**Fully Distributed Work-Conserving MAC Protocols for Opportunistic Optical Hyperchannels**

The MIT Faculty has made this article openly available. Please share how this access benefits you. Your story matters.

| **Citation** | Jing Chen et al. “Fully distributed work-conserving MAC protocols for opportunistic optical hyperchannels.” Communications, IEEE Transactions on 57.12 (2009): 3691-3702. © 2009 Institute of Electrical and Electronics Engineers |
| **As Published** | http://dx.doi.org/10.1109/TCOMM.2009.12.080370 |
| **Publisher** | Institute of Electrical and Electronics Engineers |
| **Version** | Final published version |
| **Accessed** | Mon Jan 11 13:02:42 EST 2016 |
| **Citable Link** | http://hdl.handle.net/1721.1/52406 |
| **Terms of Use** | Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use. |
| **Detailed Terms** | |
Abstract—Light-trail is proposed as a candidate to carry IP traffic over wavelength division multiplexing (WDM) optical networks given its capability of accommodating multi-granularity traffic by time-division multiplexing (TDM). Light-trail’s unidirectional shared-media multicast nature makes it hard to implement distributed access control and restricts that at most one packet transmission can take place at any time. Recently, opportunistic optical hyperchannel was proposed to improve light-trail by permitting easy distributed access control. In this paper, we propose a set of distributed access control protocols, namely, 1-persistent protocols, for opportunistic optical hyperchannels to maximize the throughput and provide fair service among contending nodes by taking their inherent advantage of adaptive space-division multiplexing (SDM). We also point out a possible generalization of opportunistic optical hyperchannels by removing the restriction of linear structure, and demonstrate possible applications of such a generalization.

Index Terms—Optical network, WDM, TDM, SDM, CSMA, MAC, light-trail, opportunistic optical hyperchannel.

I. INTRODUCTION

Optical Wavelength Division Multiplexing (WDM) technology has been considered as the basis of the next generation wired Internet. In the past decades, great efforts have been made on elevating the wavelength utilization in WDM networks with off-the-shelf optical components. Wavelength utilization is limited in WDM networks due to the following two mismatches: (1) the mismatch between high bandwidth and relatively long delay of optical switch configuration; (2) the mismatch between high bandwidth per wavelength can provide and relatively low bandwidth requirement of a single connection. Correspondently, the work proposed to improve wavelength utilization mainly focus on either or both of the following two perspectives: (1) reducing the frequency of optical switch configuration; (2) multiplexing/grooming traffic into one optical path.

Optical packet switching (OPS) [1] and optical burst switching (OBS) [2] aim to resolve the second mismatch. However, such solutions require optical addressing and buffering, high-speed optical routing and switching, which are technologies still in nascent stages of development. Traffic grooming [3] on a virtual topology which consists of a set of lightpaths [4] aims to resolve both mismatches. Traffic grooming allows two connections to share the bandwidth of a lightpath according to time-division multiplexing (TDM) which can improve the inefficiency of lightpaths. However, in traffic grooming, two connections can share the bandwidth of a lightpath only when the start node and the end node of such lightpath are on the optical paths of both connections, which limits the flexibility of multiplexing. Light-trail [5] generalizes lightpath with improved scalability by allowing time-division multiplexing (TDM) bandwidth utilization between any two nodes within a light-trail. Light-trail is a multi-point to multi-point solution and can be viewed as a generalization of the sub-wavelength solutions proposed in the literature, e.g., super-lightpath [6] which provides point to multi-point communication and TWIN [7] which is applicable to multi-point to point communication.

In a light-trail, optical shutters at the start node and the end node are set to be in OFF state such that the start node can only send data to downstream nodes and the end node can only receive data from upstream nodes. The optical shutters at intermediate nodes are configured to be in ON state to allow the traffic from upstream nodes to be passed to downstream nodes. Once a light-trail is established, communications between pairs of nodes are accomplished in a time-multiplexing mode without optical switching. Therefore, only one packet can be transmitted at a time, which limits the system throughput.

With the recent development on ultra fast optical shutter [9], we proposed an improved light-trail architecture in [10], called opportunistic optical hyperchannel and a generalized WDM mesh network that supports opportunistic optical hyperchannels. Fig. 1 illustrates the architecture of a (unidirectional) opportunistic optical hyperchannel. In such an opportunistic optical hyperchannel, optical shutters at the start node and the end node are set to be in OFF state as in a light-
trail. The optical shutters at intermediate nodes can be set to ON or OFF state dynamically. Any intermediate node which is about to transmit data will set its optical shutter to be OFF, and “cut” the opportunistic optical hyperchannel into two segments. Therefore, the traffic from the upstream nodes will be intercepted at the current node and the current node is allowed to transmit its data without collision. If the packet intercepted from an upstream node already passed its destination, its transmission has finished and it will not consume the system bandwidth anymore. Thus, opportunistic hyperchannel can potentially improve the throughput given the fact that multiple simultaneous transmissions are possible as long as those connections are restricted to non-overlapped segments of the hyperchannel. We call this feature adaptive space-division multiplexing (adaptive SDM).

To maximize the throughput of an opportunistic optical hyperchannel, a medium access control (MAC) protocol is required to coordinate the access of the hyperchannel, which is the main focus of this paper. The goal of our investigation is to develop simple, easy-to-implement “aggressive” fully distributed MAC protocols that take the advantage of adaptive SDM to not only amplify the hyperchannel throughput with respect to the opportunistic optical hyperchannel bandwidth, but also achieve good fairness. The contributions of this paper are summarized as follows:

- We propose simple fully distributed and work-conserving dynamic scheduling protocols, namely 1-persistent protocols, for opportunistic optical hyperchannels of a bus structure, which aim at achieving maximum amplified system throughput and providing good fairness.
- We analyze our protocols by deriving lower bound and upper bound for the throughput amplification of running 1-persistent protocols on an opportunistic optical hyperchannel in comparison with any work-conserving CSMA protocol running on a light-trail. We also provides an upper bound for the relative worst-case packet delay of our protocols compared with GRR. According to our analysis, though upstream nodes may experience longer delay than downstream nodes if 1-persistent protocols are used, no node will suffer from starvation. Upper bound of average buffer size at an intermediate node is also derived. We present simulation results, and show that our simulations validate our analysis.
- We extend the concept of opportunistic optical hyperchannel, which inherently has a linear (bus) topology, into opportunistic optical hyperchannels with non-linear topologies embedded in an optical mesh network. Due to the high bandwidth that one wavelength can provide, non-linear opportunistic hyperchannel is a solution to provide high bandwidth utilization. In particular, we show non-linear opportunistic optical hyperchannels of tree topologies, namely gathering tree and multicast-and-select tree. We present an implementation of 1-persistent protocols in a gathering tree opportunistic hyperchannel and applications of a pair of gathering tree and multicast-and-select trees.

The rest of the paper is organized as follows. The related work is summarized in Section II. In Section III, we review the structure of an opportunistic optical hyperchannel. In Section IV, we describe the features of a class of 1-persistent MAC protocols. In Section V, we give the performance analysis of the proposed 1-persistent protocols. We present our simulation results in Section VI. In Section VII, we discuss non-linear opportunistic optical hyperchannels (a generalization of opportunistic optical hyperchannel), its associated 1-persistent protocols and its applications. Section VII concludes the paper.

