. We define a burst as an encapsulation of IP packets, ATM cells, frame relay frames, or some other types of packets containing data. A burst format is needed so that the destination node can correctly extract the data from the received burst. The format of a burst is outside the scope of this work. While any burst format incurs overheads that affect performance measures such as throughput and delay, the various protocols are affected in the same degree. Since we are interested in the *relative* performance of the five protocols, we have ignored this burst format overhead.

A transmit queue is *eligible* for service if its size is larger than `MinBurstSize`, or the first data of the transmit queue has waited for more than `TimeOut` time. If the size of the eligible transmit queue is less than `MaxBurstSize`, then a burst that includes all data in the transmit queue is constructed. Otherwise, a burst of at most size `MaxBurstSize` is constructed, and the data re-

maining in the transmit queue is served at a later time.

In the following five subsections, we describe the proposed OBS access protocols. Numerical results are given in Section 4.

3.1. Round-robin with random selection

The first protocol we consider uses a round-robin scheduler at each node to serve the transmit queues, and lets each receiver randomly select a burst from the bursts that arrive simultaneously. Thus, we call this protocol RR/R. More specifically, the operation of the protocol at node i is as follows:

- At the transmitting side, the scheduler of node i visits all eligible transmit queues in a round-robin fashion. Suppose that, at time t_1, transmit queue j is selected for service, then node i waits for the first control frame that arrives after time t_1. When the frame arrives, node i writes the burst information and destination address j in its own control slot (i.e., the ith slot of the control frame). After a delay equal to the offset value, node i transmits the burst on its home wavelength.
- At the receiving side, when a control frame arrives at node i, it scans the control slots of the control frame, checking for any slot that has i in the destination address field. If more than one such slots are found, node i randomly selects one of them, say k (since all the corresponding bursts will arrive at node i at the same time for both JET and ODD, and at most one of them can be accepted). In this case, all bursts to node i except the burst from node k will be lost. Node i then checks whether its receiver is free at the time when the burst from node k arrives at node i, and checks whether its receiver has enough time to tune to another wavelength. If so, it instructs its receiver to tune to node k's home wavelength in order to receive the burst transmission. Otherwise, it gives up on the burst from node k.

Because of the randomness involved in resolving receiver conflicts, RR/R is a fair protocol. However, burst loss may occur due to these conflicts.

3.2. Round-robin with persistent service

The RR/P protocol is similar to the RR/R protocol, but it is designed to eliminate receiver conflicts that can be detected prior to the transmission of a burst. The operation of this protocol at node i is as follows:

- At the transmitting side, node i maintains a variable `EarliestFreeTime(j)` for each destination node j, which specifies the earliest time at which the receiver of node j would be free. This variable is updated by monitoring the burst information in control slots that have j in the destination address field.

  The scheduler at node i visits all eligible transmit queues in a round-robin fashion. Suppose that, at time t_1, transmit queue j is selected for service, then node i waits for the first control frame that arrives after time t_1. Suppose it arrives at time t_2, then node i updates the variable `EarliestFreeTime(j)` based on relevant information (if any) in the control frame. Node i also computes the time t_3 that the first bit of its burst would arrive at node j. t_3 is calculated as follows:

$$t_3 = t_2 + T_{\mathrm{i}}^{(\mathrm{p})} + \mathrm{offset} + \delta_{ij} \tag{3}$$

  where δ_{ij} is the burst propagation delay from node i to node j. If `EarliestFreeTime(j)` plus the receiver tuning time at node j is less than t_3, then node i writes its burst information in its own control slot, and sends the burst after a delay equal to the offset. If, on the other hand, `EarliestFreeTime(j)` plus the receiver tuning time at node j is greater than t_3, then node i knows that sending its burst will result in a receiver conflict. In this case, node i does not transmit the burst; instead it waits for the next control frame and repeats the process of transmitting the burst to node j. This is the *persistent* feature of the protocol, in that the round-robin scheduler does not proceed to serve the next transmit queue until the burst to node j has been sent.

We note that deferring the transmission of a burst based on a calculation of the earliest free time for receiver j does not altogether eliminate receiver collisions. Suppose that two nodes simultaneously determine (based on information they read in *different* control frames) that it is safe to send a burst to some destination j. This simultaneous transmission may result in a receiver conflict, which neither of the nodes is able to predict. When the downstream node later receives the control frame with the upstream nodes burst information, it will detect the conflict. Despite this fact, the downstream node proceeds with its burst transmission, and its scheduler also proceeds to serve the next eligible transmit queue after queue j.

- At the receiving side, the operation of the protocol is identical to RR/R.

RR/P does eliminate some receiver collisions, but it does not completely eliminate receiver collisions.

3.3. Round-robin with non-persistent service

The operation of the RR/NP protocol is identical to the operation of the RR/P protocol with one exception. Suppose that at time t_1 node i has selected transmit queue j for service using the RR scheduler. Suppose also that once the first control frame arrives after time t_1, the node determines that transmitting a burst to j would result in a collision. The node refrains from transmitting the burst, but, instead of continuing its attempt to serve transmit queue j (the persistent feature of RR/P), its scheduler proceeds to serve the next eligible transmit queue upon arrival of the next control frame.

The RR/NP protocol may result in lower delay than RR/P. However, since a node gives up its burst transmission whenever it determines that it will lead to a collision, RR/NP may lead to the starvation of certain transmit queues, and thus, it has fairness problems. Specifically, a node's priority to transmit to a given destination depends on the relative location in the ring. Node i has the highest (lowest) priority to transmit bursts to node $(i \ominus 1)$ (respectively, node $(i \oplus 1)$), where $\ominus$ and $\oplus$ denote subtraction and addition, respectively, modulo N.

As in RR/P, RR/NP does not completely eliminate receiver collisions.

3.4. Round-robin with tokens

This protocol uses tokens to resolve receiver collisions at the receivers. Different from traditional token-based protocols, such as the IBM token ring and FDDI, which are single token access protocols, this protocol uses multiple tokens (Cai et al. [1] proposed a multiple token access protocol for a different WDM ring architecture). There are N tokens, one for each destination node. A token may be either available or in use. The status of token j is indicated in a binary field (located in the "other fields") of the jth control slot. If it is available, then the binary field is set to one. Otherwise, it is set to zero. If token j is available, then this will be marked in the jth control slot of only one control frame. In the remaining control frames, this binary field will be set to zero. A node can only transmit to a destination node j, if it captures the jth token. The transmit queues at each node are served in a round-robin manner. Thus, we call this protocol RR/Token.

