Evaluation of single-hop architecture for metropolitan area networks

K.V.S.S.S.S.SAIRAM (s5kanduri@rediffmail.com) Dr.
T. Janardhana Rao & Dr. P.V.D. Somasekhar Rao

Abstract

With the widespread use of broadband access technologies and the development of high-speed Internet backbones, the requirement for high-performance metropolitan area networks (MANs) is increasing. Traditional ring- or star-based metro networks are costly to scale up to high speed and cannot recover from multiple failures, while backbone solutions are too expensive to fit into the cost-sensitive metro market. This paper proposes a virtual fully connected (VFC) architecture for metro networks to provide high-performance node-to-node all-optical transportation. The architecture emulates a fully connected network by providing optical channels between node pairs without intermediate buffering, and thus realizes single-hop transportation and avoids expensive packet routers. In addition, a scheduling algorithm is developed for medium access control and dynamic bandwidth allocation, which achieves 100% throughput and provides a fairness guarantee. Simulations show that the VFC network achieves good performance under both uniform and non-uniform loads.

Keywords: WDM network; Metropolitan area network; Scheduling algorithm; Bandwidth allocation; Throughput;Fairness

1. Introduction

With the development of Internet technologies, metropolitan area networks (MANs) are becoming increasingly important in providing high-performance interconnections among access networks (ANs), high-end users (such as Internet service providers (ISPs)) and wide-area backbones. Unlike ANs used for traffic aggregation and distribution among end users, traffic flows carried by a MAN are rather arbitrary; some are within the same MAN (e.g., from one AN to another or to an ISP) and others are routed to and from the backbone , which generates fully meshed traffic matrices for MANs. At the same time, broadband access technologies (such as digital subscriber line (DSL) technologies, cable modem and Ethernet passive optical network (PON)), as well as emerging applications (e.g., peer-to-peer and video communication) bring increasing bandwidth requirements in metro networks. From this point of view, MANs are more like backbones by providing broadband meshed channels, even though they span much shorter distances (usually 200–300 km) Nevertheless, they differ from backbones in that the metro market is more cost sensitive , which prevents the application of high-performance yet costly backbone solutions to metro networks.

This paper proposes a novel architecture for a WDM MAN, called a virtual fully connected (VFC) network. The main advantages of a VFC network lie in the following aspects. First, by providing bufferless single-hop transportation between node pairs, the architecture introduces a cost-effective high-speed solution without using expensive routers. Second, it is designed to support dynamic bandwidth allocation according to traffic fluctuation, which is of special importance to metro networks where the traffic tends to be bursty . At the same time, the proposed scheduling algorithm provides 100% throughput, as well as guaranteed fairness. Third, it is topology independent, and thus can be combined with mesh physical topology to provide high reliability under multiple failures.

The remainder of this paper is organized as follows. The next subsection gives a brief review of related work. Section 2 describes VFC architecture in detail; Section 3 explains the scheduling algorithm; Section 4 presents simulated performance evaluation; and Section 5 concludes the paper and addresses our future work.

1.1. Related work

Several bufferless WDM network architectures have been proposed, such as RAINBOW [8], LAMBDANET [6], HORNET and WDM star based on an arrayed-waveguide grating (AWG) [11]. However, all of them are based on either ring or star topology, which suffer from poor reliability and scalability. In ring networks, although a single failure can be recovered very fast, double failures separate the network into two parts. In addition, an increase in node number leads to a longer circumference, where the potential long light path may cause low bandwidth utilization and poor signal quality. On the other hand, stars cannot even recover from a single failure. A recent improvement in AWG-based stars is to introduce a passive star coupler (PSC)-based broadcasting network in parallel ; however, this still cannot recover from multiple failures. Besides, an increase in node number puts a heavy burden on the center node and a PSC may not work with a high split degree.

Time domain multiplexing is employed for fine granularity bandwidth allocation in WDM networks and, where the problem of routing,

wavelength and time-slot assignment is similar to the routing and wavelength assignment in traditional WDM networks. This architecture requires high-speed optical switches for slot channel establishment, which is non-trivial. The proposal in [3] avoids frequent network configuration by connecting a number of nodes using a unidirectional wavelength channel called a light trail, which functions as a time-domain shared medium. Each node is able to receive from the upstream and send to the downstream nodes by decoupling from and coupling to the traversing optical signal. Due to the power splitting at each node, the length of a light trail is limited and the expected length is 5 hops [3].