II. RELATED WORK

As pointed out in [10], the capability of “intercepting” packets from upstream nodes in an opportunistic optical hyperchannel makes it possible to design a family of distributed MAC protocols. In [10], we proposed a distributed QoS-assuring MAC protocol, called minSrcRR protocol. In minSrcRR protocol, each node selects one of its upstream nodes as the reference node and adjust its accumulated quota by observing the traffic coming from its reference upstream node. When it has enough quota, it will set its optical shutter to be in OFF state and start the local transmission. Thus, QoS is assured. Since some nodes may hold their transmission until they have enough quota, opportunistic optical hyperchannel may not achieve its capacity of high throughput in minSrcRR protocol. In this paper, our focus is to design distributed protocols to maximize the throughput of an opportunistic optical hyperchannel and provide good fairness. Besides comparing the protocol proposed in this paper with minSrcRR protocol, we are also interested in comparing our protocol with other MAC protocols in the generic family of opportunistic optical hyperchannel, unidirectional optical bus in terms of throughput and fairness.

Basically, there are three kinds of architectures for optical buses: folded, dual, and unidirectional. A dual bus consists of two unidirectional buses with signals propagating in opposite directions. All-to-all communications are supported in both folded and dual optical bus, while only upstream nodes can transmit packets to downstream nodes in a unidirectional optical bus. Without incorporating a “feedback” mechanism, upstream nodes has no knowledge of what is happening on the downstream segment of a unidirectional bus. Thus, most of the previous studies were focused on the architecture of either folded or dual optical bus and were carrier sense multiple access (CSMA) protocols. These protocols use collision detection to avoid packet transmission collisions. Representative CSMA protocols for optical folded and dual buses can be found in [11]–[16].

CSMA protocols for unidirectional optical buses [17]–[22] can be classified into two categories: CSMA protocols maximizing system throughput and CSMA protocols providing fairness among contending nodes. A representative CSMA protocol maximizing system throughput in optical buses is called priority based CSMA protocol [17]. In a priority based CSMA protocol, whenever the downstream nodes sense data from the upstream nodes, the downstream nodes will interrupt their transmissions and free the bus for the transmissions from the upstream nodes. Priority based CSMA protocol can obviously achieve high throughput at the cost of severe unfairness to the downstream nodes. A representative CSMA protocol providing fairness in a unidirectional optical bus is
The opportunistic optical hyperchannel can only transmit and receive packets, respectively. The packets to downstream nodes, whereas the start node and end node can receive packets from upstream nodes and likewise transmit packets to upstream nodes. However, \( p_i \)-persistent CSMA protocols also have drawbacks: (1) Obviously, any \( p_i \)-persistent CSMA protocol is not work-conserving; i.e. the channel may be idle with waiting packets. (2) In addition, packets transmitting collision cannot be completely avoided, which wastes channel bandwidth. (3) The responsiveness of \( p_i \)-persistent CSMA protocols is slow. To achieve desired performance, \( p_i \)'s have to be computed beforehand for specific traffic pattern and/or service requirement. Even for a small traffic change in one node, \( p_i \)'s have to be recalculated and nodes have to be notified of new \( p_i \)'s. For dynamic traffic and/or service requirements, this process is not only wasting channel bandwidth, but also slow in response to changes of network conditions.

The work proposed in [22] can provide fairness in a unidirectional optical bus where each destination receives information in a fixed wavelength and each source has a tunable transmitter. Each node maintains a Virtual Output Queue (VOQ) for each wavelength and determines to which wavelength it should tune such that collision will be reduced and fairness can be provided. When there is only one wavelength in an unidirectional optical bus, the proposed work is equivalent to priority-based CSMA protocol.

### III. System Model

In this section, we review the architecture of opportunistic optical hyperchannel proposed in [10]. In an \((n+1)\)-node opportunistic optical hyperchannel, the nodes are labeled \(N_0, N_1, \cdots, N_n\) with \(N_0\) and \(N_n\) as the start node and end node, respectively, as shown in Fig. 1. Each intermediate node can receive packets from upstream nodes and likewise transmit packets to downstream nodes, whereas the start node and end node can only transmit and receive packets, respectively. The architecture of an opportunistic hyperchannel is different from the architecture of a light-trail in the following two aspects:

- **Controllable Optical Shutter.** In an opportunistic optical hyperchannel, each intermediate node can control its optical shutter through the control component according to its traffic dynamically. When a node’s optical shutter is set to be in ON state, it allows the traffic from its upstream nodes to continue. It can also set its optical shutter to be in OFF state and thus send its own traffic.

- **Electronic Buffer.** When the optical shutter of an intermediate node is in OFF state, packets transmitted from upstream nodes are electronically buffered, which may be forwarded later if the destinations are the downstream nodes. Since each node in any optical bus is capable of transmitting and receiving, such Optical-Electro-Optical (OEO) conversion does not incur any extra hardware cost. A packet may be intercepted before reaching its destination only when a downstream node has traffic to send at the same time slot. In such cases, the process of receiving packets from the upstream nodes takes place with the process of transmitting a packet generated by the downstream node simultaneously. In other words, at least one packet from the downstream node will reach its destination. Thus, OEO conversion will not affect bandwidth utilization. In terms of the capacity requirement of electronic buffer, as we will show shortly, the proposed 1-persistent protocol will not require too large electronic buffer when the total traffic load of all connections in an opportunistic optical hyperchannel is less than the wavelength capacity.

Since each intermediate node \(N_k\) in an opportunistic hyperchannel can dynamically control its optical shutter to be in ON or OFF states, it is equivalent to that each intermediate node \(N_k\) controls a \(2 \times 1\) optical switch \(S_k\), referred to as a continue-add (CA) switch. Each CA switch consists of an optical shutter, a combiner, and a transmitter as shown in Fig. 2. \(S_k\) operates in two states: drop-and-continue state (DC state), in which the optical shutter is in ON state, and drop-and-add state (DA state), in which the optical shutter is in OFF state. When \(S_k\) is set to be in DC state, the signal from the upstream nodes can drop a copy at node \(N_k\) and pass \(S_k\) simultaneously. When \(S_k\) is set to be in DA state, it intercepts the signal from the upstream nodes for later processing and transmits packets from local buffers of \(N_k\) without collision.

### IV. 1-Persistent MAC Protocols

In this section, we define 1-persistent MAC protocols for an \((n+1)\)-node opportunistic hyperchannel, and propose two simple such protocols, and discuss related implementation issues.

A generic 1-persistent protocol work as follows. Each intermediate node \(N_k\), \(0 < k < n\), maintains two queues \(q_k^0\) and \(q_k^k\). The packets from the upstream nodes of \(N_k\) are always dropped at node \(N_k\), some of which will be added to \(q_k^0\) depending on the state of \(S_k\). The locally generated packets are stored in \(q_k^k\). If there is no locally generated packet in \(q_k^k\), \(S_k\) remains in DC state. The packet in \(q_k^0\) or \(q_k^k\) of \(N_k\) will trigger \(S_k\) to be changed from DC state to DA state. We define that a 1-persistent protocol for an opportunistic hyperchannel is a
fully distributed protocol such that any $N_k$, which is not the end node, transmits a packet whenever its $q_{k}^{0} \neq \emptyset$ or $q_{k}^{k} \neq \emptyset$. Note that in this definition persistent packet transmission is the key condition, where the usage of the two queues, $q_{k}^{0}$ and $q_{k}^{k}$, is not essential. A node running a particular 1-persistent protocol may have one queue (merging $q_{k}^{0}$ and $q_{k}^{k}$ into one), or even have more than two queues. Using a 1-persistent protocol, parallel packet transmissions without collision can be achieved by dynamically setting $S_{k}$ to DA or DC state.

