A scalable and collision-free MAC protocol for all-optical ring networks

Ashok K. Turuk*, Rajeev Kumar

Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, WB 721 302, India

Abstract

In this paper, we propose a collision free MAC protocol for an all-optical WDM ring network where packets remain in optical domain from source to destination. The protocol prevents the contention for shared data-channels and the destination by sharing global status information among the nodes. The proposed protocol which we call reserve–transmit–release (RTR) is based on a reservation scheme and requires no synchronization among the nodes. The node architecture in this scheme uses a tunable transceiver and, thus, makes the scheme scalable. We study the performance of the algorithm by network simulation. We consider both the Poisson and Pareto distributions for modeling the traffic. We present results for both the models and compare. It is observed that the delay is much higher for bursty traffic and the performance degrades gracefully with increase in nodes as well as the traffic.

Keywords: Optical ring network; Wavelength division multiplexing; MAC protocol; Tunable transceiver; Pareto distribution

1. Introduction

It is widely acknowledged that the rapid growth in demand for bandwidth due to the Internet explosion can be satisfied by optical networks, and in particular using the wavelength-division multiplexing (WDM) technology. A single fiber can support hundreds of wavelength channels. With the successful deployment of WDM in core networks the access networks viz. local area networks (LANs) and metropolitan area networks (MANs) are the bottlenecks. Recently lot of work has been reported in the literature for the deployment of WDM technology in the access network. Most of the work on LANs reported in the literature employ either a star or a ring as the underlying physical topology of the network. In a LAN the available bandwidth is shared among all the network users. To deal with multiuser access a media access control (MAC) protocol is needed in such networks. In recent years many MAC protocols have been proposed for WDM LANs based on star or a ring as the underlying physical topology [20].

LANs based on star as the underlying physical topology contains a passive star coupler with N inputs and N outputs where N is the number of nodes in the network. The star coupler combines the transmissions from different stations and relays it to the output. Nodes are mostly equipped with a tunable/fixed transmitter and a fixed/tunable receiver. Ganz and Koren [1] proposed a node architecture using tunable transmitter and fixed receiver where the tunability range of the transmitter is limited to few wavelengths, and the receiver receives from a set of pre-assigned wavelengths. Janniello et al. [2] discussed a MAN called Rainbow which has a fixed transmitter and a tunable receiver and it uses broadcast-and-select network architecture. Jia and Mukherjee [3], in their architecture used a separate control channel for the pre-transmission coordination between the source and destination and assigned a fixed control slot to each node on the control channel. In order to overcome the performance penalty of large tuning time the packet transmission of one node is overlapped with the tuning time of other node in their architecture. Li et al. [4] assigned a data-channel and a signaling channel to each node. The signaling channel can be shared among several nodes. In their architecture, the number of data channels has to be increased in proportion to the increase in the number of nodes. Moreover, it may also require to increase the number of control channels with increase in the number of nodes. Laarhuis and Koonen [5] divided a given network into disjoint sub-networks on the basis of wavelengths. In their work a node may have a transmit function in one sub-network and a receive function in another in the same time

* Corresponding author. Tel.: +32-2228-3464; fax: +32-2227-9579.
E-mail address: akturuk@cse.iitkgp.ernet.in (A.K. Turuk).

doi:10.1016/j.comcom.2004.04.004

slot. Synchronization among the slots is an issue in such networks. Spencer and Summerfield [6] proposed an architecture which allocates the bandwidth in three-stages—request, allocate, and transmit.

LANs based on ring topology [7–10] mostly use slotted channels. Slot synchronization is a major design factor in the networks. Nodes must be synchronized so that a slot starts at the same time on all the wavelengths in the network. Synchronization points are the network hub for star topology and the WDM ADMs for ring topology [6]. Bengi et al. [7] discussed multihop WDM metro ring where several nodes use the same channel for reception of packets. Marsan et al. [8] discussed an all-optical network where number of nodes in the network are equal to the number of wavelength channels. In their modified work, Marsan et al. [9] assigned separate slotted channels to a disjoint subset of destination nodes. A group of nodes share the same channel for reception. Fransson et al. [11] equipped the nodes with a WDM laser-array transmitter capable of transmitting at all wavelengths used in the network, and receive at a particular wavelength.

In such LAN where nodes are equipped with fixed or an array of transmitter/receiver or where the number of nodes is equal to the number of available wavelengths, the network's scalability is constrained. This is because for any change in the number of nodes in the network there has to be a corresponding change in the number of transmitters/receivers at each node in the network. A network where nodes are equipped with tunable transceivers, such constraints are not applicable and, the network is scalable.

To equip a node with tunable transmitter/receiver, it is desirable that the transmitter/receiver should change channels quickly to reduce the channel access time. Tunable lasers with tuning time of less than 5 ns is reported in the literature. Chan et al. [12] discussed tunable lasers with channel switching time of less than 5 ns among sixteen 1-Gbps channels in a 16-Gbps all-optical TDMA networks. Lavrora et al. [13] reported tunable laser with tuning time of less than 5 ns among 40 ITU channels. Mason et al. [14] discussed sampled grating distributed bragg reflector tunable lasers, with a tuning time of approximately 5 ns among 50 channels. Doerr et al. [15] discussed a multi-frequency laser with expected tuning time of less than 10 ns within 40 channels. Thus, the state-of-the-art of technology permits the use of tunable transceivers with very fast switching response.

In this paper we use tunable transceivers and propose a MAC protocol for all-optical LANs based on ring topology. We identify two types of collisions viz. destination collision and channel collision in the network. The proposed protocol is free from both types of collisions and operates in three stages: reservation, transmission and release, hence we call the proposed protocol a reserve–transmit–release (RTR) algorithm. In the first stage, receiver of the destination node and a data-channel is reserved, then the source starts transmitting its packets, and finally the source releases the reserved sources after completion of the transmission.

A node architecture with tunable components is proposed by Marsan et al. [10]. However in their model, the tunability range is limited to a subset of available wavelength whereas in our proposed architecture the range of tunability is the entire set of available wavelengths. Unlike [7] where packets may have to undergo into electronic domain at intermediate nodes in the present work packets remain in the optical domain from source to destination. It takes a single hop to transmit data from the source to destination. Most of the MAC protocols proposed for optical ring networks operate on slotted mode which requires a mechanism for synchronization at slot boundaries. Our present work is based on a reservation scheme which requires no synchronization and is simple to implement.

We define scalability of a network in two ways. We call a network scalable if its performance does not degrade drastically with increase in number of nodes. However, the performance does degrade with increase in load and nodes, a desirable feature of the network is that such degradation should be graceful. Next, we call a network scalable if a change in the number of nodes does not necessitate for additional resource requirements at each other node. In this work, we use tunable components and thus, an increase in nodes does not have any additional hardware requirements. Additionally, we demonstrate a graceful degradation in performance with increase of nodes.