Recently, a single-hop optical network architecture called time-domain wavelength interleaved network (TWIN) has been proposed . In this architecture, each edge node has a tunable laser transmitter and a fixed optical receiver. The tunable laser can be dynamically configured to generate signals at one of a number of wavelengths; while the fix receiver works at a predetermined wavelength. In the intermediate nodes, incoming signals at the same wavelength are simply combined together and directed to a predetermined route. A unique wavelength is allocated to the receiver of each edge node in advance, thus sending signals to a particular edge node can be realized by tuning the laser to the corresponding wavelength. The advantage of this proposal is that there is no high speed packet switching and electrical processing in the intermediate nodes, which reduces cost and complexity of the network.

Two types of contentions exist in TWIN: An edge node may have traffic to multiple destinations but the tunable laser can transmit to a single destination at a certain time; On the other hand, multiple nodes may have traffic to the same destination but the destination can accept signals from at most one node at any time. Therefore, a network wide scheduling is required to coordinate the transmission of the tunable lasers in the whole network. This issue is similar to the scheduling in packet switches, but the significant difference comes from the non-negligible propagation delay, which makes it very difficult to achieve high throughput and low delay.

A centralized scheduling algorithm called TWIN iterative independent set (TIIS) is proposed in TIIS assumes the scheduler knows the traffic change at each node immediately and uses this information to arrange the transmission. However, the delay to collect such information cannot be avoided. Extending the algorithm to deal with such delay is not straightforward. The reason can be briefly explained using a simple example where node S_1 and S_2 are sending to D. Suppose the delays between S_1, S_2 and the scheduler are t_1 and t_2, respectively, if $t_1=t_2=0$, then any change of the queues in S_1 and S_2 can be immediately observed and used for the scheduling. In contrast, if $t_1=0$ and $t_2=1$ ms, then a new arrival at S_2 will experience a delay of at least $2t_2$ before being sent out. This delay is inevitable even if the queue is empty. Therefore, the delay performance is seriously affected and the algorithm does not provide fairness among nodes with different propagation delays. Under dynamic traffic, such delay may deteriorate the throughput performance as well. In addition, the algorithm is designed to work under the assumption that the traffic is admissible, i.e., no overload may occur. However, such assumptions may not apply to real networks where overload may be caused by special events, denial of service (DoS) attacks, etc.

2. Architecture of VFC network

2.1. Node architecture

Each node consists of two parts: a service access module (SAM) and an OXC, as illustrated in Fig. 1.

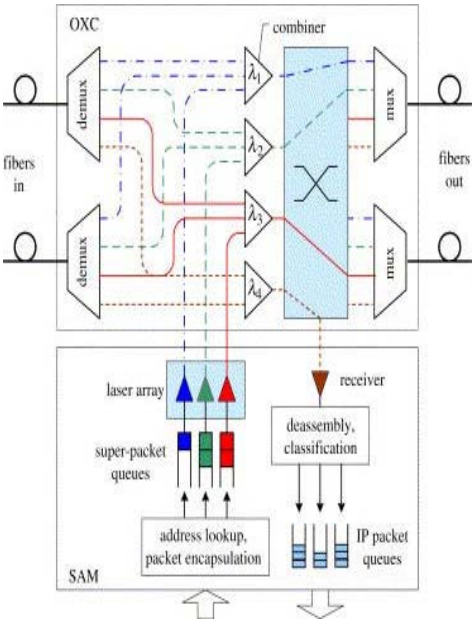


Fig. 1. Node architecture.

The OXC is different from traditional ones in that it contains passive optical combiners inside. For clarity, Fig. 1 shows a simple case where all the signals on the same wavelength are combined together and routed to a certain output fiber. In real cases, the node can be designed such that each input wavelength can be routed to any output fiber according to the configuration, and only those destined for the same output port are combined together. In particular, signals on the node's home wavelength are combined and terminated in a local receiver. With recent advances in low-loss optical combiners [12], it is feasible to include none or only a few optical amplifiers in metro networks using the proposed nodes, since there is no power splitting and such networks span a limited distance.

2.2. Network architecture

With the above node architecture, a WDM network with N nodes can be decoupled into N wavelength trees by configuring the OXC of each node properly. Suppose node i utilizes λ_i as its home wavelength, then it acts as the root of a spanning tree occupying λ_i, and the other nodes are the leaves. Fig. 2 illustrates a six-node network and three wavelength trees destined to nodes A, B and C, respectively; the trees to D, E and F can be created similarly.