The operation of the protocol at node i is as follows:

- At the transmitter side, node i monitors each received control frame. If it finds an available token, node i removes it from the control frame, and puts it in its FIFO token queue. Node i also serves the transmit queues in the arrival order of tokens. More specifically, suppose that the first token in the token queue is token j, node i first checks whether transmit queue j is eligible for service. If not, node i releases token j, i.e., it removes it from its token queue, and it places it in the next control frame, and it then proceeds with the next token in the queue. Otherwise, node i constructs the burst to node j, writes the burst information in the next control frame, and sends it after a delay equal to the offset value. Once the burst transmission is complete, node i releases token j to the next control frame. It then proceeds to serve the transmit queue corresponding to the next token in the token queue. Since every node has a FIFO token queue, the

order in which tokens circulate around the ring is fixed. Recall that there are only N tokens, one for each destination node. Therefore, transmit queues are served in a round-robin manner.
- At the receiver side, node i checks each incoming control frame for any control slot indicating a burst transmission to this node. If such a control slot is found, node i instructs its receiver to tune to the appropriate home wavelength for receiving the burst.

Because of the token operation, there will be at most one burst transmission arriving at a destination node at any time. That is, RR/Token is a receiver collision free protocol.

3.5. Round-robin with acknowledgement

The RR/ACK protocol is based on the TAW scheme [7]. A source node i first sends a request (including destination and size) to transmit a burst to the destination node j. When node j receives the request, it calculates an offset value, and sends it back to node i in the offset field of control slot i. We note that a source node is not allowed to have more than one outstanding request; in other words, it is not permitted to send out another request to a different destination node while it is waiting for an acknowledgement. This rule avoids transmitter conflicts, i.e., the situation in which a source node receives acknowledgements from two or more different destinations which may cause overlapping burst transmissions.

- At the transmitter side, node i selects a transmit queue j using round-robin among all `eligible` transmit queues. It then waits for the next incoming control frame, and it writes a request in its own control slot i. The request consists of the destination address (in this case, j), and the burst length. Note that the source node i does not write the offset field in the control slot; the offset value will be provided by the destination node as the acknowledgment. After node i receives the acknowledgment from node j one round-trip time later, it instructs its transmitter to send out the burst at the time specified by node j in the acknowledgement.

Let τ be the round-trip delay of a control frame (i.e., the propagation time around the ring plus the sum of the processing time of a control frame at each node in the ring). Let t be the time (in the future) at which the current burst transmission to node j will complete. In order to improve the utilization of the ring under the RR/ACK scheme, we define the *next safe request point* for node i as $t-\tau$. Node i will wait until time $t-\tau$ before it submits a new request for transmission to another destination k. Thus, when node i receives the acknowledgment from node k at time $(t-\tau)+\tau=t$, the burst transmission to node j will be complete and its transmitter will be free to transmit a burst to node k.
- At the receiver side, node j acknowledges (i.e., fills in the corresponding offset field) each request it receives in a first-come, first-served manner. Specifically, after acknowledging a request from node i, node j computes the time t' at which it will receive the last bit of node i's burst. Node j's receiver is free after time t'. When the next request arrives, say, from node k, node j sends an offset that is computed such that the first bit of node k's burst will arrive at node j after time t' plus the tuning latency of the receiver.

RR/ACK is a receiver collision free protocol.

4. Numerical results

In this section we use simulation to compare the protocols listed in Table 1. For each of the four protocols RR/R, RR/P, RR/NP, and RR/Token, we consider two variants: one in which the offset calculation is based on ODD, using expression (2), and one in which the offset calculation is based on JET, using expression (1). Recall that the main difference between the two offset calculations is that the ODD offset includes only the processing and setup delay at the destination, while the JET offset includes additional terms representing the processing delays at intermediate nodes. As we shall see, the ODD offset calculation results in smaller delay for all four protocols, and higher throughput for the RR/Token protocol, the only receiver collision free protocol. Moreover, ODD

Table 1
OBS protocols used in the simulation

No.	Protocol name	Offset calculation
1	RR/R	ODD
2	RR/P	ODD
3	RR/NP	ODD
4	RR/Token	ODD
5	RR/R	JET
6	RR/P	JET
7	RR/NP	JET
8	RR/Token	JET
9	RR/ACK	TAW

makes it possible to design a delay fair protocol. However, the reader should keep in mind that this performance improvement is achieved at the expense of more complex burst switching nodes, since the latter must implement fiber delay lines to delay incoming bursts for an amount of time equal to the processing delay of a burst header packet. Finally, we also simulate the RR/ACK protocol which is a TAW protocol.

In our simulation study we consider a ring network with 10 nodes, each with an electronic buffer of 10 MB. The distance between two successive nodes in the ring is taken to be 5 km. We assume that the control wavelength runs at 622 Mbps, while each burst wavelength runs at 2.5 Gbps. Each control slot in a control frame is 100 bytes long regardless of the protocol used in the ring. That is, the duration of a control slot is 1.286 μs. The processing time of a control frame at both the intermediate ($T_i^{(p)}$) and destination nodes ($T_d^{(p)}$) is set to be 10 slot times, or 12.86 μs, and the setup time at the destination nodes $T_d^{(s)}$ is 1 μs.

We assume that data arrives in packets, and the packet arrival process to each node is described by a modified interrupted poisson process (IPP) [2]. This modified IPP is an ON/OFF process, where both the ON and the OFF periods are exponentially distributed. Packets arrive back to back during the ON period at the rate of 2.5 Gbps. No packets arrive during the OFF period. The packet size is assumed to follow a truncated exponential distribution with an average size of 500 bytes and a maximum size of 5000 bytes. The last packet in an ON period may be truncated so that its last bit arrives at the end of the ON period. We use the squared coefficient of variation, c^2, of the packet inter-arrival time to measure the burstiness of the arrival process. c^2 is defined as the ratio of the variance of the packet inter-arrival time divided by the squared mean of the packet inter-arrival time. We use the expression for the c^2 of an IPP, where the packet size is not truncated. We have

$$c_{\mathrm{IPP}}^2 = 1 + \frac{2\lambda\mu_1}{(\mu_1+\mu_2)^2} \tag{4}$$

where $1/\lambda = (500 \text{ bytes})/(2.5 \text{ Gbps}) = 1.6$ μs, and $1/\mu_1$ and $1/\mu_2$ are the mean times of the ON and OFF periods, respectively. We have found experimentally that it is very close to the c^2 of the modified IPP used in this simulation. To completely characterize the arrival process, we use the above expression for the c^2 and another equation that involves the mean times of the ON and OFF periods. We define the quantity

$$\text{average arrival rate} = (2.5 \text{ Gbps})\frac{\mu_2}{\mu_1+\mu_2}. \tag{5}$$

Given the c^2 and the average packet arrival rate, we can calculate the quantities μ_1 and μ_2, and therefore the arrival process is completely characterized.