The rest of the paper is organized as follows. In Section 2, we describe node architecture and notations used in the paper. The contention free RTR algorithm is detailed in Section 3 along with a correctness proof for no collision. We include simulation results in Section 4 and show graceful degradation in performance with increase in number of nodes and the traffic. Finally, conclusions are drawn in Section 5.

2. System model

2.1. Node architecture

There are N nodes in the network, numbered as 0, 1, 2, …, N − 1. Node i of the network is connected to node j by an optical fiber such that $j = (i+1) \bmod N$ where $i \neq j$, and for all $i, j \in 0, 1, 2, \ldots, N-1$; node j is the successor of node i and node i is the predecessor of node j; such a pair of nodes i and j is said to be adjacent. R is the ring latency. The system supports W wavelengths $\lambda_0, \lambda_1, \ldots, \lambda_{W-1}$. There are $W-1$ data-channels and one control-channel. One of the wavelength, λ_0, is dedicated to a control-channel, and rest of the wavelengths are used as data-channels. A circuit is established on wavelength, λ_0, between every pair of adjacent nodes i and j. The circuit thus established is the dedicated control-channel.

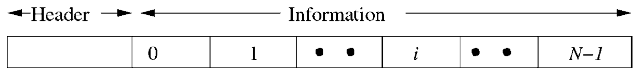

Fig. 1. A token showing N slots for an N-node ring network.

Each node is equipped with one fixed transceiver and one tunable transceiver. The fixed transmitters/receivers are tuned to wavelength, λ_0, to transmit and receive control information between adjacent nodes. Tunable transmitters/receivers are tuned to data-channels as and when required. For two nodes in the network to communicate, tunable transmitter of the source node and tunable receiver of the destination node must be tuned to the same wavelength (data-channel). Information transfer takes a single hop over a data-channel.

The system has a single token that circulates around the ring on the control-channel. Structure of the token is shown in Fig. 1. The token consists of header field and information field. Header field specifies the number of slots in the information field along with other header fields. The information field consists of N fields which we call slots with slot i assigned to node i. Each slot is subdivided into three mini-fields which we collectively called control information of a slot.

Control information in a $\text{slot}_i(x, y, z)$ of the token are:

x : a zero indicates node i is transmitting/idle, one indicates node is requesting for reservation, and two indicates node is releasing the resources,

y : identity of the destination node requested for reservation by node i, and

z : identity of the data-channel requested for reservation by node i.

We define a token period (TP) as the period between two successive receives of the token by a node. We calculate TP as $\text{TP} = ds/v + Nb$ bits where d is the length of the ring, s is the rate of control-channels, N is the number of nodes in the network, and b is the delay in bits for processing of token at each node. Since TP is same for all nodes in the network, each node gets a fair chance to access the shared medium.

A node on receiving the token processes a slot, $l, (0 \leq l < N)$ only if the node l is requesting or releasing the reservation. Prior to communication between a pair of nodes, the source must reserve the destination and a data-channel. A node reserves the destination and a data-channel by writing control information at its assigned slot in the token. Reservation mechanism is explained in Section 3.

Every node has one *IN* and one *OUT* buffer. Packets that are terminated at a node are stored in IN buffer and that are originated from the node are stored in OUT buffer. A detailed node architecture as used in our ring network (Fig. 2) is shown in Fig. 3; the data structures

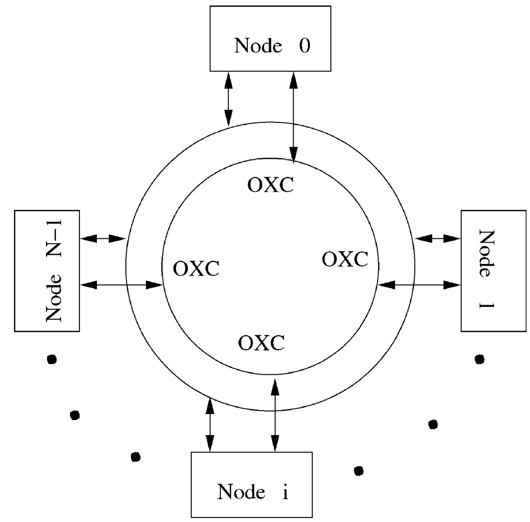

Fig. 2. An optical ring network with N nodes.

and the fields used in the figure are explained in the subsequent Section 2.2.

2.2. Notations and definitions

Each node maintains the status of other nodes and data channels in the network. The status is maintained by the following vectors: destination available table (DAT), destination release table (DRT), channel available table (CAT) and channel release table (CRT). DAT is a vector of N elements indicating the status of other nodes, and CAT is a vector of W elements indicating the status of data-channels. DAT_i and CAT_i indicate the DAT and CAT vectors at node i respectively. At node i, the semantics of each of the fields of the vectors is given below:

$\text{DAT}_i[j]$, indicates the status of the receiver of node j(free/busy) for $0 \leq j < N$ and $i \neq j$. A value of one indicates receiver of node j is busy and a value of zero indicates the receiver is free.

$\text{CAT}_i[c]$, indicates status of the wavelength, λ_c (free/busy) for $1 \leq c < W$. A value of one indicates wavelength, λ_c is busy and a value of zero indicates the wavelength is free.

$\text{DRT}_i[j]$, a value of one indicates node i is releasing destination node j.

$\text{CRT}_i[c]$, a value of one indicates node i is releasing the wavelength (data-channel) λ_c.

In addition to the above vectors each node maintains the following three flags:

Status (S): indicates the status of the node's transmitter (reserved/free). A value of zero indicates the node's transmitter is free and a value of one indicates it is reserved for transmission.

Finish (F): Indicates the status of transmission from the node (completed/not-completed). Value of one indicates the transmission is completed.

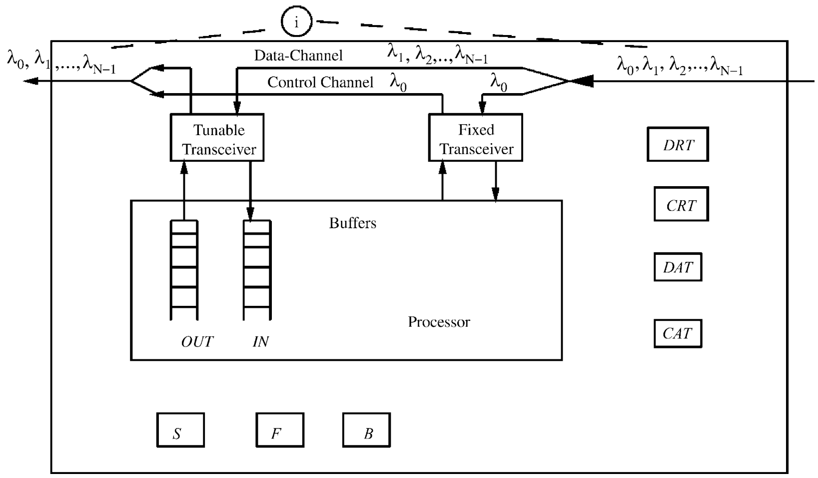

Fig. 3. Architecture of a node *i* used in the ring network. Description of the fields is included in text.