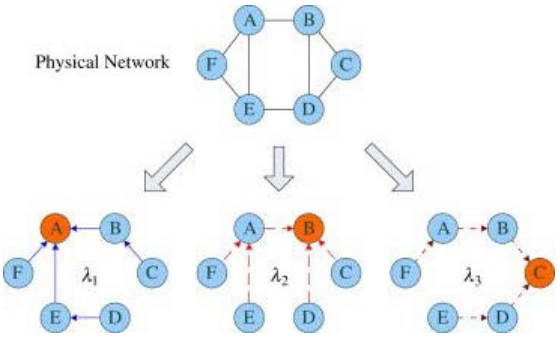


Fig. 2. Decoupling of a VFC network.

Within a tree, signals from any leaf node can be transported to the root through the parent nodes and, since no buffering is introduced along the route, the transportation can be regarded as a single hop even if the signal traverses multiple intermediate nodes. Note that each node acts as the root of a unique wavelength tree and is able to receive from all the other nodes; transportation between any node pair is via a single hop, and thus we can say that the network is virtual fully connected.

Although the size of a VFC network is constrained by the number of wavelengths in each fiber, a large network can be divided into multiple sub-networks, as discussed later. In addition, advances in dense WDM technology have greatly relieved this constraint by increasing the available wavelength channels in each fiber.

The policy used to generate the spanning trees does not affect the throughput (as shown in the next section). Nevertheless, the tree based on the shortest paths offers the best delay performance since such trees have the minimum propagation delay. The concept of wavelength tree in this paper is similar to the destination tree in the TWIN architecture . However, our architecture is different in that each node consists of a fixed laser array instead of a single tunable laser. Although a seemingly minor modification, our new architecture enables the design of network scheduling algorithms that provide high performance in terms of throughput, delay and fairness. The difference between the two architectures can be explained using Fig. 2 with three demands: F→A using λ_1, F→B using λ_2 and B→A using λ_1. In case of TWIN, F has a single laser thus cannot send to both A and B. At the same time, node A cannot receive from both F and B. Therefore, the scheduler needs to resolve the contentions at both source and destination nodes. Taking into account of the propagation delay, design of such algorithms is non-trivial. On the other hand, our architecture eliminates the contentions at source nodes by using a fixed laser array, i.e., F is allowed to send to both A and B simultaneously. Thus the algorithm design is greatly simplified. An algorithm is proposed in this paper and is proved to provide 100% throughput.

In each wavelength tree, all the leaf nodes send signals using the same wavelength and the signals are simply combined together on the way to the destination. Therefore, the wavelength tree is a shared medium among the leaf nodes, which requires network scheduling for the media access control to avoid signal collision in both the intermediate and destination nodes. In Fig. 3, suppose nodes A, B and D are sending to F, the paths share three nodes C, E and F, which are the potential places to experience signal collision. The scheduling algorithm must control the transmission time of A, B and D such that their signals do not arrive at those nodes simultaneously. The problem can be simplified by only considering the collisions at the root node, as stated by the following theorem.

Theorem 1

Given a wavelength tree and supposing the width of each optical signal is 0, as long as two signals arrive at the root at different times, they do not collide anywhere in the tree.

Proof

Suppose signals 1 and 2 (either from a single node or from two different nodes) arrive at the root at times t_1 and t_2 without collision, there must be

$$t_1 \neq t_2. \quad (1)$$

If signals 1 and 2 do not share any intermediate node, there will be no collision at all. Otherwise, suppose both of them traverse node i. Since no alternate route can be found from any node to the root in a tree, the two signals must take the same route from i to the root. Denote the delay from i to the root as τ, the arrival times of signals 1 and 2 at node i can be expressed as $t_1-\tau$ and $t_2-\tau$, respectively. According to (1), we have:

$$t_1-\tau \neq t_2-\tau. \quad (2)$$

This means they do not collide in i, and the proof is completed.

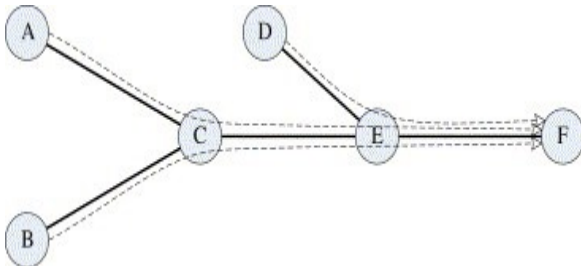


Fig. 3. Signal propagation in a wavelength tree.