In the rest of this paper, we assume that packets have a fixed length, and their transmissions are slotted. Suppose that it takes $t_{g}$ time to switch from one state to the other at a $2 \times 1$ optical switch. Let the transmission delay of a packet be $T$. With the advance of the switching technology, $t_{g}$ can be expected to be much smaller than $T$. The timing of state change of a CA switch is controlled as follows. At each node, a $t_{g}$ time latency, referred to as “guard time”, is inserted between the transmission of two consecutive packets. Thus, the period of each time slot will be $T + t_{g}$. Time slot $i$ is the time interval $(i - 1) \cdot (T + t_{g}) + T$. If no more packets in either queue, at the time period of $(i - 1) \cdot (T + t_{g}) + T$, $S_{k}$ is switched from DA state to DC state. Clearly, node is able to transmit a packet and also change its state in one time slot. This approach divides each time slot into two portions, one portion for packet transmission and one portion for the potential state change. During the time portion of the state change, no data transmission is ongoing, thus incomplete packet transmission is avoided. This approach is very efficient when $t_{g} \ll T$, which is highly feasible nowadays for optical shutter [9].

To implement such a 1-persistent protocol, the following technical issue has to be addressed:

- Since each packet from the upstream nodes will be dropped at $N_{k}$, $N_{k}$ needs to determine which packets should go to $q_{k}^{0}$ to be relayed to downstream nodes.
- When $S_{k}$ is in DA state, which packet from either $q_{k}^{0}$ or from $q_{k}^{k}$ should be relayed to downstream nodes to assure fairness.

In the rest of this section, we present a procedure for each node to determine whether a packet from the upstream nodes should be buffered or not, followed by introducing the implementation of two particular 1-persistent protocols at each node. At the end of this section, we summarize the properties of the proposed protocols.

### A. Queuing Decisions for Received Packets

The packets dropped at $N_{k}$ include: (i) packets from $N_{k}$’s upstream nodes that have passed $S_{k}$ when $S_{k}$ is in DC state. The destinations of these packets can be the upstream nodes of $N_{k}$, $N_{k}$ itself, or the downstream nodes of $N_{k}$. (ii) packets from $N_{k}$’s upstream nodes that are stored in $N_{k}$ when $S_{k}$ is in DA state. For these packets, some of them should be discarded, some of them should enter queue $q_{k}^{k}$ for later transmission, and some of them should be taken (locally “consumed”) if the destination of the packets is $N_{k}$.

With respect to a received packet $p$, at the end of a time slot, $N_{k}$ will process $p$ according to the following procedure:

- **Case 1:** if $S_{k}$ is in DC state and the destination of $p$ is a downstream node, then discard $p$; otherwise,
- **Case 2:** if the destination of $p$ is an upstream node of $N_{k}$, then discard $p$; otherwise,
- **Case 3:** if the destination of $p$ is $N_{k}$, then take $p$; otherwise,
- **Case 4:** put $p$ at the end of $q_{k}^{k}$.

### B. Simple 1-Persistent Protocols

According to the definition of 1-persistent protocols, many 1-persistent protocols can be designed. Our goal is to design fair 1-persistent protocols that are easy to implement.

We assume that all nodes share the same system clock and each packet is associated with two attributes, timestamp $t_{stamp}(p)$ and source $s(p)$, which are respectively the system time at which $p$ is originally generated and the node index that packet $p$ is originally generated, in addition to the destination of $p$. Let $p_{i,k}$ be the packet originally generated at $N_{k}$ at time slot $i$. Define that packet $p_{i',k'} < p_{i,k}$ if and only if $i' < i$ or $i' = i$ and $k' > k$. We say that packet $p_{i',k'}$’s timestamp is smaller than the timestamp of $p_{i,k}$ if $p_{i',k'} < p_{i,k}$. Clearly, there is a total order on the timestamps of all packets. Our 1-persistent protocol makes a decision for the next time slot after the first $T$ time interval of the current slot as follows:

- When both $q_{k}^{0}$ and $q_{k}^{k}$ are empty, $S_{k}$ is set in DC state.
- When there are packets in either $q_{k}^{0}$ or $q_{k}^{k}$, $S_{k}$ is changed to DA state. Among the head packets of $q_{k}^{0}$ and $q_{k}^{k}$, the one with the smaller $t_{stamp}$ will be transmitted in the next time slot.

Implementing $q_{k}^{k}$ as a First-In-First-Out (FIFO) queue, the packets in $q_{k}^{k}$ are inherently sorted according to increasing timestamp order. However, if $q_{k}^{0}$ is a FIFO queue its packets might not be in sorted order. To see this, consider the following situation. All packets are destined for the end node $N_{n}$. Before time slot $i$, all nodes $N_{0}$, $N_{1}$, $N_{2}$ and $N_{3}$ do not have any packet. During slot $i$, $N_{0}$ generates a packet $p_{i,0}$, $N_{1}$ generates $p_{i,1}$ and $N_{3}$ generates $p_{i,3}$. During slot $i + 1$, $N_{2}$ generates a packet $p_{i+1,2}$, $N_{3}$ generates $p_{i+1,3}$. In the following slots, none of $N_{0}$ through $N_{3}$ generates a packet. Let us observe $q_{0}^{0}$. In slot $i + 1$, $p_{i+1,3}$ is transmitted to $N_{n}$, $p_{i+1,3}$ is intercepted by $N_{3}$ and buffered in $q_{3}^{0}$, and $p_{i,0}$ is intercepted by $N_{1}$. In slot $i + 2$, $p_{i+1,2}$ in $q_{0}^{0}$ is transmitted to $N_{n}$, $p_{i+1,2}$ is intercepted by $N_{3}$ and buffered in $q_{3}^{0}$, and $p_{i,0}$ is intercepted by $N_{2}$. In slot $i + 3$, $p_{i+1,3}$ in $q_{0}^{0}$ is transmitted to $N_{n}$, $p_{i,0}$ is intercepted by $N_{3}$ and buffered in $q_{3}^{0}$. Clearly, $p_{i+1,2}$ entered $q_{3}^{0}$ before $p_{i,0}$. Thus, packets in $q_{0}^{0}$ are out of order.

Given the above property, in order to achieve better fairness, we need to sort the packets in $q_{k}^{0}$ for $1 \leq k \leq n$ in the non-decreasing timestamp order. With the sorted packets in
Among all packets available in CHEN et al., the min-heap queue for software min-heap queue is used to implement \( t \) large, it might not be feasible to maintain the min-heap queue if we store packets at the heads of \( q_k^0 \) and \( q_k^1 \). A priority queue implemented with a min-heap for \( q_k^0 \) and a first-in-first-out (FIFO) queue for \( q_k^1 \) can well achieve the fairness. If a software min-heap queue is used to implement \( q_k^0 \), the time complexity for maintaining \( q_k^0 \) is \( O(\log m_k) \) where \( m_k \) is the number of packets in \( q_k^0 \). The 1-persistent protocol using the min-heap queue for \( q_k^0 \) (1 \( \leq k \leq n \)) is referred to as 1PER/1 in this paper.