Busy (B): A value of one indicates the node is busy in transmitting and a value of zero indicates the transmission has not started.

3. Collision free reserve–transmit–release (RTR) algorithm

RTR algorithm operates in three steps: (i) reservation, (ii) transmission and (iii) release. In the first step a node reserves the following resources—transmitter of the source node, receiver of the destination node and a data-channel—required for transmission. Transmission takes place after resources are reserved. After completion of transmission the resources are released. It takes a TP period each to reserve and release the resources. A node having data to transmit waits for the arrival of the token. On receiving the token it updates the values of DAT and CAT vectors at its node. The node makes a reservation request if its transmitter was not busy. It checks DAT vector for the status of the destination node's receiver and CAT vector for status of the data-channel. A reservation request is made by the node if the destination node's receiver and a data-channel is free. It takes a TP period to complete the reservation request. In this TP period every other node is informed about the reservation. When the node receives back the token its reservation process is complete and it begins its transmission. After completion of the transmission the node must release the reserved resource. It waits for the arrival of the token after completion of transmission. On receiving the token it processes the token and writes release information at its allotted slot. It takes a TP period to release the resource. In this TP period every other node is informed about the release of the resource.

Since, every node maintains the status of every other nodes' receiver and data-channels in the network, no two nodes make request for the same destination node and the data-channel. In other words collision never occurs.

We first illustrate the working of RTR algorithm, in the following paragraphs, with an example and the pseudocode of the algorithm is given in Section 3.1. We use the notation $a \rightarrow b$ to denote that node a wants to communicate with node b. A slot of the token is processed only if its x value is non-zero. For illustration we consider a ring network with four nodes for simplicity but without loss of generality. In the subsequent paragraphs, we explain the Fig. 4 that we consider for illustrating the RTR algorithm.

Let at time instant, t, the contents of DAT and CAT vectors at nodes are as shown in Fig. 4(a). Consider the node '1' in Fig. 4 for our explanation. The DAT and CAT vectors at node '1' give the information about the receiver of node '1' and status of the data-channel λ_3 at time t. The status of S, B and F flags at nodes before receiving the token, and after sending the token are shown in Fig. 4(d). At node '1' all flags before receiving the token are set to zero. Token received at a node from its predecessor and sent by it to its successor is shown in Fig. 4(e). Token received at a node is indicated by r, and s indicates the token sent by it. Token received at node '1' has a reservation request from node '0' requesting for reservation of receiver of node '2' and data-channel λ_1. Fig. 4(b) shows the contents of the DAT and CAT vectors at nodes after processing the token. Contents of the DAT vector at node '1' after processing the token are shown in Fig. 4(b) which indicate that the receivers of node '1' and '2' are busy. Similarly, the contents of CAT vector indicate data-channel λ_1 and λ_3 are busy (Fig. 4(b)). Node '1' is making reservation request for the receiver of node '3' and data-channel λ_2 (Fig. 4(e)). Its reservation request is entered in its DAT and CAT vectors (Fig. 4(c)). Contents of the DAT and CAT vectors after sending the token to its successor node are shown in Fig. 4(c). Contents of the DAT vector at node '1' after sending the token to its successor node are shown in Fig. 4(c) which indicate that the receivers of nodes '1', '2' and '3' are busy. Similarly the contents of the CAT vector indicate the data-channels λ_1, λ_2 and λ_3 are busy (Fig. 4(c)). Note that before node '1' receives the token,

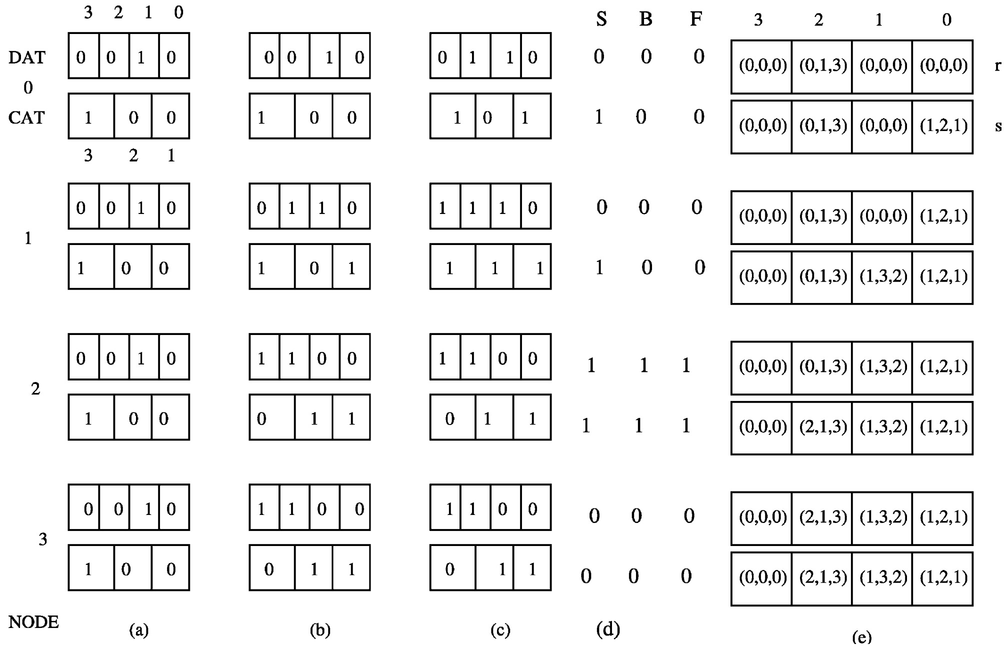

Fig. 4. Illustration of an example showing the DAT and CAT vectors, S, B and F flags, and the input and output tokens. 'r' is the input token, 's' the output token; r-token at node $(i+1)$ mod 4 is the same as the s-token at node i. (a) Status of DAT and CAT vectors before receiving the token, (b) Status of DAT and CAT vectors after processing the token, (c) Status of DAT and CAT vectors after sending the token to its successor node, (d) Status of S, B, F flags before receiving r-token and after sending s-token, and (e) input token (r) to a node, and output token (s) from the node.

the receiver of node '1' was set to busy. After processing the token it found a reservation request for the receiver of node '2'. So it sets the receiver of node '2' to busy. Node '1' itself is making a reservation request to node '3'. So it sets a receiver of node '3' to busy. Similar actions are taken for data-channels. The transmitter of node '1' is tuned to data-channel λ_3 by setting the S flag to one.