2.3. Control network

Since the network is decoupled into multiple trees, a scheduler is needed by each of them to coordinate the medium access of the transmitters for collision avoidance and bandwidth allocation. The schedulers can be located either centralized or distributed:

• Centralized: all the schedulers are put together in a single node. This facilitates network management and algorithm upgrade, but it brings the drawback that the scheduling node must have high reliability and strong computation ability. The node is critical and its failure disrupts the whole network; usually a backup scheduler is highly necessary for this solution.

• Distributed: the scheduler for each tree is geographically distributed, usually located in its root node. In this case, each scheduler has low complexity and its failure does not affect other trees.

A control network is employed to transport two kinds of signaling data: queuing information from the TXs to the schedulers and scheduling results in the reverse direction. A packet-switched control network with the same topology as the physical one is used in our proposal, which is constructed by putting a packet switch in each node and connecting the neighboring switches with a certain wavelength, as shown in Fig. 4.

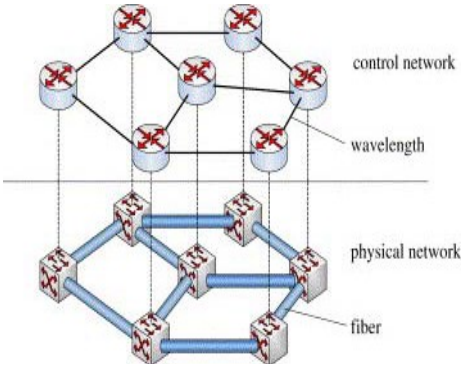


Fig. 4. Control network.

Note that the signaling packets do not introduce heavy traffic, and thus a lightly loaded network can be achieved without high-speed packet switches. In addition, since the number of signaling packets is determined by the control algorithm, the traffic distribution is highly predictable, which makes it possible to achieve low packet delay by carefully designing the routing protocol.

2.4. Timing

Consider the tree in Fig. 5; the propagation delay from node i to the root is denoted as d_i, and the delay between two neighboring nodes i and j is denoted as $d_{i,j}$. If each of the nodes i and j sends out an optical pulse with width w, according to (1), the transmission times t_i and t_j must satisfy the following condition to avoid conflict (Fig. 5 shows the case i=1 and j=4):

$$|(t_i+d_i)-(t_j+d_j)|>w. \quad (3)$$

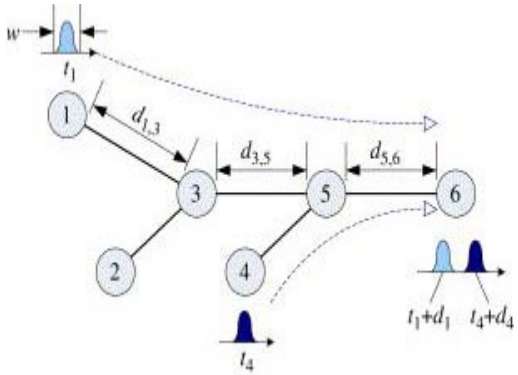


Fig. 5. Wavelength tree.

To satisfy the above condition, two issues related to timing must be solved in advance:

(1) The propagation delay d_i must be obtained; and

(2) The system time for all the nodes must be the same.

For the first issue, since no buffering is introduced in the tree and the physical network is constant once deployed, the delay can be calculated from the link delays along each route, e.g., in Fig. 5, $d_1=d_{1,3}+d_{3,5}+d_{5,6}$. The delay between neighboring nodes can be precisely measured by an optical loop-back. If a non-negligible delay exists within each OXC, the calculation can easily be modified accordingly.

For the second issue, a possible solution is to synchronize the whole network using a global positioning system (GPS); however, this introduces considerable complexity and cost. Alternatively, we propose a compensation solution that does not require network-wide synchronization. In Fig. 5, we denote the system clock for node i as clk_i, and use the root clock as the standard time clk. By equipping each node with a high-accuracy digital clock, the offset $\Delta_i=clk-clk_i$ can be considered constant during a certain period of time (in the millisecond range). In this case,

letting node i transmit at time t according to clk is equivalent to starting the transmission at $t-\Delta_i$ according to clk_i. Thus, the difference between the local clock and the standard clock is compensated by the offset, which can be measured by periodically sending time stamps to the root node. For instance, suppose the root clock counts to t when a time stamp arrives from node i indicating its transmission time as t', then the offset can be calculated as:

$$\Delta_i=t-(t'+d_i). \quad (4)$$

For simplicity, the following discussion assumes that $\Delta_i=0$ holds over the whole network.