For 1PER/1, maintaining the min-heap queue and comparing the timestamp of two head packets in \( q_k^0 \) and \( q_k^1 \) must be accomplished within \( t_g \). However, if the queue size is too large, it might not be feasible to maintain the min-heap queue within \( t_g \). There are three options for solving this problem: (a) A hardware priority queue of \( O(1) \)-time insertion and delete-min operations introduced in [23] can be used. (b) We may apply pipeline principle to implement insertion/delete-min operations in one slot time \( T \) in the following way: during the \( T \) interval of slot \( i \), packets stored before the start of slot \( i \) are considered for transmission in slot \( i + 1 \). Thus, min-heap maintenance time can be extended to \( T + t_g \). This method may result in one more time slot delay for the newly arrived packet without affecting the overall performance of the opportunistic optical hyperchannel. (c) Combination of (a) and (b).

We also propose an alternative protocol in which both \( q_k^0 \) and \( q_k^1 \) are FIFO queues. This protocol is referred to as 1PER/2. In 1PER/2, operations on \( q_k^0 \) and \( q_k^1 \) have \( O(1) \) time complexity. Through the simulation, we can see that fairness that 1PER/2 can achieve is almost the same as that of 1PER/1. This experimental result has important practical implication: using FIFOs for both of \( q_k^0 \) and \( q_k^1 \) at intermediate nodes can remove priority queue insertion/delete-min operations.

C. Properties of 1-Persistent Protocols

Our generic 1-persistent MAC protocol exhibits the following properties:

- Work-Conserving and Fully Distributed. Each intermediate node \( N_k \), \( 0 < k < n \), persistently transmits a packet to its downstream nodes whenever either \( q_k^0 \) or \( q_k^1 \) is not empty. The decisions of when to transmit and what to transmit are made completely locally.
- Collision Free. Suppose that two nodes \( N_k \) and \( N_j \) transmit a packet simultaneously at time slot \( i \) and \( k < j \). Then both \( N_k \) and \( N_j \) must be in DA state. The packet transmitted by \( N_k \) cannot pass \( S_k \) at time slot \( i \); thus, the packet transmitted by \( N_k \) will never collide with the packet transmitted by \( N_j \). We would like to note that packet collision is avoided without carrier sensing.
- Fair. If each intermediate node \( N_k \) selects a packet with the smallest timestamp for transmission when it is in DA state, fairness is enforced among nodes.
- High Efficiency. Once a packet is transmitted by a node \( N_k \), the packet is transmitted in optical domain as far as possible until it encounters a downstream node \( N_j \) whose switch \( S_l \) is in DA state and has packets to transmit.

Compared with any carrier sense multiple access protocol running on an optical bus, 1-persistent protocol running on an opportunistic hyperchannel does not require collision detection hardware, which is hard to implement for optical media. Instead, high speed shutters are needed. OE and EO converters, receivers and transmitters used in opportunistic hyperchannel do not incur extra cost in comparison with any optical bus because they can not be avoided.

V. PERFORMANCE ANALYSIS

In this section, we analyze the performance of 1PER/1 by calculating a lower and upper bound for the amplification ratio of the throughput, deriving a relative worst-case upper bound for packet delays in comparison with an ideal fair packet scheduler, and estimating average buffer size at any intermediate node.

A. Throughput Amplification

We compare 1-persistent protocols running on an opportunistic hyperchannel and work-conserving protocols running on a light-trail in terms of their throughput. To focus on throughput, we assume that each node has unlimited amount of buffer space. We define the throughput \( TP \) of a communication channel, which is either a light-trail (optical bus) or an opportunistic hyperchannel in our context, as the ratio of packets successfully delivered to their destinations.

Let \( TP_L \) and \( TP_H \) be the throughput (packet delivering rate) of a light-trail running a work-conserving protocol and the throughput of an opportunistic hyperchannel running a 1-persistent protocol, respectively. We define the throughput amplification factor \( F_{TA} = \frac{TP_H}{TP_L} \). Clearly, \( F_{TA} \geq 1 \). If the channel load is less than 100% of the channel bandwidth, the throughput of a light-trail and an opportunistic hyperchannel running work-conserving protocols are the same, which is 100% of the channel bandwidth. When the channel load is larger than channel bandwidth, opportunistic hyperchannel exhibits an amplification of throughput compared with a light-trail, because the opportunistic channel can be partitioned into multiple segments, and multiple packets can reach their destinations in one time slot.

We present a lower bound and an upper bound for the throughput amplification factor \( F_{TA} \) of 1-persistent protocols running on an opportunistic hyperchannel with respect to any work-conserving protocol running on the corresponding light-trail.

In an \( (n + 1) \)-node opportunistic hyperchannel with nodes \( N_0, N_1, \ldots, N_n \), suppose that traffic arrival process at each node follows Poisson distribution with arrival rate of \( \lambda_i \) for \( 0 \leq i \leq n - 1 \).

1) Lower Bound: Among those concurrent transmissions in an opportunistic hyperchannel during one time slot, some of them may have reached their destinations in the same time slot. If we only consider such transmissions when we calculate system throughput, we can achieve the lower bound of the system throughput.

In an \( (n + 1) \)-node opportunistic hyperchannel with nodes \( N_0, N_1, \ldots, N_n \), \( m \in \{1, 2, \ldots, n\} \) packets may be transmitted and reach their destinations simultaneously in a time
For the case of \( m = 1 \), the probability that one packet is transmitted in the system is \( \text{prob}_1 = \lambda \cdot e^{-\lambda} \) since the system can be regarded as having one Poisson arrival process with parameter \( \lambda = \sum_{j=0}^{m-1} \lambda_j \).

For the cases that \( m \geq 2 \), let \( \text{Com}_m \) be the set of all possible non-overlapping segments of an opportunistic hyperchannel that results in \( m \) packets reaching their destinations. Without loss of generality, let \( \{(s_1^1, d_1^1), (s_2^2, d_2^2), \ldots, (s_m^m, d_m^m)\} \) be \( i \)-th element in \( \text{Com}_m \), where \( (s_i^i, d_i^i) \) is the source-destination pair of its \( i \)-th segment in the opportunistic hyperchannel. Let \( \text{prob}_i^m \) be the probability of the \( i \)-th element in \( \text{Com}_m \). Suppose that the destinations of all packets at each node are uniformly distributed among all downstream nodes of \( N_j \). Thus, \( \lambda_{j,k} = \frac{1}{n} \) is the arrival traffic parameter generated at node \( N_j \) and destined for a downstream node \( N_k \). In a Poisson distribution, the probability that there are exactly \( s \) occurrences (\( s \) being a non-negative integer, \( s = 0, 1, 2, \ldots \)) is equal to \( \frac{\lambda^s}{s!} \). Therefore, the probability that one packet is generated at node \( N_j \) for destination node \( N_k \) is \( \lambda_{j,k} \cdot e^{-\lambda_{j,k}} \).

Thus, \( \text{prob}_i^m \) can be calculated as follows:

\[
\text{prob}_i^m = \prod_{i=1}^{m} \lambda_{s_i^i,d_i^i} \cdot e^{-\lambda_{s_i^i,d_i^i}}, \text{ for } m \geq 2
\]  

In Equation (1), \( \text{prob}_2 \) includes the concurrencies of \( m \geq 2 \) cases; \( \text{prob}_3 \) includes the concurrencies of \( m \geq 3 \) cases; and so on. Therefore, in the throughput calculation, we count 2 packets in \( \text{prob}_2 \) and 1 packet in \( \text{prob}_3 \) when \( m > 2 \). Thus, we get a lower bound \( F_{TA}^m \) of the throughput amplification factor as the following equation:

\[
F_{TA}^m = \frac{1}{\text{prob}_1} + \sum_{i=1}^{\text{Com}_m} \frac{\text{prob}_2}{\text{prob}_1} \cdot \sum_{s=0}^{\text{Com}_m} \frac{\text{prob}_3}{\text{prob}_1} \cdot \cdots \cdot \sum_{s^{m-1}}^{\text{Com}_m} \text{prob}_m \cdot \min\{1, \sum_{j=0}^{m-1} \lambda_j\}
\]  

The numerator of Equation (2) is the packet delivering rate of an \((n + 1)\)-node opportunistic hyperchannel, whereas the denominator is the packet delivering rate of a corresponding light-trail.