Let node '0' receive the token at time instant t and the requests at the nodes at time t be as follows: $0 \rightarrow 2, 1 \rightarrow 3, 3 \rightarrow 2$. The contents of the DAT and CAT vectors at node '0' after processing the token are shown in Fig. 4(b). Node '0' wishes to communicate with node '2'. For communication to take place, node '0' must reserve receiver of node '2' and a data-channel. It should check the vector element DAT [2] for zero and vector CAT for which the element CAT[c] is zero for $1 \leq c < W$. At node '0', the elements DAT [2] and CAT [1] are equal to zero (Fig. 4(b)). Node '0' requests for reservation of receiver of node '2' and data-channel λ_1. The following actions are performed at node '0': the vector elements DAT [2], CAT [1] and the flag S are set to one; transmitter of node '0' is tuned to λ_1; control information is written in slot '0' and the token is sent to node '1'. Node '1' on receiving the token performs the actions as mentioned for node '0' and sends the token to node '2' requesting for reservation of node '3' and data-channel λ_2. Node '2' processes the token and sends it to node '3'. From Fig. 4(e) it is seen that node '2' is releasing the resource-—receiver of node '1' and data-channel λ_3— which were held by it. The contents of DAT and CAT vectors at node '3' after processing the control token are shown in Fig. 4(b). Node '3' wishes to communicate with Node '2'. But it finds from its DAT vector that the receiver of node '2' is busy i.e., the vector element DAT[2] at its node is set to one. Hence, it makes no attempt to reserve the resources and sends the token to node '1' as shown in Fig. 4(e). When node '0' gets back the token its reservation request is completed. It sets B to one and starts transmitting again. Rest of the process continues as per the algorithm.

3.1. Pseudocode

The following steps are performed when a node i receives the token:

Step 1: if (transmission from node i is completed) F $\leftarrow$ 1.
Step 2: if (S = 1 and B = 0)
 –B $\leftarrow$ 1
 –Start transmitting and goto Step 3
Step 3: For each slot$_j(x, y, z)$ in the token, where $j \neq i$ do the following:
 Case $x = 1$: (This is the case of node j requesting for reservation)
  *if ($y = i$) do the following:
   1. DAT$_i[y] \leftarrow 1$
   2. CAT$_i[z] \leftarrow 1$
   3. Tune the receiver of node i to data-channel z
  (In the above case node j is requesting for transmission to node i. DAT and CAT vectors at node i are updated and the receiver is tuned to the requested data-channel z.)

*else

1. $\mathrm{DAT}_i[y] \leftarrow 1$
2. $\mathrm{CAT}_i[z] \leftarrow 1$

(In the above case node j is requesting for transmission to node y. DAT and CAT vectors at node i are updated to indicate that the receiver of node y and data-channel z are busy.)

Case $(x = 2)$: (This is the case of resource release message from node j)

*$\mathrm{DAT}_i[y] \leftarrow 0$

*$\mathrm{CAT}_i[z] \leftarrow 0$

*$\mathrm{CRT}_i[z] \leftarrow 1$

*$\mathrm{DRT}_i[y] \leftarrow 1$

(When all the slots are processed node i will have complete information on the status of the receiver of all nodes and data-channels in the network)

Step 4: Modify $\mathrm{slot}_i(x, y, z)$ as follows

−if $(x = 1) x \leftarrow 0$, values of y and z remain unchanged. (Node i requested for reservation in the last TP period)

−if ($x = 0$ and $\mathrm{F} = 0$) values of x, y, z remain unchanged. (Node i is idle or its transmission is not completed)

−if ($x = 0$ and $\mathrm{F} = 1$) $x \leftarrow 2$, values of y, z remain unchanged, $\mathrm{CRT}_i[z] \leftarrow 1$, and $\mathrm{DRT}_i[y] \leftarrow 1$. (Transmission from node i is completed and the reserved resources are to be released)

−if $(x = 2) x \leftarrow 0, \leftarrow 0, z \leftarrow 0, \mathrm{S} = \mathrm{B} = \mathrm{F} \leftarrow 0$. (Resources reserved by node i are released)

Step 5: if ($\mathrm{S} = 0$ and node i has data to send to node j for $i \neq j$)

−if ($\mathrm{DAT}_i[j] = 0$ and $\mathrm{DRT}_i[j] = 0$ and there exists k such that $\mathrm{CAT}_i[k] = 0$ and $\mathrm{CRT}_i[k] = 0$ for $1 \leq k < W$) then do the following:

*$\mathrm{S} \leftarrow 1$,

*$\mathrm{DAT}_i[j] \leftarrow 1$,

*$\mathrm{CAT}_i[k] \leftarrow 1$,

*Tune the transmitter of node i to data-channel k.

*Set the control information for node i to $x \leftarrow 1, y \leftarrow j, z \leftarrow k$.

Step 6: Write the control information for node i in slot i of the information field of the token

Step 7: Set all the elements of DRT and CRT vectors to zero, and send the token to successor node.

3.2. *Proof of correctness*

Destination collision occurs when a destination node receives data from two different nodes at the same time, and channel collision occurs when two different nodes select the same data-channel for transmission.

Claim 1: No two nodes i and j can reserve the destination node m during the interval t to $t + \mathrm{TP}$.

Proof: Suppose node i and node j reserved the node m at t and t_1 respectively, during the interval t to $t + \mathrm{TP}$ where $t < t_1 < t + \mathrm{TP}$. This implies that when node i reserved m at t both $\mathrm{DAT}_i[m]$ and $\mathrm{DAT}_j[m]$ elements were zero. Also when node j reserved m at t_1 both $\mathrm{DAT}_i[m]$ and $\mathrm{DAT}_j[m]$ elements were zero.

From the algorithm a node can reserve a destination only when it receives the token. Node i has reserved node m at t, this implies that node i has received the token at t, and has taken the following action. DAT at node i is updated, node m is reserved by setting $\mathrm{DAT}_i[m]$ to one, and the control information is written in slot i of the token. Node i then sends the token to its successor node.

Node j has reserved m at t_1, this implies that, node j has received the token at t_1. On receiving the token, node j shall update its DAT vector setting $\mathrm{DAT}_j[m]$ element to one. To reserve node m, node j checks $\mathrm{DAT}_j[m]$ element for zero. But $\mathrm{DAT}_j[m]$ element has been set to one which indicates that some other node has already reserved node m. Hence, node j fails to reserve node m. This contradicts that node j reserved node m at t_1 when node i has reserved m at t for $t < t_1 < t + \mathrm{TP}$. Similarly, it can be shown that when node j reserved node m at t, node i can not reserve node m at t_1 for $t < t_1 < t + \mathrm{TP}$. Thus, we conclude that no two nodes i and j can reserve a destination node m during the interval t to $t + \mathrm{TP}$. Note that when a node reserves a destination or a data channel, it informs every other node of its reservation, and this takes a TP period. □

Claim 2: No two nodes i and j can reserve the wavelength (data-channel), λ_c, during the interval t to $t + \mathrm{TP}$.