In each wavelength tree, super-packets are transmitted in fixed-length slots, together with a guard time, a series of preamble bits and a time stamp, as shown in Fig. 6. The guard time compensates for the inaccuracy of the propagation delay, the preamble bits are used for clock data recovery (CDR) at the receiver, and the time stamp can be used for the calculation in (4).

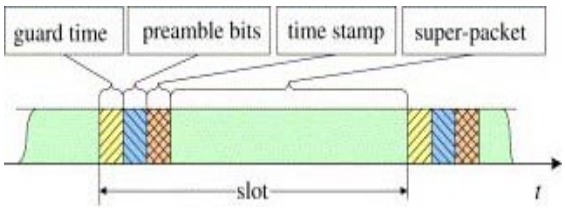


Fig. 6. Slot structure.

2.5. Survivability

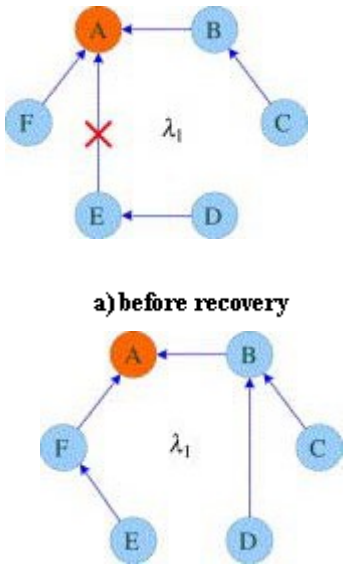
Failure recovery in VFC networks is simple and efficient. Note that data transmissions to a particular node follow a spanning tree with that node as the root, any single or multiple failures can be recovered as long as the physical topology remains as a connected graph. The detection of a failure triggers the following three steps for service recovery:

(1) Generate a new spanning tree for each node using its home wavelength;

(2) Reconfigure the related OXCs to construct the spanning trees; and

(3) Update the delay parameters of the scheduler.

The OXC reconfiguration is straightforward and how to obtain the propagation delay is discussed in Section 2.4. Thus the remaining problem is how to find the new spanning tree. The issue on finding a spanning tree has been researched in graph theory and the results can be directly applied to our proposal. Such algorithms do not have high complexity, it has been shown that even the minimum spanning tree can be found with linear complexity [9]. In particular, the recovery does not have to recalculate the whole spanning tree. Instead, only the nodes that are separated from the original tree need to be considered while the other nodes can be kept unchanged. This means the complexity of the spanning tree recalculation can be further reduced. Fig. 7 shows the recovery of the spanning tree for node A under the failure of link A–E (refer to the physical topology in Fig. 2). Only nodes E and D need to be reconnected to the tree through alternate paths, while the connections from the other nodes remain unchanged.



(b) After recovery.

Fig. 7. Failure recovery for a spanning tree.

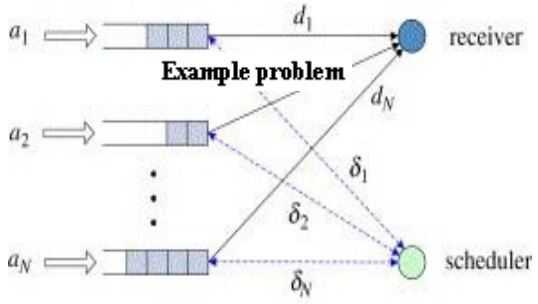
It is worth noting that the control network carrying signaling packets also needs to be recovered in the case of failure; the rapid restoration of such networks under multiple failures has been investigated in our previous work and can be directly applied to VFC networks .

3. Weighted slot-channel scheduling (WSCS)

Since each wavelength tree is a shared medium among the TXs, a scheduling algorithm is needed to coordinate transmissions for collision avoidance, as well as bandwidth allocation. The basic idea of a scheduling algorithm is to arrange the transmission time (start time and duration) for each TX according to its backlog, within the context of a non-negligible propagation delay (i.e., the delay from the TXs to the receiver and the delay between each TX and the scheduler). This section gives the mathematical formulation of the problem and presents the details of our scheduling algorithm. Without loss of generality, the following discussion assumes the bandwidth of each wavelength to be 1.