2) Upper Bound: An intermediate node \( N_j \) contributes to throughput amplification if its CA switch \( S_j \) is in DA state, and \( N_j \) receives a packet \( p' \) whose destination is not a downstream node of \( N_j \). Though the destination of \( p' \) can be any upstream node of \( N_j \), we only consider the case that the destination of \( p' \) is \( N_j \) to avoid duplicated calculation. This is because the packets destined for \( N_{j-1} \) are already counted at node \( N_{j-1} \).

The average number of packets arrived from upstream nodes and destined for node \( N_j \) is \( \sum_{k=0}^{j-1} \frac{\lambda_k}{(n-k)} \). So the maximum average number of packets contributed by node \( N_j \) to the throughput amplification is no larger than \( \min\{\lambda_j, \sum_{k=0}^{j-1} \frac{\lambda_k}{(n-k)}\} \).

Thus, we can use the following formula to calculate an upper bound \( F_{TA}^+ \) of the throughput amplification factor:

\[
F_{TA}^+ = \frac{\sum_{n=0}^{n-1} \min\{\lambda_j, \sum_{k=0}^{j-1} \frac{\lambda_k}{(n-k)}\} + \min\{1, \sum_{j=0}^{n-1} \lambda_j\}}{\min\{1, \sum_{j=0}^{n-1} \lambda_j\}}
\]  

B. Relative Maximum Packet Delay

To find an upper bound for the relative maximum packet delay, we assume that the destinations of all packets are the same, the end node. We compare our 1-persistent protocols with an ideal protocol, General Round Robin (GRR), which enforces packet \( p_{r,k} \) reaching the end node before packet \( p_{i,k} \) if and only if \( p_{r,k} < p_{i,k} \). Conceptually, GRR transmits packets to the end node in rounds, with each round consisting of packets generated at the same time, in the increasing order of packet generating times, and transmits packets in the same round in the decreasing order of node indices. Clearly, GRR is a fair algorithm which cannot be implemented distributively without knowing the arriving order of packets. We analyze the performance of 1PER/1 by comparing them with algorithm GRR.

Let \( t_{GRR}(p), t_{1PER/1}(p) \) be the time at which packet \( p \) reaches the end node in an \((n + 1)\)-node opportunistic hyperchannel according to the GRR algorithm and the 1PER/1 protocol respectively. Define the delay of packet \( p \) using 1PER/1 with respect to GRR as \( D_{1}^{1}(p) = t_{1PER/1}(p) - t_{GRR}(p) \). Note that \( D_{1}^{1}(p) \) can be negative, when \( p \) reaches the end node using 1PER/1 earlier than the time it reaches the end node using GRR. Define

\[
D_{n}^{1*} = \max_{\text{all packets } p} \{D_{1}^{1}(p)\},
\]

which is the worst-case delay upper bound for the 1PER/1 protocol compared with the GRR algorithm.

**Theorem 1:** For 1PER/1, \( D_{n}^{1*} \leq (n - 1) \cdot (T + t_g) \).

**Proof:** Consider a packet \( p_{i,k} \) where \( 1 \leq k \leq n \). Let \( A_{i,k} \) be the set of packets that are in the system when \( p_{i,k} \) is generated with \( p' < p_{i,k} \) for \( p' \in A_{i,k} \). Given the work-conserving property of the 1PER protocol, there must also be \( |A_{i,k}| \) packets in the system when \( p_{i,k} \) is generated no matter which scheduling algorithm is applied. Clearly, in the GRR algorithm, packet \( p_{i,k} \) will have a delay of \( |A_{i,k}| \cdot (T + t_g) \).

Suppose that packet \( p_{i,k} \) reaches the destination at time slot \( t \) under 1-persistent 1PER/1. Because the system will keep busy as long as there are packets in the system, there will be one packet reaching destination at any time period \( i' \) for \( i' = i + 1, \ldots, t \). Among the packets arrived at the destination during the period from time slot \( i \) to time slot \( t \), if there is a packet \( p_{i',k'} \) with larger timestamp than \( p_{i,k} \), then we say \( p_{i,k} \) has an extra delay caused by \( p_{i',k'} \).

Now we claim that the total extra delay incurred to \( p_{i,k} \) cannot be more than \( n - k \); i.e. using 1-persistent protocol, \( p_{i,k} \) will not have delay that is \( (n - k) \cdot (T + t_g) \) longer than it has using GRR.

Since the only packets which may cause extra delay to \( p_{i,k} \) are the packets from \( N_k \)'s downstream nodes, if we can prove that at most one packet with the smaller timestamp can be inserted in front of \( p_{i,k} \) at each downstream node of \( N_k \) during the time period from time slot \( i + 1 \) to time slot \( t - 1 \), our claims hold. Consider a node \( N_j \) with \( j \geq k \), which is a downstream node of \( N_k \).

- Case 1. Suppose node \( N_j \) is in DA state at time slot \( i \). Then node \( N_j \) cannot be switched to DC state before \( p_{i,k} \) passes node \( N_j \) because \( d_j^0 \) will not be \( 0 \) as long as there are packets in the queues of \( N_j \)'s upstream nodes.
such a case, any packet $p_{i',j}$ with $i' \geq i$, i.e., with lower priority than $p_{i,k}$, will be first inserted into the local queue at node $N_j$. Consequently, $p_{i',j}$ cannot be sent out from node $N_j$ until $p_{i,k}$ passes node $N_j$. In other words, $p_{i',j}$ will not cause any extra delay to $p_{i,k}$.

- Case 2. Suppose that node $N_j$ is in DC state at time slot $i$. Then it keeps in DC state and does not block any packet until a packet $p_{i',j}$ is generated at time $i' \geq i$.

  - Case 2.1. If $p_{i',j}$ is generated after $p_{i,k}$ passes node $N_j$, then $p_{i',j}$ will not cause any extra delay to $p_{i,k}$.
  - Case 2.2. If $p_{i',j}$ is generated before $p_{i,k}$ passes node $N_j$, then $p_{i',j}$ causes node $N_j$ to be switched from DC state to DA state, and may reach destination earlier than $p_{i,k}$. In such a case, $p_{i',j}$ causes one extra delay to $p_{i,k}$. After this, node $N_j$ cannot be switched to DC state before $p_{i,k}$ passes node $j$, an analysis similar to Case 1 as we have conducted.

Summarizing the above cases, we know that at most one extra delay will be caused to $p_{i,k}$ at any node $N_j$ for $j \geq k$. Thus, any packet generated at $N_k$ will experience at most $(n-k) \cdot (T+t_g)$ more delay in the 1PER/1 protocol than in GRR algorithm. The packets generated at $N_1$ may experience $(n-1) \cdot (T+t_g)$ more delay in 1-persistent 1PER/1 protocol than in GRR algorithm. This completes the proof of Theorem 1.