In all the figures given in this section, simulation results are plotted along with 95% confidence intervals estimated by the method of batch mean. The number of batches was set to 30, with each batch run lasting until each node has transmitted at least 10,000 bursts. As the reader will notice, however, most confidence intervals are very narrow and are barely visible in these figures.

In Section 4.1 we present a comparison of the performance of the RR/R, RR/P, RR/NP, and RR/Token protocols with ODD offsets. In Section 4.2 we investigate the impact of the offset calculation JET versus ODD on the performance of the protocols. In Section 4.3, we compare RR/ACK that uses the TAW scheme with RR/Token that uses the ODD offset. In these three sections, the traffic to the ring is symmetric. That is, each node is fed with an arrival process that has the same parameters, and a packet arriving at a node is assigned a destination node following the uniform distribution. In Section 4.4, we study the performance of the access protocols assuming asymmetric traffic.

4.1. Performance of protocols with ODD offset

4.1.1. Effect of average arrival rate

In this section, we investigate the performance of the first four protocols listed in Table 1 for which the calculation of the offset is based on ODD. Specifically, we are interested in five performance measures, namely throughput, loss, delay, fairness, and buffer requirement. These performance measures are estimated by varying the average arrival rate from 0.5 to 2.0 Gbps with an increment of 0.3 Gbps. (The average arrival rate we refer to, is the average arrival rate into a single node). Packets arriving at a node are assigned a destination node following the uniform distribution. c^2 of the packet inter-arrival time at each node is set to 20. We also set `MaxBurstSize` to 112 KB, `MinBurstSize` to 16 KB, and `TimeOut` to 4 ms, which is about 10 times the round-trip delay of the control frame.

Fig. 5 plots the mean node throughput versus the average arrival rate for all four protocols. The mean node throughput is defined as the average number of bits received by all nodes in a unit time divided by the number of nodes. We observe that RR/Token, a receiver collision free protocol, achieves the highest throughput. Among the three protocols in which receiver collisions are possible, RR/P achieves the highest throughput, followed by RR/NP and RR/R.

We distinguish between two types of loss. First, packets arriving to find a full buffer at the source node are dropped. In our simulation experiments, we observed that only RR/Token has a 0.01% packet loss rate (i.e., the number of packets lost divided by the number of all packets arrived) due to buffer overflow, when the average arrival rate is 2.0 Gbps. That means RR/Token requires a larger buffer than the other three protocols.

The second type of loss occurs when a burst is dropped at the destination due to a receiver collision. Fig. 6 plots the burst loss rate due to receiver collisions versus the average arrival rate. The burst loss rate is the total number of lost bursts in all nodes divided by the total number of transmitted bursts on the ring. As a receiver collision free protocol, RR/Token never incurs loss due to receiver collisions. For the other three protocols, RR/P has the least burst loss rate, followed by RR/NP and RR/R.

Next, we give an intuitive explanation of the burst loss plots in Fig. 6. The behavior of these plots is related to the c^2 of the burst size. If all other parameters are kept the same, a larger burst size c^2 leads to a larger burst loss rate due to receiver collisions. Fig. 7 shows the c^2 of the burst sizes as a function of the average arrival rate. We note that the plots in both Figs. 6 and 7 have the same pattern. As the average arrival rate increases, the c^2 of the burst size of RR/R and RR/NP increases, and so does the burst loss rates. As for RR/P, as the average arrival rate increases, the burst size c^2 first increases, then peaks at 1.4 Gbps,

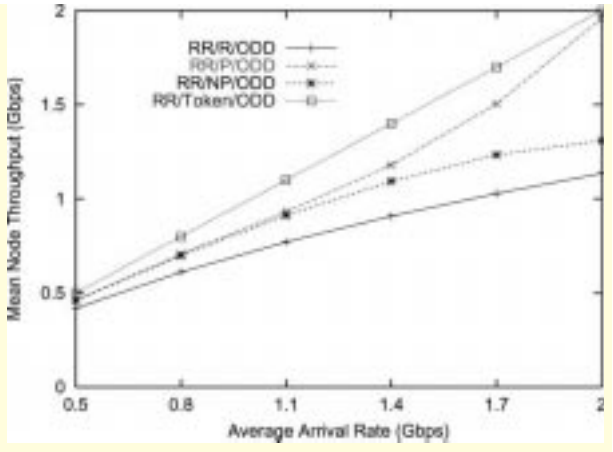

Fig. 5. Mean node throughput vs. average arrival rate.

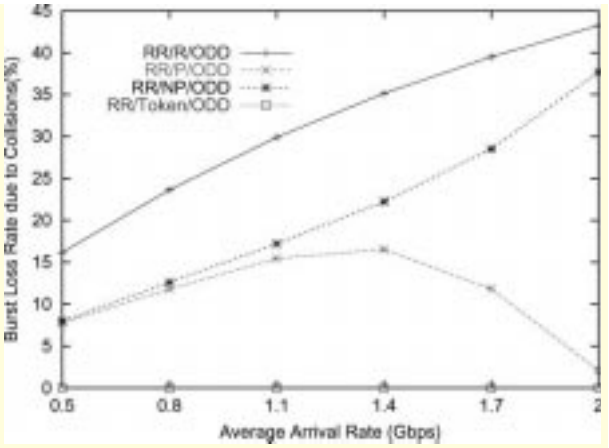

Fig. 6. Burst loss rate due to receiver collisions vs. average arrival rate.

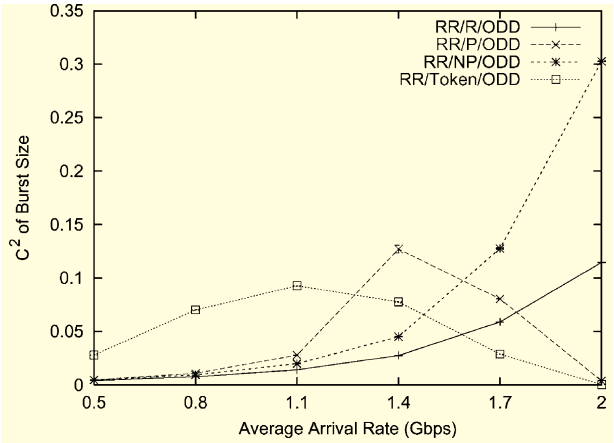

Fig. 7. c^2 of burst size vs. average arrival rate.