3.1. Problem description

With the network decoupled into multiple independent wavelength trees, we only need to consider a single tree for the problem of scheduling. Given a tree with TX set



$V = \{1, 2, \ldots, N\}$, the arrival rate of TX i $(i \in V)$ is denoted as a_i, and its propagation delay to the scheduler and the receiver are δ_i and d_i, respectively. Scheduling of a tree can be modeled as a service control problem in a multi-queue one-server system. If the scheduler is located in the receiver, we have $d_i = d$ $(i \in V)$. This paper considers the general case without regulation between d_i and δ_i.

Assume the kth transmission of TX i takes place during $[s_i^k, t_i^k]$, then the time occupation of the first K transmissions at the receiver is:

and the total length of the occupation time is $|T_i^K| = \sum_{k=1,\ldots,K} (t_i^k - s_i^k)$.

The requirements for a good scheduling algorithm can be expressed as:

(1) Collision-free: signals from different TXs never overlap in the receiver; according to (1), this means

$$\bigcap_{i \in V} T_i^K = \emptyset, \quad K = 1, 2, \ldots. \tag{6}$$

(2) 100% Throughput: the arrival rate at the receiver equals the aggregated arrival rate at the TXs under admissible traffic and is 1 in the case of overload:

$$\lim_{K \to \infty} \frac{\sum_{i \in V} |T_i^K|}{\max_{i \in V} t_i^K} = \min\left(1, \sum_{i \in V} a_i\right), \tag{7}$$

(3) Fairness: each TX is guaranteed a bandwidth of 1/N:

$$\lim_{K \to \infty} \frac{|T_i^K|}{\max_{i \in V} t_i^K} \geq \min\left(\frac{1}{N}, a_i\right), \quad i \in V. \tag{8}$$

3.2. Related work and basic ideas

Although several scheduling algorithms for multi-queue one-server systems have been proposed (e.g., generalized processor sharing (GPS) [15], weighted fair queuing (WFQ) [4] and deficit round robin (DRR) , they address the case without propagation delay, where the scheduler determines the next slot transmission based on current queue information. On the other hand, our problem deals with a non-negligible delay, whereby the scheduler has to depend on the old queue information to determine future transmissions. Thus, existing algorithms are not applicable to our problem, as depicted in Fig. 8.

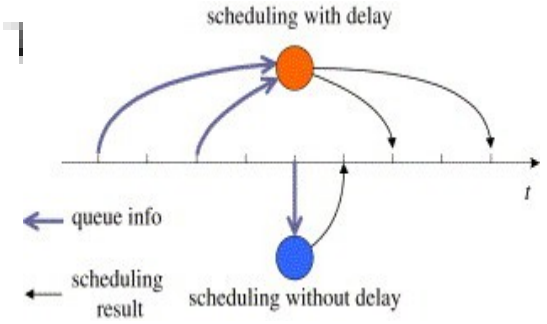


Fig. 8. Scheduling with/without propagation delay.

A similar issue has been investigated in the context of an Ethernet passive optical network (EPON), in which multiple optical network units (ONUs) are connected to a single optical link terminal (OLT). The OLT broadcasts to all the ONUs, while the upstream optical channel is shared among the ONUs. A framework called a multi-point control protocol (MPCP) has been developed by the IEEE 802.3ah Task Force for upstream media access [14], and a number of dynamic bandwidth allocation (DBA) algorithms have been proposed to control the transmission of each ONU [1], [13]. Due to the short length of the fibers in EPON (usually several miles), the propagation delay between the OLT and each ONU is in the microsecond range, and thus the transmission can be assigned on-demand, for example, by polling [10]. However, metro networks usually span a much greater distance, which drives us to develop a new scheduling algorithm for VFC networks.

The basic ideas behind our algorithm can be stated as follows:

(1) In the case of a light load, it is important to allocate the bandwidth proactively among the TXs rather than have pure on-demand transmission time assignment, whereby each packet has to experience the propagation delay of sending the queue length to the scheduler and waiting for feedback.

(2) Under a heavy load where the aggregate queue length in the transmitters remains non-zero for a long time, keeping the server in the busy state guarantees 100% throughput, which means the scheduler has to consider the propagation delay and carefully arrange the medium access switchover from one TX to another, so that the bandwidth is fully utilized, i.e., the backlog of the first TX is not emptied before the switchover point.

(3) With non-uniform traffic, a greedy TX is eligible to obtain excess bandwidth not used by the other TXs; once a normal TX is found to be insufficiently served, the greedy TX is punished by reassigning some of the bandwidth to the normal one.