Proof: Suppose node i and node j reserved the wavelength, λ_c, at t and t_1 respectively during the interval t to $t + \mathrm{TP}$ where $t < t_1 < t + \mathrm{TP}$. This implies that when node i reserved λ_c at t both $\mathrm{CAT}_i[c]$ and $\mathrm{CAT}_j[c]$ elements were zero. Also, when node j reserves λ_c at t_1 both $\mathrm{CAT}_i[c]$ and $\mathrm{CAT}_j[c]$ elements were zero.

From the algorithm a node can reserve a wavelength only when it receives the token. Node i has reserved the wavelength λ_c at t, this implies that node i has received the token at t, and has taken the following action. CAT vector at node i is updated, wavelength λ_c is reserved by setting $CAT_i[c]$ element to one, and the control information is written in slot i of the token. Node i then sends the token to its successor node.

Node j has reserved the wavelength, λ_c, at t_1 which implies that node j has received the token at t_1. On receiving the token, node j updates its CAT vector, setting $\mathrm{CAT}_j[c]$ element to one. To reserve wavelength, λ_c, node j checks $\mathrm{CAT}_j[c]$ element for zero. But $\mathrm{CAT}_j[c]$ element has already been set to one, which indicates that some other node has already reserved wavelength λ_c. Hence, node j fails to reserve λ_c. This contradicts node j has reserved wavelength λ_c at t_1 when node i has reserved wavelength λ_c at t for

$t < t_1 < t + \mathrm{TP}$. Similarly, it can be shown that when node j reserved wavelength λ_c at t, node i can not reserve wavelength λ_c at t_1 for $t < t_1 < t + \mathrm{TP}$. Thus we conclude that no two nodes i and j can reserve wavelength λ_c during the interval t to $t + \mathrm{TP}$. □

Claim 3: Destination collision never occurs

Proof: Let node i start transmitting to node j at t, and the transmission continues for $t + t'$ where $n\mathrm{TP} \geq t'$ for $n \geq 1$.

Since node i started its transmission at t, it must have reserved node j during the interval $t - \mathrm{TP}$ to t. This reservation is possible only if some other node, say x has released node j during the interval $t - 2\mathrm{TP}$ to $t - \mathrm{TP}$, or node j is free during the interval $t - 2\mathrm{TP}$ to $t - \mathrm{TP}$ and no other node is reserving node j during this interval. Also, the following conditions hold good:

$$\mathrm{DAT}_m[i] = 0, \forall m = \{0, 1, \ldots, N-1\} \text{ at } t - \mathrm{TP} \tag{1}$$

$$\mathrm{DAT}_m[i] = 1, \forall m = \{0, 1, \ldots, N-1\} \text{ at } t \tag{2}$$

Suppose destination collision occurs at node j. For destination collision to occur at node j, there must exist a node k, that transmits to node j during the interval $t - \mathrm{TP} + \mathrm{PD}_{ik}$ to $t + t' + \mathrm{PD}_{ik}$.
We consider two extreme cases, i.e. $t - \mathrm{TP} + \mathrm{PD}_{ik}$ and $t + t' + \mathrm{PD}_{ik}$ to show that there does not exist a node k, that transmits to node j during the interval $t - \mathrm{TP} + \mathrm{PD}_{ik}$ to $t + t' + \mathrm{PD}_{ik}$.
Case 1: There does not exist a node k which transmits to node j at $t - \mathrm{TP} + \mathrm{PD}_{ik}$.
Suppose node k transmits to node j at $t - \mathrm{TP} + \mathrm{PD}_{ik}$. For node k to transmit at $t - \mathrm{TP} + \mathrm{PD}_{ik}$ it must reserve node j during the interval $t - 2\mathrm{TP}$ to $t - \mathrm{TP}$ and the following conditions must hold:

$$\mathrm{DAT}_m[i] = 0, \forall m = \{0, 1, \ldots, N-1\} \text{ at } t - 2\mathrm{TP} \tag{3}$$

$$\mathrm{DAT}_m[i] = 1, \forall m = \{0, 1, \ldots, N-1\} \text{ at } t - \mathrm{TP} \tag{4}$$

But condition (4) contradicts condition (1). That means when node i reserved node j during the interval $t - \mathrm{TP}$ to t, node j was not reserved by node k during the interval $t - 2\ \mathrm{TP}$ to $t - \mathrm{TP}$. Therefore node k can not transmit to node j at $t - \mathrm{TP} + \mathrm{PD}_{ik}$. Thus, there does not exist a node k which transmits to node j at $t - \mathrm{TP} + \mathrm{PD}_{ik}$.
Case 2: There does not exist a node k which transmits to node j at $t + t' + \mathrm{PD}_{ik}$.
Suppose node k transmits to node j at $t + t' + \mathrm{PD}_{ik}$. This is possible only if node k has reserved node j during the interval $t + (\lfloor t'/\mathrm{TP} \rfloor)\mathrm{TP} - \mathrm{TP}$ to $t + (\lfloor t'/\mathrm{TP} \rfloor)\mathrm{TP}$. Node k can reserve, only when node i releases node j. But node i will release node j at $t + (\lceil t' + \mathrm{TP} \rceil)\mathrm{TP} + \mathrm{TP}$, and node k can reserve node j only at $t + (\lceil t' + \mathrm{TP} \rceil)\mathrm{TP} + 2\mathrm{TP}$ (it takes a TP period for a node to reserve a destination). Therefore node k can not transmit to node j at $t + t' + \mathrm{PD}_{ik}$. □

From cases 1 and 2 we conclude that destination conflict never occurs.

Claim 4: Channel (data-channel) collision never occurs.

Proof: Let node i, begin its transmission on wavelength λ_m at t, and the transmission continues for $t + t'$ where $n\mathrm{TP} \geq t'$ for $n \geq 1$. To transmit at t, node i must reserve the wavelength λ_m during the interval $t - TP$ to t, and the following conditions hold good:

$$\mathrm{CAT}_l[m] = 0, \forall l \in \{0, 1, \ldots, N-1\} \text{ at } t - \mathrm{TP} \tag{5}$$

$$\mathrm{CAT}_l[m] = 1, \forall l \in 0, 1, \ldots, N-1\} \text{ at } t \tag{6}$$

Suppose channel collision occurs on wavelength λ_m. For this to happen, there must exist a node j that transmits on wavelength λ_m during the interval $t - \mathrm{TP} + \mathrm{PD}_{ij}$ to $t + t' + \mathrm{PD}_{ij}$.