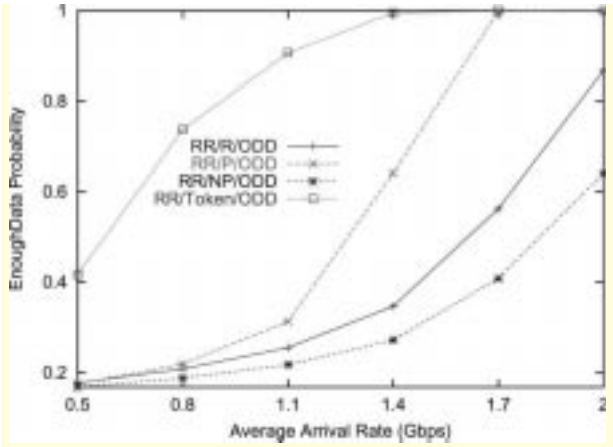

Fig. 8. `EnoughData` probability vs. average arrival rate.

and finally it decreases. The burst loss rate follows the same pattern. The reason for the change in the c^2 of burst size is that when the burst size reaches a specific point, the `MaxBurstSize` starts to limit the c^2 of burst size.

From the simulation, we also found that the burst loss rate due to receiver collisions of RR/P depends not only on the c^2 of the burst size, but also on another important parameter, the `EnoughData` probability. Recall that in a node, a transmit queue is not eligible for service unless its size is at least equal to the value of `MinBurstSize`. Therefore, when a node turns to serve a transmit queue, the transmit queue may or may not be eligible for service. The probability that a transmit queue is eligible for service when a node turns to serve it is the `EnoughData` probability. We found that for RR/P, an `EnoughData` probability equal to or very close to one leads to a lower burst loss rate due to receiver collisions than an `EnoughData` probability close to zero. Fig. 8 shows the `EnoughData` probability versus the average arrival rate. The `EnoughData` probability of RR/P increases as the average arrival rate increases. Especially, it reaches almost 1 when the average arrival rate reaches 1.7 Gbps.

Fig. 9 plots the mean packet delay versus the average arrival rate. The mean packet delay is the average packet delay over all transmit queues and nodes, where the packet delay includes both the queueing and the propagation delay. The packet queueing delay is defined as the time interval from

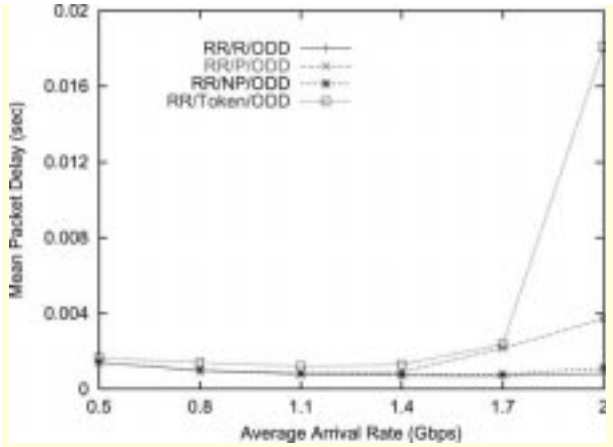

Fig. 9. Mean packet delay vs. average arrival rate.

the instance that the packet arrives at a node to the instance that the packet leaves the node. RR/R has the least delay, followed by RR/NP, RR/P and RR/Token. We observe that, as the average arrival rate increases, the mean packet delay in all protocols first decreases, and then it increases. This is due to the fact that when the traffic intensity is low, the time for a transmit queue to reach the `MinBurstSize` accounts for the major part of the packet delay. Therefore, as the average arrival rate increases, the time for a transmit queue to reach `MinBurstSize` decreases, which causes the mean packet delay to decrease. The 95% percentile packet delay was also calculated in the simulation. Since the plot trend is the same as that of the mean packet delay, the figure is not shown here.

Let us now compare the four protocols in terms of fairness. We distinguish two types of fairness, namely, throughput fairness and delay fairness. We define the *throughput fairness index of a node i* as the c^2 of the throughput from node i to all other nodes:

$$\text{throughput fairness index of node } i = \left(\sum_{j=1, j\neq i}^{10} (H_{ij} - \overline{H_i})^2 \right) \frac{1}{\overline{H_i}^2} \quad (6)$$

where H_{ij} is the throughput from node i to node j, i.e., the average number of bits transmitted by node i and received by node j in a unit time, and $\overline{H_i} = (\sum_{j=1,j\neq i}^{10} H_{ij})/9$. We then define the *throughput fairness index of a protocol* as the average of the throughput fairness indeces of all nodes. According to this definition, the smaller the throughput fairness index of a protocol, the better the throughput fairness of the protocol.

Fig. 10 shows the throughput fairness index of the four protocols versus the average arrival rate. We observe that RR/R and RR/Token have values very close to zero, meaning that they are throughput fair protocols. In [10], we give additional figures of the throughout from node 0 to all other nodes under the four protocols. We observed that both RR/NP and RR/P protocol provide better throughput to nodes closer to the source than to nodes far away. This follows directly from the operation of RR/NP and RR/P described in Sections 3.3 and 3.2.

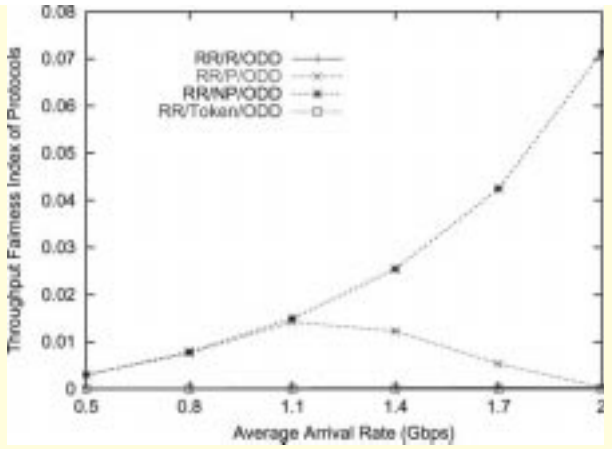

Fig. 10. Throughput fairness index of protocols vs. average arrival rate.