3.3. Definition of a slot-channel

According to Section 2, the transmission of signals is based on the unit of a slot. Without loss of generality, we set the slot length to 1 and denote the time interval $[m, m+1),\ (m = 0, 1, \ldots)$ in the receiver as slot m; thus, a wavelength can be divided into N slot-channels, as defined below.

Definition: Given an index n(n=0,1,…,N−1), slot-channel CH_n is defined to include a series of periodic slots n+kN,k=0,1,2,….

For simplicity, we use the word channel for slot-channel in the following discussion. It is clear that two TXs never collide with each other as long as they transmit on different channels. Once CH_n is assigned to TX i, super-packets from the TX arrive at the receiver in slots n+kN (for k=K,K+1,…,K+M) periodically, where K and K+M are the beginning and ending period determined by the scheduler. Based on slot-channels, a good scheduling algorithm should be designed according to the following principles:

(1) At any time a channel is assigned to at most one TX to avoid medium access collision;

(2) Channels are dynamically reassigned among the TXs for adaptive and fair bandwidth allocation, where reassignment is based on the queue states of the transmitters; and

(3) To guarantee bandwidth utilization, switching channel access from one TX to another needs to be contiguous (i.e., no slot is left unused during the switchover), which requires careful consideration of the propagation delay.

Fig. 9 shows a tree with nodes 1–4 sending to node 5, and the wavelength is divided into four channels: a,b,c and d. TXs 1 and 4 are assigned to channels d and b, respectively; TX 2 is under a heavy load and occupies two channels: a and c; TX 3 is idle and has no occupation.

3.4. Weight and state of each transmitter

Generally speaking, TXs with long queues require more bandwidth. However, a non-negligible propagation delay prevents the scheduler from obtaining up-to-date queue lengths, which may lead to a mismatch between the bandwidth requirement and the real allocation. In the proposed algorithm, a weight is first calculated for each TX according to its most

recent queue length and the propagation delay between the TX and the scheduler; then the TX is classified as a certain state, according to the value of its weight. Channel allocation among the competing TXs is carried out based on their weights and states, which indicate whether a TX occupies excessive or insufficient bandwidth.

At time t, we denote the queue length and occupied channel number of TX i as $Q_i(t)$ and $C_i(t)$, respectively. With a predetermined threshold H_i

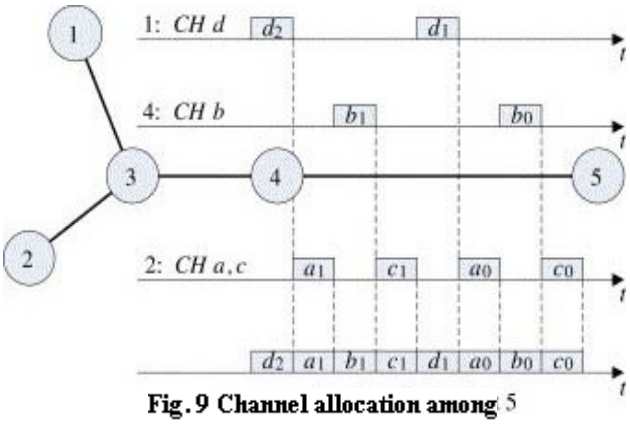


Fig. 9 Channel allocation among 5 competing TXs.

(which increases with the propagation delay and is explained later), an expected channel occupation is derived for the TX according to the following definition:

$$C_i^T(t) = \max\left(N, \left\lfloor \frac{Q_i(t)}{H_i} \right\rfloor\right), \quad i \in V, \tag{9}$$

where the Gaussian function $\lfloor x \rfloor$ is the maximum integer less than or equal to x.

With $C_i(t)$ and $C_i^T(t)$, the weight of TX i is calculated according to Table 1 and the state is derived accordingly. Comparison between $C_i^T(t)$ and $C_i(t)$ indicates whether the bandwidth assigned to TX i is excessive or insufficient. It is worth noting that the case in which a TX occupies either no or a single channel is specifically considered to avoid starvation and achieve low latency, which is explained in the following section.

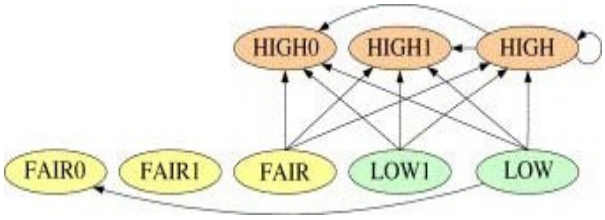


Fig. 10. Channel reassignment between different TXs.