We consider two extreme cases, i.e., $t - \mathrm{TP} + \mathrm{PD}_{ij}$ and $t + t' + \mathrm{PD}_{ij}$ to show that no node other than node i transmits on wavelength λ_m during the interval $t - \mathrm{TP} + \mathrm{PD}_{ij}$ to $t + t' + \mathrm{PD}_{ij}$
Case 1: There does not exist a node j that transmits on wavelength λ_m at $t - \mathrm{TP} + \mathrm{PD}_{ij}$.
Suppose node j transmits on wavelength λ_m at $t - \mathrm{TP} + \mathrm{PD}_{ij}$. This is possible only if node j has reserved wavelength λ_m during the interval $t - 2\mathrm{TP}$ to $t - \mathrm{TP}$, and the following conditions hold good:

$$\mathrm{CAT}_l[m] = 0, \forall l \in \{0, 1, \ldots, N-1\} \text{ at } t - 2\mathrm{TP} \tag{7}$$

$$\mathrm{CAT}_l[m] = 1, \forall l \in \{0, 1, \ldots, N-1\} \text{ at } t - \mathrm{TP} \tag{8}$$

But condition (8) contradicts condition (5). This means when node i has reserved wavelength λ_m during the interval $t - \mathrm{TP}$ to t, wavelength λ_m was not reserved by node j during the interval $t - 2\mathrm{TP}$ to $t - \mathrm{TP}$. Hence, node j can not transmit on wavelength λ_m at $t - \mathrm{TP} + \mathrm{PD}_{ij}$.
Case 2: There does not exist a node j that transmits on wavelength λ_m at $t + t' + \mathrm{PD}_{ij}$.
Suppose node j transmits on wavelength λ_m at $t + t' + \mathrm{PD}_{ij}$. For this to happen node j must reserve the wavelength λ_m during the interval $t + (\lfloor t'/\mathrm{TP} \rfloor)\mathrm{TP} - \mathrm{TP}$ to $t + (\lfloor t'/\mathrm{TP} \rfloor)\mathrm{TP}$. Node j can reserve, only when node i releases the wavelength λ_m. But node i will release wavelength λ_m at $t + (\lceil t' + \mathrm{TP} \rceil)\mathrm{TP} + \mathrm{TP}$, and node j can reserve wavelength λ_m only at $t + (\lceil t' + \mathrm{TP} \rceil)\mathrm{TP} + 2\mathrm{TP}$ (it takes a TP period for a node to reserve a wavelength). Therefore, node j can not transmit on wavelength λ_m at $t + t' + \mathrm{PD}_{ij}$. □

From cases 1 and 2 we conclude that wavelength collision never occurs (Enunciations 1–8).

4. Simulation results

In this section we simulated a 96 km ring network with five data-channels. The number of nodes considered for the simulation are 8, 12 and 16 to study the performance. We consider (i) the mean delay experienced by packets in milliseconds (ms), (ii) throughput in bits per second (bps), and (iii) wavelength utilization in percentage as the three performance metrics. To calculate the value of TP as mentioned in Section 2, we use the following quanta of values: $d = 96$ kms, $s = 1$ Gbps, $b = 1000$ bits, $v = 2 \times 10^8$ m/s. Computed value of TP $= 480,000 + N \times 1000$ bits.

Poisson distribution is widely used by authors for modeling data traffic. But, it is shown by many authors that Poisson distribution is not adequate for modeling local-area and wide area network traffic [16–18]. Present day traffic in Internet and in LANs is mostly bursty in nature [19]. Therefore, in our simulation we consider both distributions—Poisson and Pareto—for modeling data traffic. We used 5×10^5 packets, each of a fixed size of 1000 bits for simulation.

4.1. Simulation with data generated from poisson distribution

The mean arrival rate of packets at each node is assumed to be λ/s. The service rate at each node is fixed at μ packets per sec. The load at each node, in Erlang, is λ/μ. In our simulation we assume that all packets arriving at a node between the period the node makes the reservation request to the start of transmission are destined to a particular destination node. The destination node is selected from a uniform distribution.

First, we include the graph showing plots for load vs. mean delay experienced by packets in Fig. 5. The delay that we consider is the average delay experienced by packets at each node from the time of its arrival at the node to the start of its transmission. From Fig. 5 it is observed that the delay increases with load. However, the increase is slower at higher load. This is because at higher load the arrival rate of packets to a node is high and a large number of packets are accumulated at a given time. All these packets are transmitted once a circuit is established reducing their waiting period at the node.

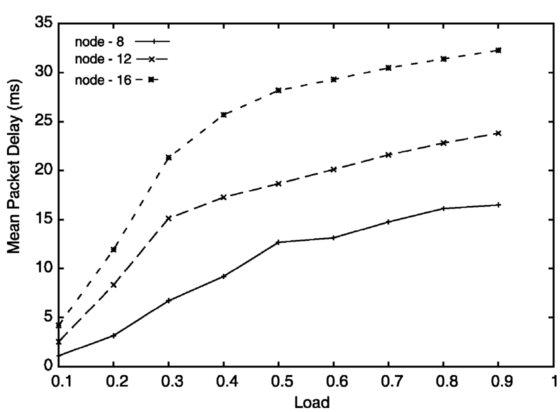

Fig. 5. Load (Erlang) vs. mean delay (ms) for Poisson distributed data.

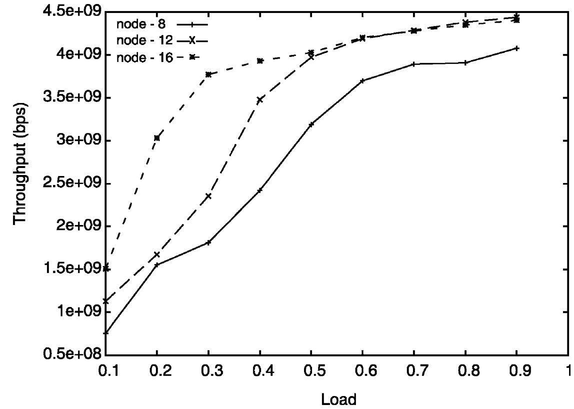

Fig. 6. Load (Erlang) vs. throughput (bps) for Poisson distributed data.

It is also observed that delay increases with increase in the number of nodes for a given load. The increase in the delay is due to the fixed number of data-channels. It is obvious that delay increases with increase in number nodes for a fixed number of data-channels.

Next, we include the plots for load vs. throughput in Fig. 6. We have plotted the throughput aggregated over all data-channels in the network. Throughput increases with increase in load. With increase in load, packets arrive at a node at high rate. As a result large number of packets are accumulated at a node in a given period and is transmitted once the circuit is established giving higher wavelength utilization and higher throughput. Throughput is proportional to wavelength utilization. The aggregated wavelength utilization of all data-channel is shown in Fig. 7. The symmetry in the nature of the plots in Figs. 6 and 7 confirms the behavior and shows the correctness of the simulation. As the wavelength utilization increases the throughput also increases. With increase in number of nodes and the load, the wavelength utilization increases but at a higher load and larger nodes the difference in wavelength utilization is almost negligible so is the throughput.