The second type of fairness we consider is related to delay. For this, we define the *delay fairness index of a node i* as the c^2 of the mean packet queueing delay of the transmit queues. We have

$$\text{delay fairness index of node } i = \left(\sum_{j=1, j\neq i}^{10} (W_{ij} - \overline{W_i})^2 \right) \frac{1}{\overline{W_i}^2} \quad (7)$$

where W_{ij} is the mean queueing delay of a packet in transmit queue j in node i, and $\overline{W_i} = (\sum_{j=1,j\neq i}^{10} W_{ij})/9$. We also define the *delay fairness index of a protocol* as the average of the delay fairness indeces of all nodes. (Note that in defining the fairness index we use the queueing delay only, not the total delay which also includes the propagation delay which depends on the destination node). According to this definition, the smaller the delay fairness index of a protocol, the better the delay fairness of the protocol. Specifically, if the delay fairness index of a protocol is zero, the protocol is perfectly fair since the queueing delay of a packet is insensitive to the source and destination of the packet. For unfair protocols, access to the burst wavelengths may depend on factors such as the relative position of the source and destination nodes in the ring. In this case, some transmit queues may take longer to serve than others, increasing the queueing delay of the respective packets relative to others, and thus, increasing the delay fairness index of the node and protocol.

Fig. 11 shows the delay fairness index of the four protocols versus the average arrival rate. We observe that only RR/R has delay fairness index values very close to zero, meaning that it is the only fair protocol in terms of delay. In [10], we give additional figures of the mean packet queueing delay of each transmit queue in node 0 for all protocols. We observed that RR/NP provides better delay access to wavelengths of nodes far away than to wavelengths of nodes close to the source of bursts, and RR/P and RR/Token do not always provide the best or worst delay access to a specific node. For further details, the reader is referred to [10].

Overall, based on the above experimentation, RR/Token achieves the highest mean node throughput, followed by RR/P, RR/NP and RR/R.

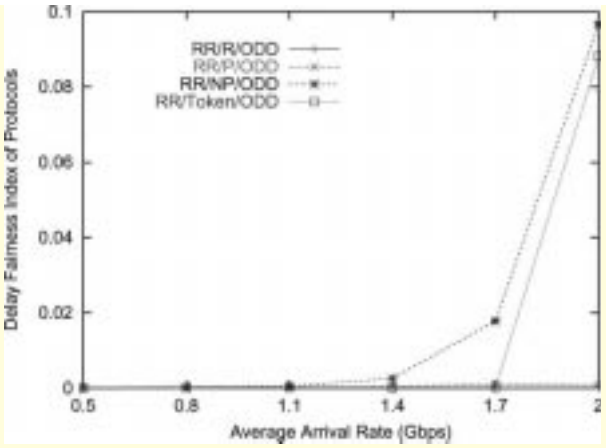

Fig. 11. Delay fairness index of protocols vs. average arrival rate.

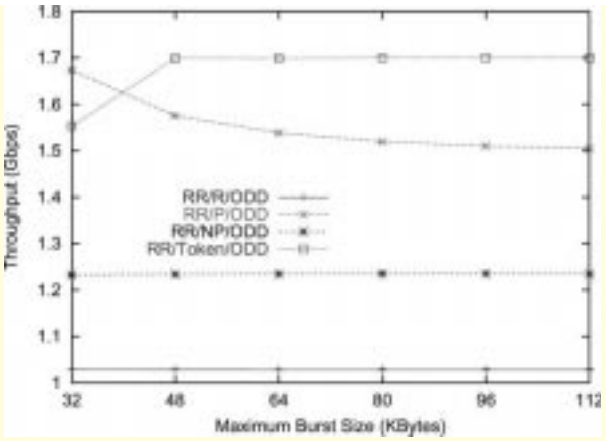

Fig. 12. Mean node throughput vs. maximum burst size.

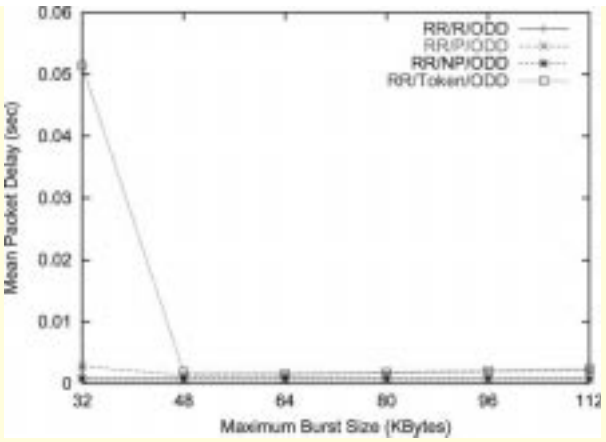

Fig. 13. Mean packet delay vs. maximum burst size.

RR/R has the smallest mean packet delay, followed by RR/NP, RR/P and RR/Token. RR/R also requires the smallest mean buffer requirement, followed by RR/NP, RR/P and RR/Token. The burst loss rate due to receiver collisions for the protocols which are not receiver collision free depends on the burst size c^2. The burst loss rate of RR/P also depends on the `EnoughData` probability. Only RR/R is a delay fair protocol, while both RR/R and RR/Token are throughput fair protocols.

4.1.2. *Effect of* `MaxBurstSize`

We varied the value of `MaxBurstSize` from 32 to 112 KB with an increment of 16 KB. `MinBurstSize` is 16 KB. The average arrival rate to each node is 1.7 Gbps, c^2 of the packet inter-arrival time at each node is 20, and, and `TimeOut` is 4 ms. A packet arriving at a node is assigned a destination node following the uniform distribution.

Simulation results showed that an increase in `MaxBurstSize` leads to an increase in the burst size c^2 and to a small change of the `EnoughData` probability, which lead to the increase in the burst loss rate due to receiver collisions, and finally lead to the decrease in the throughput of RR/R, RR/NP, and RR/P, as shown in Fig. 12. However, the decrease in the throughput of RR/R and RR/NP is very small. RR/Token requires a large `MaxBurstSize` so that no packets will be lost due to buffer overflow.

Fig. 13 plots the mean burst delay against `MaxBurstSize`. We observe that the delay of RR/R and RR/NP is not sensitive to the `MaxBurstSize`. We also observe that as the `MaxBurstSize` increases, the delay of RR/P and RR/Token first decreases, and then increases. The intuitive reason for the decrease is that a larger `MaxBurstSize` means that a node can transmit more packets at a time, which makes the delay go down. The intuitive reason for the increase is that a larger `MaxBurstSize` permits other nodes to transmit more packets at a time, so that the node must wait for a longer time to transmit its burst, which causes the delay to increase. Finally, we observe that only a very small `MaxBurstSize` can lead to a very long delay.