C. Average Buffer Size

As mentioned in Section IV-A, every intermediate node $N_k$ would buffer and forward packets which were intercepted in DA state and destined to downstream nodes of it. Using the same assumptions as Section V-A1, the Poisson arrival rate for the traffic destined to downstream nodes of node $N_k$ is $\lambda_k = \sum_{i=0}^{n-1} \sum_{j=i+1}^{n} \lambda_{i,j}$. However, some of such traffic may pass node $N_k$ when $S_k$ is in DC state, which does not need to be buffered. Let $prob_{D,A,k}$ be the probability that $S_k$ is in DA state. Then $prob_{D,A,k} \cdot \lambda_k$ is the traffic that need to be buffered. Adding the traffic generated locally at node $N_k$, the total arrival traffic would be

$$\lambda_{total,k} = prob_{D,A,k} \cdot \lambda_k + \lambda_k \leq \lambda_k^2 + \lambda_k.$$

Analyzing the queuing behavior at node $N_k$, it is a typical M/D/1 model with Poisson arrival rate $\lambda_{total,k}$ and deterministic departure rate $\mu_k = 1.0, \sigma_k^2 = 0.0$. Then, the average buffer size of node $N_k$ is

$$B_k = \frac{1}{2} \cdot \frac{\mu_k^2}{1 - \rho_k}, \quad \text{where } \rho_k = \frac{\lambda_{total,k}}{\mu_k} \in [0, 1).$$

Clearly, $B_k$ is a monotone increasing function of $\rho_k$. Thus, we have

$$B_k \leq \frac{1}{2} \left( \frac{\mu_k}{1 - \rho_k} \right)^2 \tag{4}$$

VI. SIMULATION

As mentioned earlier, the design principle of 1-persistent protocols for opportunistic optical hyperchannels is to design work-conserving protocols with fairness. In this section, we conducted simulations to compare the performance of 1-persistent protocols running on an opportunistic hyperchannel with the performance of $\text{minSrcRR}$ protocol running on an opportunistic optical hyperchannel, $p_i$-persistent CSMA protocol and priority-based CSMA protocol running on a light-trail. We compare these protocols in terms of the following performance metrics: (1) throughput, (2) average delay at each node, (3) worst delay at each node. A higher throughput indicates that the system is more work-conserving. Both the average delay at each node and the worst delay at each node can be used as the measurements of fairness. We also conducted simulation to compare performance of our proposed 1-persistent protocols with the theoretical bounds we derived in Section V in terms of the relative maximum delay compared with an ideal fair protocol GRR, throughput amplification ratio, and buffer size.

The unidirectional opportunistic optical hyperchannel we simulated has 11 nodes. The data rate in this opportunistic optical hyperchannel is set to 2.5Gbps, packet size is 1500 bytes, and the guard time $t_g$ is $10^3$ seconds. The destinations of all packets are uniformly distributed among all its downstream nodes. For example, the destinations of packets generated at node $N_j$ are uniformly distributed in $[j+1, n]$. The packet arrival process (i.e., packet generating process) at node $N_j$ is assumed to follow Poisson distribution with parameter $\lambda_j$, and load system is uniformly distributed among all nodes, which means $\lambda_j = \lambda_k, 0 \leq j, k < n$. Then, system load can be identified by $\lambda = \sum_{j=0}^{n-1} \lambda_j$.

A. Throughput

In an opportunistic optical hyperchannel, a packet will be dropped at the first downstream node which is in DA state after it passes its destination instead of occupying the whole optical bus. Thus, multiple non-overlapping segments can be used to transmit packets simultaneously without conflict. Therefore, a larger throughput can be achieved in an opportunistic optical hyperchannel compared with the corresponding light-trail. Fig. 3 verifies this property. Since both 1-persistent protocols and $\text{minSrcRR}$ protocol are designed for opportunistic optical hyperchannel, both exhibits the throughput amplification feature and can achieve throughput higher than 2.5Gbps when system load is higher than 100%. Figs. 3 also shows that the
curves of 1PER/1 and 1PER/2 protocols almost completely overlap. This indicates that for practical purpose 1PER/2 is a better choice than 1PER/1 because of its \(O(1)\) time complexity. From Fig. 3, we can see that protocols specifically designed for opportunistic optical hyperchannels can achieve much higher throughput than priority-based CSMA protocol and \(p_i\)-persistent CSMA protocol under different system loads.

### B. Average Packet Delay

For priority-based CSMA protocol, the packets originated from upstream nodes have smaller average packet delay than the packets originated from downstream nodes simply because of the fixed node priority. This is verified by Fig. 4.

For \(p_i\)-persistent CSMA protocol, the difference between average delays of different nodes is small, given that \(p_i\) are correctly calculated (i.e. the real traffic demand matches the estimated traffic demand). As shown in Fig. 4(a), when \(p_i\) are calculated according to a low system load, the curve of the average delay for \(p_i\)-persistent CSMA protocol is quite flat. However, when the system load is high, as shown in Fig. 4(b), the differences of average delays among nodes can be dramatic, which means that the fairness of \(p_i\)-persistent protocol is hard to achieve if the real traffic demand does not match the estimated traffic demand.

For \(\text{minSrcRR}\) protocol, the first node has no reference node, thus, its average delay will be less than others. From Fig. 4(a), we can see that the average delay at all nodes except the first node is quite close. When system load is high as shown in Fig. 4(b), \(\text{minSrcRR}\) protocol can still provide good fairness assurance while \(p_i\)-persistent protocol fails to do so.

For 1-persistent protocols, packets generated at downstream nodes tend to have lower average delays since their transmissions may be enabled when their originated nodes’ switches are changed from DC state to DA state, blocking packets from upstream nodes. Thus, the average delay curves of 1-persistent protocols are decreasing curves. However, the slope of such decreasing curve is very small, which means that 1-persistent protocols can still provide good fairness. Moreover, as we can see from Fig. 4(a) and (b), the average delay of packets generated at each node of an opportunistic hyperchannel running a 1-persistent protocol is much smaller than that of other protocols.

### C. Worst Packet Delay

Though 1-persistent protocol may allow the downstream nodes to block the packets from the upstream nodes and transmit their locally generated packets each time when they switch from DC state to DA state, the fair schedule is enforced once the CA switch at a node is in DA state. As proved in Theorem 1, at most one packet with larger timestamp can be inserted before a packet with lower timestamp at any downstream node. Thus, each node will experience similar worst delay. Fig. 5 shows that the curves of worst delay under 1-persistent protocols and \(\text{minSrcRR}\) protocol are quite flat especially when the system load is heavy. The difference of the worst delays among different nodes under both \(p_i\)-persistent CSMA protocol and priority-based CSMA protocol are quite large. Fig. 5 shows that 1-persistent protocols can provide similar fairness as \(\text{minSrcRR}\) protocol in terms of the worst delay at each node.

Though the performance in terms of throughput and fairness of 1-persistent protocols and \(\text{minSrcRR}\) protocol is quite close in the simulation, these two protocols can not replace each other due to the following reasons: (1) 1-persistent protocols can provide good fairness, however, it can not assure QoS, i.e., nodes can not transmit according to their weight. Thus, 1-persistent protocols can not replace \(\text{minSrcRR}\) protocol if differentiated QoS metric is important; (2) In an opportunistic optical hyperchannel, if throughput is more important than providing QoS, 1-persistent protocols outperform \(\text{minSrcRR}\) protocol due to its higher throughput and simpler implementation.