Thus, we observe that the mean-delay increases with load and number of nodes. However, the increase is marginal and the delay increases slowly with increase in nodes and

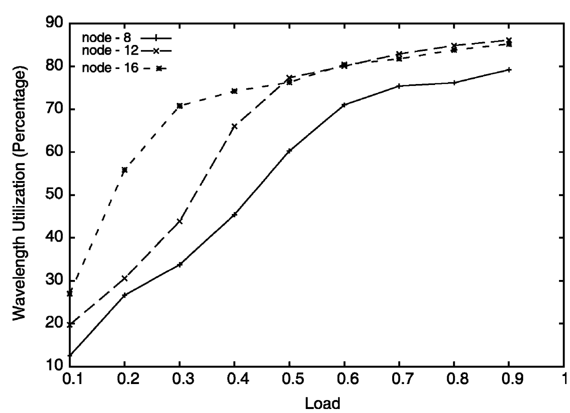

Fig. 7. Load (Erlang) vs. wavelength utilization (percentage) for Poisson distributed data.

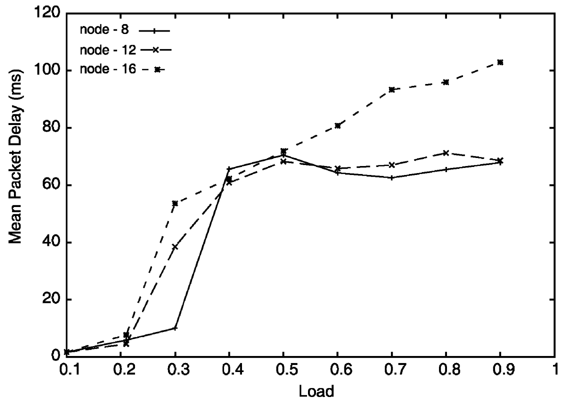

Fig. 8. Load (Erlang) vs. mean delay (ms) for bursty traffic.

the traffic. We can say that the performance degrades gracefully with increase in load and nodes. Regarding throughput and wavelength utilization, the increase is steep at lower load but it gets almost saturated at higher load.

4.2. Simulation with bursty traffic

Bursty traffic is generated at each node using $M/G/\infty$ model [19]. We consider Pareto ($\alpha = 1.1$) distributed burst length and Pareto ($\alpha = 1.1$) distribution for inter-arrival of burst. (We have taken a value of α as used by other researchers in their work). A uniform inter-arrival of packets within a burst is assumed. In our simulation we assume all bursts arriving at a node between the period the node makes the reservation request to the start of transmission are destined to a particular destination node. The destination node is selected from a uniform distribution.

The plot for load vs. mean delay is included in Fig. 8. It is observed that delay is low at lower loads, however, it increases with the load and gets almost saturated at higher loads. This phenomenon is expected from a bursty traffic and results due to the larger variance of burstiness. When the load is lower, the delay encountered are similar in nature to that of Poisson data. However, as the load increases, burstiness also increases, and the delay takes much larger values. The delay-plots (Fig. 8) do not have a monotonic

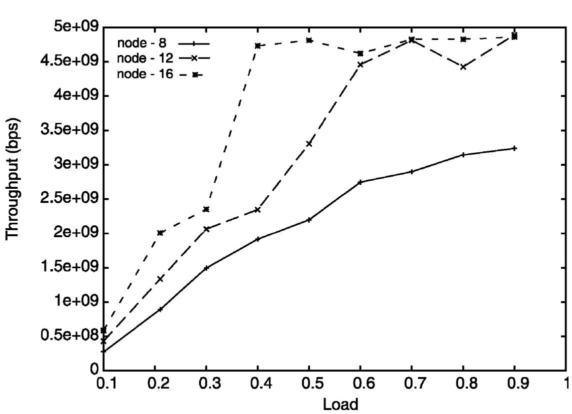

Fig. 9. Load (Erlang) vs. throughput (bps) for bursty traffic.

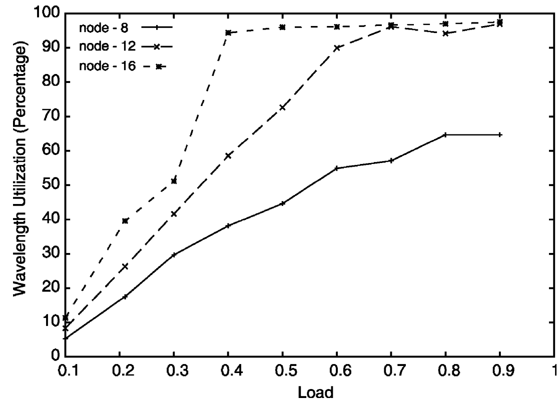

Fig. 10. Load (Erlang) vs. wavelength utilization (percentage) for bursty traffic.

increase as depicted for Poisson data (Fig. 5). This is an inherent nature of burstiness and depends on the degree of variability of the burst-sizes.

The plots for load vs. throughput are given in Fig. 9. It is observed that throughput increases with increase in load. This is because with increase in load higher number of packets are accumulated at a node for a given time period and once a circuit is established a large volume of data is transmitted giving higher throughput. Throughput is proportional to wavelength utilization, and this can be seen from the Figs. 9 and 10; plots in both the Figs. 9 and 10 have identical shapes.

Similar to delay vs. load plots, the throughput in terms of bps and wavelength utilization is poor at lower loads. However, it increases with increase in nodes and the traffic. Analogous to the delay behavior, there does not exist a monotonic increase; this is due to the degree of burstiness. Nonetheless, the plots gets almost saturated at higher loads and exhibit some degree of scalability.

4.3. Comparison

In this section we give a comparison for both types of traffic. We separately include the plots for an eight node network for load vs. mean delay in Fig. 11 and load vs.

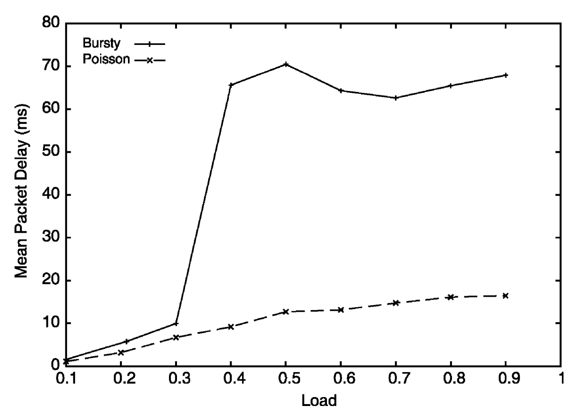

Fig. 11. Load (Erlang) vs. delay (ms) for data modeled by Poisson and Pareto distributions.