4.1.3. *Effect of* `MinBurstSize`

We varied `MinBurstSize` from 16 to 96 KB with an increment of 16 KB. `MaxBurstSize` is

112 KB. The average arrival rate to each node is 1.7 Gbps, c^2 of the packet inter-arrival time at each node is 20, and `TimeOut` is 4 ms. A packet arriving at a node is assigned a destination node following the uniform distribution.

Simulation results showed that an increase in `MinBurstSize` leads to a decrease in the burst size c^2 and a decrease in `EnoughData`. For RR/R and RR/NP, the decrease in the burst size c^2 leads to a small decrease in the burst loss rate due to collisions, which finally leads to a small increase in the mean node throughout, as shown in Fig. 14. However, for RR/P, a big decrease in the `EnoughData` probability leads to an increase in the burst loss rate due to receiver collisions, which finally leads to a decrease in the mean node throughout. Changes in `MinBurstSize` do not lead to any change in the mean node throughput of RR/Token. Increases in `MinBurstSize` also lead to increases in the mean packet delay of all protocols, as shown in Fig. 15.

4.2. JET vs. ODD

In this section we focus on the difference between the JET and ODD offset calculations. In our comparisons, we will only consider two protocols: RR/Token and RR/R. RR/Token is selected since it is free of receiver collisions, while RR/R is selected as a representative protocol among the three protocols that suffer from receiver collisions.

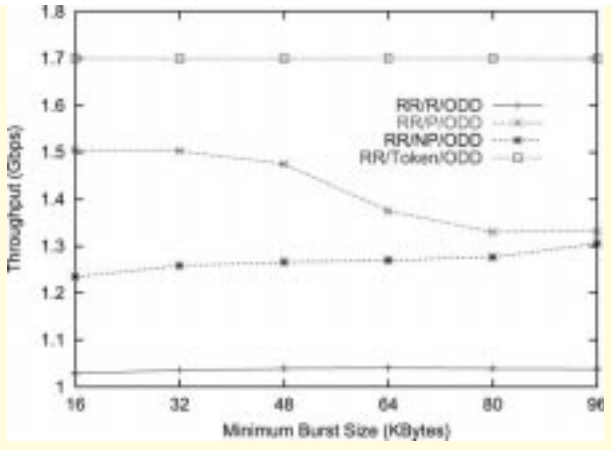

Fig. 14. Mean node throughput vs. minimum burst size.

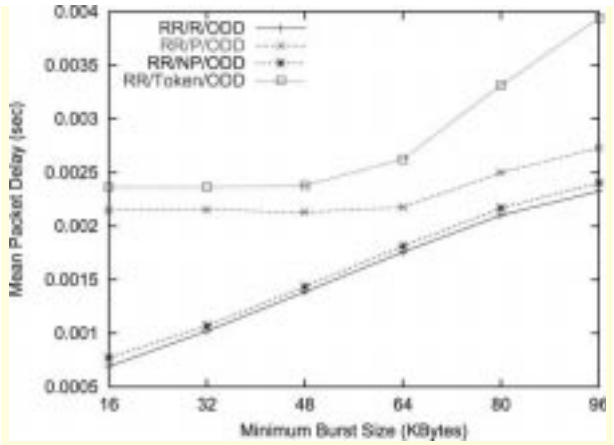

Fig. 15. Mean packet delay vs. minimum burst size.

Simulation experiments were carried out with the same parameters as in Section 4.1. The results showed that, compared to ODD, JET leads to a longer mean packet delay for all protocols (see Fig. 16), which in turn leads to a larger mean buffer requirement (see Fig. 17), and to a larger packet loss rate due to buffer overflow (see Fig. 18). Therefore, as a receiver collision protocol, RR/Token has a lower mean node throughput with JET than with ODD. Moreover, JET naturally leads to delay unfair protocols, but does not change the throughput fairness property of the protocols.

The effect of `MaxBurstSize` was also investigated. The results showed that all protocols are more sensitive to `MaxBurstSize` with JET than with ODD. A much larger `MaxBurstSize` is

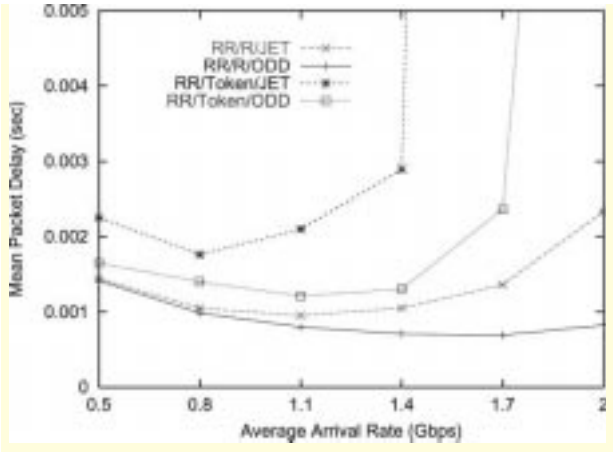

Fig. 16. Mean packet delay vs. average arrival rate.

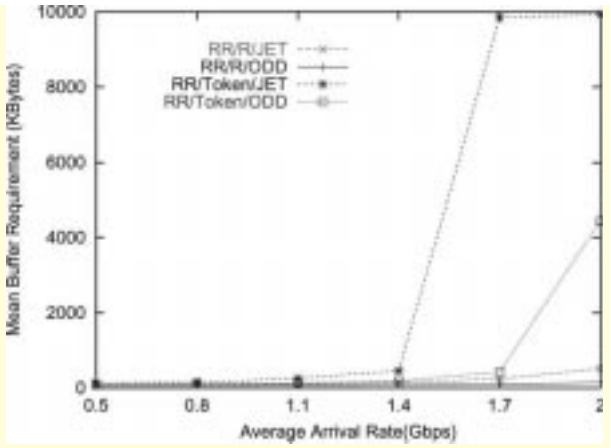

Fig. 17. Mean buffer requirement vs. average arrival rate.