### D. Throughput Amplification Factor

In the previous section, we have derived the theoretical upper bound and lower bound of throughput amplification ratio. In this section, we conduct simulation to see how close that the real throughput amplification ratio is to our derived bounds. Fig. 6 compares the throughput amplification factor \(T_A\) of 1-persistent protocol running on an opportunistic hyperchannel with the lower bound \(F_{T_A}^-\) and upper bound \(F_{T_A}^+\) of throughput amplification factor calculated according to Equation (2) and Equation (3) under heavy system load. From Fig. 6, we can see that the actual throughput amplification ratio is quite close to the upper bound.

### E. Relative Maximum Packet Delay

We also conduct simulations to compare the packet delay in an opportunistic hyperchannel implementing 1-persistent protocol with an ideal fair protocol, called GRR protocol. To be comparable, we assume that the destination of all traffic is the last node. Fig. 7 shows that the maximum delay difference between 1PER/1 and GRR is bounded by
System Load = 99.85%

A more appropriate opportunistic optical hyperchannel shall be setup such that the total traffic is reasonable since an opportunistic optical hyperchannel is less than the wavelength capacity. Otherwise, a setup such that the total traffic is too large electronic buffer capacity at each node when system load is not significant heavier than the wavelength capacity. This is reasonable since an opportunistic optical hyperchannel is setup such that the total traffic load in an opportunistic optical hyperchannel is less than the wavelength capacity. Otherwise, a more appropriate opportunistic optical hyperchannel shall be setup.

We generalize the opportunistic hyperchannel architecture to one that removes the linear topology limitation, thereby improving the system utilization for the case as shown in Example 1. Such a generalization is useful for constructing special purpose networks. We call the generalized opportunistic hyperchannel the non-linear opportunistic hyperchannel. In general, a non-linear opportunistic hyperchannel is informally

Fig. 6. The throughput amplification factor of the theoretical lower bound, simulation results of 1-persistent protocol, and the theoretical upper bound under different system loads.

Fig. 7. The relative maximum packet delay of the theoretical upper bound and simulation results of 1-persistent protocol compared with GRR protocol.

9 \((n - 1 = 10 - 1 = 9)\) time intervals where each time interval is 4.801\(\mu s\) in our simulation, which verifies the proof of Theorem 1.

F. Buffer Size

Fig. 8 compares the theoretical average buffer size with the simulation results, where the former is an upper bound of the later. Maximum buffer size at each node under different system loads is shown in Fig. 9. As analyzed in Section V-C, more buffers are needed at intermediate nodes with larger \(\lambda_{ij}\) which is demonstrated in Fig. 9. From Fig. 9, we can see that an opportunistic optical hyperchannel will not require too large electronic buffer capacity at each node when system load is not significant heavier than the wavelength capacity. Otherwise, a more appropriate opportunistic optical hyperchannel shall be setup.

VII. GENERALIZATION OF OPPORTUNISTIC HYPERCHANNEL

We call the opportunistic hyperchannel discussed so far a linear opportunistic hyperchannel, simply because the nodes in such a hyperchannel are ordered as a linear array. We denote a linear opportunistic hyperchannel as an ordered node sequence \((N_0, N_1, \ldots, N_n)\) according to the positions of the nodes in the hyperchannel. In some applications, linear opportunistic hyperchannel are not efficient, as shown in the following example.

Example 1: Consider the tree optical network topology shown in Fig. 10(a). We use \(N_i \rightarrow N_j\) to represent a connection request for sending packets from node \(N_i\) to node \(N_j\). Let \(R = \{N_0 \rightarrow N_1, N_1 \rightarrow N_3, N_5 \rightarrow N_1, N_5 \rightarrow N_2, N_1 \rightarrow N_3, N_4 \rightarrow N_3, N_2 \rightarrow N_3\}\) be a set of connection requests. In order to accommodate such requests, three linear opportunistic hyperchannels may be established, as shown in Fig. 10(a), with one distinct wavelength for each channel. These linear opportunistic hyperchannels are not efficient because one wavelength can support all these connections.

A. Non-Linear Opportunistic Hyperchannels

We generalize the opportunistic hyperchannel architecture to one that removes the linear topology limitation, thereby improving the system utilization for the case as shown in Example 1. Such a generalization is useful for constructing special purpose networks. We call the generalized opportunistic hyperchannel the non-linear opportunistic hyperchannel. In general, a non-linear opportunistic hyperchannel is informally
defined as a subnetwork $G$ of a mesh network $M$ that satisfies the following conditions:

- $G$ is a structure obtained by superimposing a set of linear opportunistic hyperchannels in $M$;
- optical connections in $M$ can be shared by different component linear opportunistic hyperchannels, and the transmissions on shared connections must have the same direction;
- $G$ uses a single wavelength; and
- in any switch setting, no optical cycle exhibits in $G$.

Based on this definition and assuming appropriate node architectures with adequate dynamic optical switching and electronic buffering capabilities, one can construct many different types of non-linear opportunistic hyperchannels. We restrict our discussions to implementations of non-linear opportunistic hyperchannels whose underlying topologies are trees.

### B. Gathering Tree

A gathering tree $T_M$ is a directed tree with leaf nodes being the start nodes of some component linear opportunistic hyperchannels, the root node being the end node of some component linear opportunistic hyperchannel. The packet transmissions are in the direction toward the root. Each intermediate node in a gathering tree has only one optical output, but may have multiple optical inputs; one input can be connected to the output through a switch at any time.

Assuming that for a gathering tree $T_M$, each node has a constant in-degree, i.e. a fixed number of optical inputs. We show how $T_M$ can be constructed in mesh network $M$ by considering Example 1. The gathering tree $T_M$ can be constructed to accommodate all requests using one wavelength as shown in Fig. 10(b).

The node architecture for $T_M$ is shown in Fig. 10(c), where an Integrated Logic Controller (ILC) controls two optical shutters simultaneously. Integrating the control of two optical shutters, it is equivalent to that each internal node of $T_M$ tree is associated with a $3 \times 1$ bi-continue-add (BCA) switch $S_k$.

There are three buffers in each $T_M$ node: $BUF_A$ is used to save the packets intercepted from input $A$; $BUF_B$ is used to save the packets from input $B$; and $BUF_C$ is used to save the packets generated locally. The optical signals from both input $A$ and input $B$ will drop a copy at the electronic buffer, and at the same time go to its corresponding optical shutter. Depending on the state of the BCA switch, the signal may go to the downstream (parent) node directly or need to be re-transmitted from the local buffers. The local buffers are responsible for receiving and re-transmitting the packets intercepted from input $A$ and $B$, and transmitting the packets generated locally. A $3 \times 1$ BCA switch $S_k$ can be in one of the following three states as shown in Fig. 10(c):

- **Drop-Add (DA) state.** The optical shutters associated with input $A$ and input $B$ are in OFF state. In this state, the packets from the buffers of the node are transmitted.
- **Drop-Continue from $A$ (DCA) state.** The optical shutter associated with input $A$ is in ON state and the optical shutter associated with input $B$ is in OFF state. In this state, the optical signals from $A$ can pass through $S_k$ and the signals from $B$, if any, are saved in $BUF_B$.
- **Drop-Continue from $B$ (DCB) state.** The optical shutter associated with $A$ is in OFF state and the optical shutter associated with $B$ is in ON state. In this state, the optical signals from $A$, if any, are saved in $BUF_A$ and the signals from $B$ can pass through $S_k$.

By choosing the setting scheme of $S_k$, a class of packet scheduling protocols can be designed for a gathering tree $T_M$. In what follows, we introduce the implementation of 1-persistent protocols in a gathering tree.

In the 1-persistent protocols for a linear opportunistic hyperchannel, a CA switch is set to be in DC state when the local buffers are empty. To implement the 1-persistent idea in a gathering tree $T_M$, a little modification should be made for the $T_M$ nodes due to three states of a BCA switch.