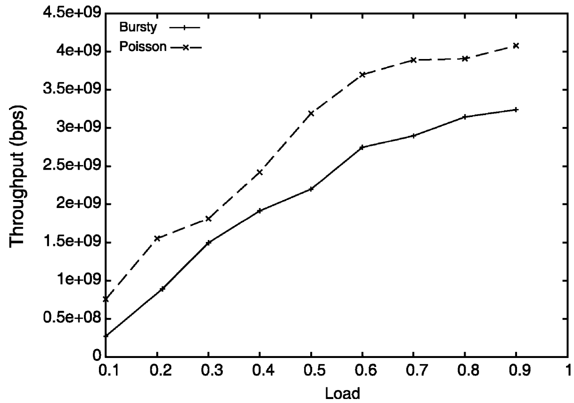

Fig. 12. Load (Erlang) vs. throughput (bps) for data modeled by Poisson and Pareto distributions.

throughput in Fig. 12. It is observed from the figures that the delay experienced by bursty traffic at lower load is identical to that of Poisson traffic. However, as the load increases, the delay gets much larger. As the load increases, burstiness also increases, and the delay takes much larger values. Delays experienced by bursty traffic are much larger than that by Poisson traffic. The delays at higher loads get almost saturated.

At lower loads, the throughput in terms of bps and wavelength utilization is lower for both Poisson and Pareto distributions. However, as the burstiness increases with increase in load, the throughput also increases.

Network response for a bursty traffic is an area of current research interest. Not much study has been done. This is an area to be further investigated through simulation as well as analytical study.

5. Conclusions

In this paper we have proposed a collision-free MAC protocol called RTR for an all-optical ring network. RTR is based on circuit switching and operates on three stages: reservation, transmission and release. It takes a token period for reservation and release. Once the reservation process is completed, source continues to transmit till the completion of data. Data transfer takes a single hop over the data-channel between the source and destination. RTR requires no synchronization among nodes; this is different from most of the MAC protocols available in the literature for optical ring network where synchronization among nodes is required. We have proved that destination collision and data channel collision does not occur in the RTR algorithm. Hence, no loss of data takes place. The node architecture is configured around a tunable transceiver and thus makes the scheme scalable.

We simulated for data-traffic modeled by Poisson and Pareto distributions. We observed that delay, throughput and wavelength utilization increases with increase in load and number of nodes in the network. For data traffic modeled by Poisson traffic, the increase is graceful. For data traffic modeled by Pareto distribution, the delay experienced at very low load is lower; this may be due to not much effect of burstiness experienced at very lower load. However, the delay increases significantly at normal load and are much higher than that of Poisson traffic. With increase in load, delay gets almost saturated within a range though it does not increase strictly in a monotonic manner. This may be explained by the degree of burstiness and the characteristics of Pareto distribution. There does not exist much analytical as well as simulation work related to Pareto distribution. This is an active research area.

The future work also includes extension of the work for priority among the nodes, and to provide desired QoS in the network.

Acknowledgements

Authors gratefully acknowledge discussions with R. Badrinath during the initial stages of this work. The authors would like to thank anonymous reviewer for his/her valuable comments and helpful suggestions which could greatly improve the quality.

References

[1] A. Ganz, Z. Koren, Performance and design evaluation of WDM stars, J. Lightwave Technol. 11 (1993) 358–366.

[2] F.J. Janniello, R. Ramaswami, D.G. Steinberg, A prototype circuit-switched multi-wavelength optical metropolitan area network, J. Lightwave Technol. 11 (1993) 777–782.

[3] F. Jia, B. Mukherjee, The receiver collision avoidance (rca) protocol for a single-hop WDM ligthwave network, J. Lightwave Technol. 11 (1993) 1053–1065.

[4] C.S. Li, M.S. Chen, F.F. Key, POPSMAC: a medium access protocol for packet switched passive optical networks using WDMA, J. Lightwave Technol. 11 (1993) 1066–1077.

[5] J.H. Laarhuis, A. Koonen, An efficient medium access control strategy for high-speed WDM multiaccess networks, J. Lightwave Technol. 11 (1993) 1078–1087.

[6] M.J. Spencer, M.A. Summerfield, WRAP: a medium access control protocol for wavelength-routed passive optical networks, J. Lightwave Technol. 18 (2000) 1657–1676.

[7] K. Bengi, H.R. van As, Efficient QoS support in a slotted multihop WDM metro ring, IEEE J. Selected Areas Commun. 20 (2002) 216–227.

[8] M.A. Marsan, A. Bianco, E. Leonardi, A. Morabito, F. Neri, All-optical WDM multi-rings with differentiated QoS, IEEE Commun. Magn. (1999) 58–66.

[9] M.A. Marsan, A. Bianco, E. Leonardi, F. Neri, S. Toniolo, An almost optimal macprotocol for all-optical multi-rings with tunable transmitters and fixed receivers, 1997, http://citeseer.nj.nec.com/marsan97almost.html.

[10] M.A. Marsan, A. Bianco, E.G. Abos, All-optical slotted WDM rings with partially tunable transmitters and receivers, 1996, http://citeseer.nj.nec.com/34986.html.

[11] J. Fransson, M. Johansson, M. Roughan, L. Andrew, M.A. Summerfield, design of a medium access control protocol for a WDMA/

TDMA photonic ring network, http://citeseer.nj.nec.com/487875.html.
[12] C.-K. Chan, L.-K. Chen, K.-W. Cheung, A fast channel-tunable optical transmitter for ultrahigh-speed all-optical time-division multiaccess networks, IEEE J. Selected Areas Commun. 14 (1996) 1052–1056.
[13] O.A. Lavrora, G. Rossi, D.J. Blumenthal, Rapid Tunable Transmitter with Large Number of ITUChannel Accessible in Less Than 5 ns, Proc. ECOC 2000, vol. 2, 2000, p. 169–170.
[14] B. Mason, G.A. Fish, S.P. Denbaars, L.A. Coldren, Widely tunable sampled grating DBR laser with integrated electroabsorption modulator, IEEE Photonic Technol. Lett. 11 (1999) 638–640.
[15] C.R. Doerr, C.H. Joyner, L.W. Stulz, 40-Wavelength rapidly digitally tunable laser, IEEE Photonic Technol. Lett. 11 (1999) 1348–1350.
[16] P. Danzig, S. Jamin, R. Caceres, D. Mitzel, D. Estrin, An empirical workload model for driving wide-area TCP/IP network simulation, Internetworking: Res. Exp. 3 (1992) 1–26.
[17] H. Fowler, W. Leland, Local area network traffic characteristics, with implications for broadband network congestion management, IEEE J Selected Areas Commun. 9 (1991) 1139–1149.
[18] R. Jain, S. Routhier, Packets trains-measurements and a new model for computer network traffic, IEEE J. Selected Areas Commun. 4 (1986) 986–995.
[19] V. Paxson, S. Floyd, Wide area traffic: the failure of Poisson modeling, IEEE/ACM Trans. Networking 3 (1995) 226–244.
[20] R. Ramaswami, K.N. Sivarajan, Optical networks: a practical perspective, Morgan Kaufmann, 1998.