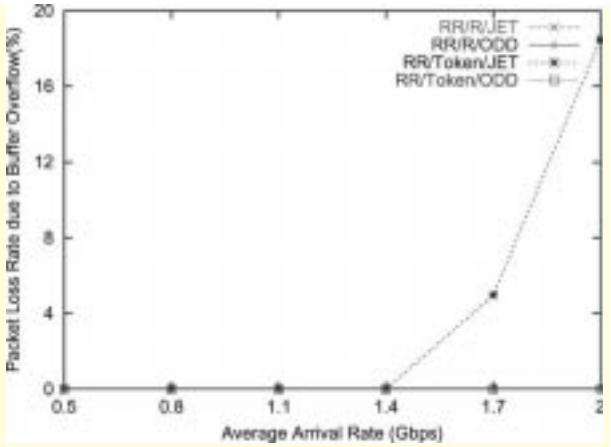

Fig. 18. Packet loss rate due to buffer overflow vs. average arrival rate.

required in JET than in ODD, in order to get a high mean node throughput and low mean packet delay.

Results also showed that both ODD and JET are not very sensitive to `MinBurstSize`. As the `MinBurstSize` increases, for RR/R, there is no big difference between ODD and JET. But for RR/Token, ODD is always much better than JET in both the mean node throughput and the mean packet delay.

4.3. TAW vs. ODD

RR/ACK is the only protocol using TAW, and it is receiver collision free. We compared RR/Token using ODD with RR/ACK using TAW using the same parameters as in Section 4.1.

Simulation results showed that when the `MaxBurstSize` is small, RR/Token with ODD gets both a higher mean node throughput and a lower mean packet delay than RR/ACK with TAW, as shown in Figs. 19 and 20, respectively. When the `MaxBurstSize` is large, in most cases, both protocols have similar mean node throughput and mean packet delay. However, when both the average arrival rate and the `MaxBurstSize` are very large, RR/ACK gives a higher mean node throughput and lower mean packet delay than RR/Token. As for fairness, RR/ACK is a throughput fair protocol, but not a delay fair protocol.

4.4. Asymmetric traffic

In this section, we investigate the performance of the protocols under asymmetric traffic. We vary the average arrival rate from 0.5 to 1.3 Gbps with an increment of 0.2 Gbps, and set the c^2 of the packet inter-arrival time to 20, `MaxBurstSize` to 112 KB, `MinBurstSize` to 16 KB, and `TimeOut` to 4 ms. At all nodes except node 0, the probability that a packet is destined to node 0 is 1/6, and the probability to other nodes is uniformly distributed. At node 0, a packet is assigned a destination node following the uniform distribution.

Among the four protocols based on ODD, we found that RR/Token achieves the highest mean node throughput, followed by RR/P, RR/NP and

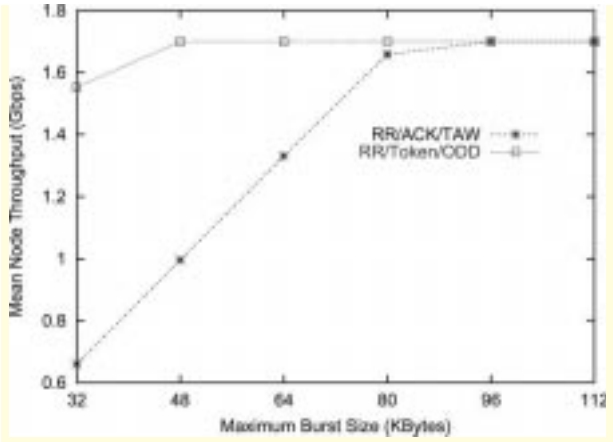

Fig. 19. Mean node throughput vs. maximum burst size.

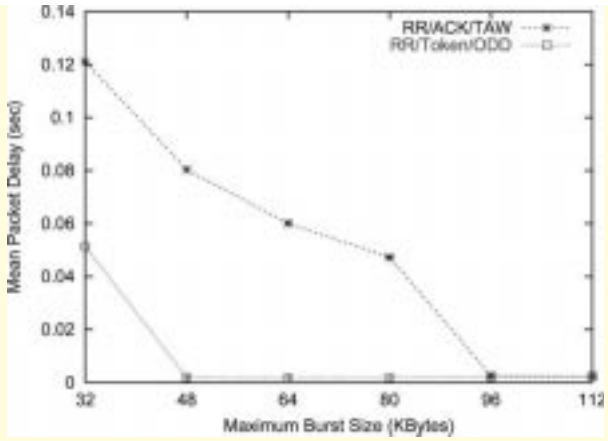

Fig. 20. Mean packet delay vs. maximum burst size.

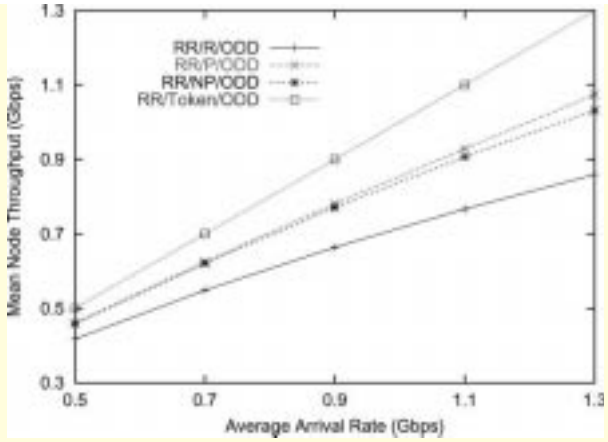

Fig. 21. Mean node throughput vs. average arrival rate.

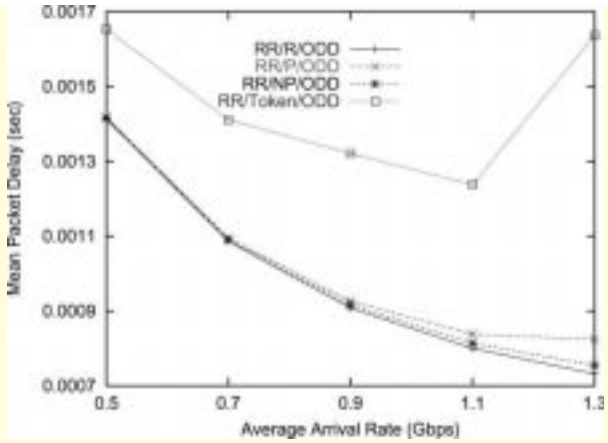

Fig. 22. Mean packet delay vs. average arrival rate.