There are two DC states for a BCA switch: DCA and DCB. Assuming Poisson arrival process at each node, let $w(i,A)$ (resp. $w(i,B)$) be the arrival process parameter of the traffic from node $N_i$’s input $A$ (resp. $B$) destined for downstream nodes of $N_i$. Let $Prob(i,DCA)$ and $Prob(i,DCB)$ be the the probabilities of a BCA switch $S_i$ in DCA state and DCB state, respectively. $Prob(i,DCA)$ and $Prob(i,DCB)$ can be computed from equations $\frac{Prob(i,DCA)}{Prob(i,DCB)} = \frac{w(i,A)}{w(i,B)}$ and $Prob(i,DCA) + Prob(i,DCB) = 1$.

When all local buffers at a $T_M$ node $N_i$ are empty, the output of BCA switch $S_i$ connects to input $A$ and $B$ alternatively, with fractions $Prob(i,DCA)$ and $Prob(i,DCB)$ of time in DCA and DCB state, respectively. Each $T_M$ node $N_i$ maintains two variables: $num_{i,A}$ and $num_{i,B}$. Initially, $num_{i,A} = num_{i,B} = 0$. $num_{i,A}$ (resp. $num_{i,B}$) is increased...
by 1 at each time slot when \( S_i \) is in DCA (resp. DCB) state. We compare \( \text{num}_{i,A} \cdot \text{Prob}_{i,(DCB)} \) with \( \text{num}_{i,B} \cdot \text{Prob}_{i,(DCA)} \) to decide the state of \( S_i \) for the next slot when local buffers are empty:

- If \( \text{num}_{i,A} \cdot \text{Prob}_{i,(DCB)} > \text{num}_{i,B} \cdot \text{Prob}_{i,(DCA)} \), set \( S_i \) to be in DCB state;
- if \( \text{num}_{i,A} \cdot \text{Prob}_{i,(DCB)} < \text{num}_{i,B} \cdot \text{Prob}_{i,(DCA)} \), set \( S_i \) to be in DCA state;
- otherwise, the state of \( S_i \) does not change.

When \( S_i \) is in DCA state, it is possible that a packet arrives from input \( B \). If such a case happens, the packet is buffered in \( BUFB \), and \( S_i \) switches to DA state in the next time slot. Then, packets in \( BUFB \) can be transmitted. This, which may violate the assigned ratio, is called slot stealing. Slot stealing is allowed by increasing \( \text{num}_{i,B} \) with 1 for this case.

1-persistent protocols for \( T_M \) can be summarized as follows:

- Any node with at most one optical input works using the same rules as in a linear opportunistic hyperchannel.
- Any node with multiple (in our case, 2) optical inputs, sets its BCA switch to DA state when at least one of its local buffers are non-empty. In this state, the packet with the smallest timestamp will be scheduled to be transmitted persistently from local buffers.
- For any node \( N_i \) with more than one optical input, if its all local buffers are empty, it sets its BCA switch \( S_i \) to be in either DCA or DCB state with probability \( \text{Prob}_{i,(DCA)} \) or \( \text{Prob}_{i,(DCB)} \), respectively.

\( \)C. Multicast-and-Select Tree

We introduce another non-linear opportunistic hyperchannel called multicast-and-select tree in a mesh network \( M \). A broadcast-and-select tree \( T_{MS} \) is a directed tree in \( M \) with the root being a unique node, which is the start node of every component linear opportunistic hyperchannel, and with leaf nodes being the end nodes of some component linear opportunistic hyperchannels. The packet transmissions are from the root to leaves. Each intermediate node has only one optical input, but may have multiple optical outputs; the input is connected to all outputs through a multicast splitter (no switching element and electronic buffers are not needed). The root node transmits packets to all nodes in the tree, and each non-leaf node decides if it keeps the broadcast packets.

\( \)D. Applications of Nonlinear Opportunistic Hyperchannels

A pair of gathering tree \( T_M \) and multicast-and-select tree \( T_{MS} \) can be used to implement an access network with the root connected to the core network and leaves as users. Tree \( T_M \) is used to carry upstream traffic and \( T_{MS} \) is used to carry downstream traffic.

A pair of gathering tree \( T_M \) and multicast-and-select tree \( T_{MS} \) can also be used to implement an access network with the root connected to the core network and leaves as users. Tree \( T_M \) is used to carry upstream traffic and \( T_{MS} \) is used to carry downstream traffic.

Compared with an access network implemented by a TDM-PON that determines the schedule for upstream traffic either through fixed TDM or centralized DBA protocols, our proposed non-linear opportunistic hyperchannel can implement distributed MAC protocol to share bandwidth for upstream traffic. Compared with WDM-PON which requires one wavelength per user for upstream traffic, a \( T_M \) opportunistic hyperchannel requires only one wavelength as long as its bandwidth can accommodate all upstream traffic.

VIII. Conclusion

As an improvement of light-trail, opportunistic optical hyperchannels inherently provide the capability of designing fully distributed and work-conserving media access control. To demonstrate this capability, we proposed simple 1-persistent protocols and studied their performance. Our analysis and simulations show that 1-persistent protocols have bounded worst-case delay and reasonable fairness. The price paid for this advantage is the additional cost for optical switches and electronic buffers. This cost overhead associated with high-speed optical shutters is further offset by throughput amplification achieved using space-division multiplexing (SDM) through self-adjusted multiple concurrent packet transmissions on non-overlapping segments. As a generalization of opportunistic hyperchannel, non-linear opportunistic hyperchannels are introduced in the context of mesh networks or access networks. Optical hyperchannels also have applications in constructing high-performance storage-area networks and interconnection subsystems for processor clusters. We believe that the unique feature of opportunistic hyperchannels that combines wavelength-division multiplexing, time-division multiplexing and space-division multiplexing deserves further investigations for the design and implementation of high-performance optical/electrical hybrid communication networks.

ACKNOWLEDGMENT

The authors would like to thank the associate editor, Dr. John Liu and the anonymous reviewers for their constructive comments that improved the paper. The work described in this paper was partially supported by the National Science Foundation, Grant NSF-0714057, and support from the City University of Hong Kong under grants 7002327 and 7002468.

REFERENCES


Ashwin Gumaste is currently the James R. Isaac Chair in the Department of Computer Science and Engineering at the Indian Institute of Technology, Bombay. He is at present also a Visiting Scientist with the Massachusetts Institute of Technology (MIT), Cambridge, USA in the Research Laboratory for Electronics - Claude-E-Shannon group. He was previously with Fujitsu Laboratories (USA) Inc in the Photonics Networking Laboratory (2001-05). Prior to this he worked in Fujitsu Network Communications R&D and prior to that with Cisco Systems in the Optical Networking Group (ONG). Ashwin’s research has been funded by vendors, providers, system integrators and government agencies. He can be reached through www.ashwin.name.

S. Q. Zheng received the PhD degree from the University of California, Santa Barbara, in 1987. After being on the faculty of Louisiana State University for eleven years since 1987, he joined the University of Texas at Dallas, where he is currently a professor in computer science. Dr. Zheng’s research interests include algorithms, computer architectures, networks, parallel and distributed processing, hardware/software codesign, real-time and embedded systems, telecommunications, and VLSI design. He has published extensively in these areas. He was a consultant to several high-tech companies, and holds numerous patents. He has served as program committee chairman of numerous international conferences and as editor of several professional journals.