RR/R (see Fig. 21), and RR/R has the smallest mean packet delay, followed by RR/NP, RR/P and RR/Token (see Fig. 22). We also compared the simulation results of RR/R and RR/Token based on ODD with that of RR/R and RR/Token based on JET, and we found that JET leads to a longer mean packet delay for both protocols. Finally, we found that RR/ACK with TAW achieves the same mean node throughput as RR/Token with ODD, but has a longer mean packet delay than RR/Token with ODD.

5. Concluding remarks

This paper described a WDM metro ring architecture with OBS. Several access protocols are proposed and their performance is analyzed by simulation.

Based on our experimentation, we found that, RR/Token achieves the highest mean node throughput, followed by RR/P, RR/NP and RR/R. RR/R has the smallest mean packet delay, followed by RR/NP, RR/P and RR/Token. `MaxBurstSize` affects both the mean node throughput and the mean packet delay, but only a very small `MaxBurstSize` leads to a much lower mean node throughput under RR/Token, and a much longer mean packet delay under RR/P and RR/Token. Increases in `MinBurstSize` lead to an increase in the mean packet delay of all protocols, but do not affect the mean node throughput of RR/Token. We also observed that JET leads to a longer mean packet delay and to a larger packet loss rate due to buffer overflow than ODD. The protocols become more sensitive to `MaxBurstSize` with JET than with ODD. Compared to RR/Token with ODD, RR/ACK with TAW achieves better performances only when both the `MaxBurstSize` and the average arrival rate are very large. In the simulations with symmetric traffic, we found that only RR/R is a delay fair protocol, while both RR/R and RR/Token are throughput fair protocols.

Finally, we note that the results were obtained by setting `MaxBurstSize` and `MinBurstSize` to values which are not very small. If we set them to very small values (for example, set

MinBurstSize to zero), we will get different results. However, very small values of MaxBurstSize and MinBurstSize are not reasonable, since some optical device overheads (e.g., the setup delay of a node) may greatly degrade the performance of the ring.

References

[1] J. Cai, A. Fumagali, I. Chlamtac, The multitoken interarrival time (MTIT) access protocol for supporting variable size packets over WDM ring network, IEEE Journal on Selected Areas in Communications 18 (10) (2000) 2094–2104.

[2] W. Fischer, K. Meier-Hellstern, The Markov-modulated Poisson process (MMPP) cookbook, Performance Evaluation 18 (1992) 149–171.

[3] B. Mukherjee, Optical Communication Networking, McGraw-Hill, New York, 1997.

[4] C. Qiao, M. Yoo, Optical burst switching (OBS)—A new paradigm for an optical Internet, Journal of High Speed Networks 8 (1) (1999) 69–84.

[5] J.S. Turner, Terabit burst switching, Journal of High Speed Networks 8 (1) (1999) 3–16.

[6] S. Verma, H. Chaskar, R. Ravikanth, Optical burst switching: A viable solution for terabit IP backbone, IEEE Network 14 (2000) 48–53.

[7] I. Widjaja, Performance analysis of burst admission control protocols, IEEE Proceeding of Communications 142 (1995) 7–14.

[8] Y. Xiong, M. Vandenhoute, H.C. Cankaya, Control architecture in optical burst-switched WDM networks, IEEE Journal on Selected Areas in Communications 18 (10) (2000) 1838–1851.

[9] L. Xu, H.G. Perros, G.N. Rouskas, Techniques for optical packet switching and optical burst switching, IEEE Communications 39 (1) (2001) 136–142.

[10] L. Xu, H.G. Perros, G.R. Rouskas, A simulation study of access protocols for optical burst-switched ring networks, Technical report, Computer Science Department, North Carolina State University, Raleigh, NC, 2001.

[11] S. Yao, S. Dixit, B. Mukherjee, Advances in photonic packet switching: An overview, IEEE Communications 38 (2) (2000) 84–94.

Lisong Xu received B.S. and M.S. degrees in Computer Science from University of Science and Technology Beijing, China, in 1994 and 1997, respectively, and received a Ph.D. degree in Computer Science from North Carolina State University, in 2002. He is currently a Post-Doc and an Adjunct Assistant Professor in Computer Science Department at North Carolina State University.

H.G. Perros is a Professor of Computer Science, an Alumni Distinguished Graduate Professor, and Program Coordinator of the Master of Science degree in Computer Networks at NC State University. He received the B.Sc. degree in Mathematics in 1970 from Athens University, Greece, the M.Sc. degree in Operational Research with Computing from Leeds University, England, in 1971, and the Ph.D. degree in Operations Research from Trinity College Dublin, Ireland, in 1975. He has held visiting faculty positions at INRIA, Rocquencourt, France (1979), NORTEL, Research Triangle Park, North Carolina (1988–1989 and 1995–1996) and University of Paris 6, France (1995–1996, 2000, and 2002). He has published extensively in the area of performance modelling of computer and communication systems, and he has organized several national and international conferences. In 1994, he published a monograph entitled "Queueing networks with blocking: exact and approximate solutions", Oxford Press, and in 2001 a textbook entitled "An introduction to ATM networks", Wiley. He is the chairman of the IFIP Working Group 6.3 on the Performance of Communication Systems, and a member of IFIP Working Groups 7.3 and 6.2. He is also an IEEE Senior Member. His current research interests are in the areas of optical networks and satellites.

George N. Rouskas is a Professor of Computer Science at North Carolina State University. He received the Diploma in Computer Engineering from the National Technical University of Athens (NTUA), Greece, in 1989, and the M.S. and Ph.D. degrees in Computer Science from the College of Computing, Georgia Institute of Technology, Atlanta, GA, in 1991 and 1994, respectively. He is a recipient of a 1997 NSF Faculty Early Career Development (CAREER) Award, and a co-author of a paper that received the Best Paper Award at the 1998 SPIE conference on All-Optical Networking. He also received the 1995 Outstanding New Teacher Award from the Department of Computer Science, North Carolina State University, and the 1994 Graduate Research Assistant Award from the College of Computing, Georgia Tech. During the 2000–2001 academic year he spent a sabbatical term at Vitesse Semiconductor, Morrisville, NC, and in May and June 2000 he was an Invited Professor at the University of Evry, France. He was a co-guest editor for the IEEE Journal on Selected Areas in Communications, Special Issue on Protocols and Architectures for Next Generation Optical WDM Networks, published in October, 2000, and is on the editorial boards of the IEEE/ACM Transactions on Networking, Computer Networks, and Optical Networks. He is a Senior Member of the IEEE, and a member of the ACM and of the Technical Chamber of Greece.