Table 1. State and weight calculation

Condition	Weight	Range	State
$C_i^g > C_i = 0$	∞	∞	HIGH0
$C_i^g > C_i = 1$	C_i^g	[2,N]	HIGH1
$C_i^g > C_i > 1$	C_i^g/C_i	$(1+\frac{1}{N-1}, \frac{N}{2})$	HIGH
$C_i^g - C_i = 0$	$1+\frac{1}{N}$	$1+\frac{1}{N}$	FAIR0
$C_i^g - C_i = 1$	$1+\frac{1}{N+1}$	$1+\frac{1}{N+1}$	FAIR1
$C_i^g - C_i > 1$	1	1	FAIR
$C_i^g < C_i = 1$	0	0	LOW1
$C_i^g < C_i \neq 1$	$C_i^g - C_i$	[−N,−1]	LOW

3.5. Channel reassignment

achieve high throughput, it is necessary to allocate more channels to the TXs with high weight; to guarantee fairness, no starvation should be introduced to TXs with non-empty queues; at the same time, a reassignment policy should be designed so as to achieve a low delay in the case of a light load. The details of the proposed policy are listed below and illustrated in Fig. 10. where an arrow from A to B means channel reassignment from a TX in state A to another TX in state B is allowed.

(1) To favor the TXs under a heavy load, channel reassignment takes place only from low-weight TXs to high-weight ones.

(2) To avoid starvation of non-empty queues, a TX in state HIGH1 or FAIR1 is never deprived of its single occupation and a TX in HIGH0 is always able to obtain a reassignment.

(3) To avoid excessive bandwidth allocation, a TX in state FAIR1, FAIR, LOW1 or LOW is never assigned a new channel.

(4) To achieve good delay performance, a TX in state FAIR0 is also eligible for channel assignment in the case of a light load, and thus a newly arrived packet is able to be transferred immediately.

(5) When TX i is in state FAIR or HIGH and one of its channels is assigned to j, it must satisfy the condition that the sum of the two TX weights is decreased after the reassignment. This assures fairness by preventing greedy traffic from being assigned too many channels. For example, suppose TX i is subject to greedy traffic and is in state HIGH with $C_i^g = 10, C_i = 8$, and TX j experiences normal traffic and is in state FAIR with $C_j^g = 2, C_j = 2$. In this case, channel reassignment from j to i is not allowed and normal traffic is protected from greedy traffic.

3.6. Threshold calculation

Suppose the scheduler sends out a command at time T to reassign CH_n from j to i, as illustrated in Fig. 11. Due to the propagation delay, the first signal from i is transmitted no earlier than $T+\delta_i$. From the time the scheduler issues the reassignment to the time when the first signal from i is sent out, we say CH_n is locked by i. A locked channel is not eligible for another reassignment until it is unlocked. In Fig. 11, CH_n is locked by i during $[T,T_i)$.

For simplicity, we assume the channel in Fig. 11 is time-continuous, which does not affect the correctness of our discussion. At the receiver, the first signal from i arrives at $T+\delta_i+d_i$, and the arrivals from j do not stop until $T+\delta_j+d_j$ or later. Combining these, the first transmission time for i is set as:

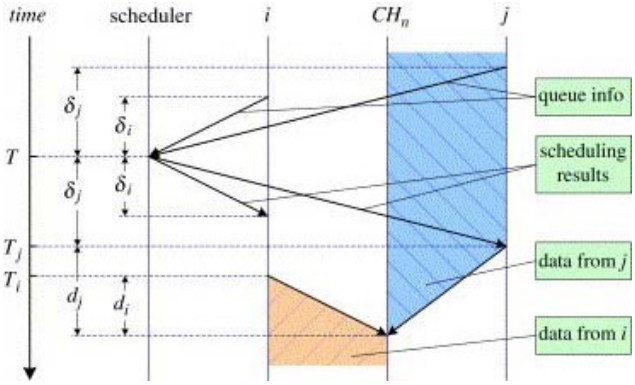


Fig. 11. Details of channel reassignment.

$$T_i = T + \max(\delta_i + d_i, \delta_j + d_j) - d_i, \quad (10)$$

and the stop time for

$$T_j=T+\max(\delta_i+d_i,\delta_j+d_j)-d_j. \quad (11)$$

The above configuration guarantees conflict-free channel reassignment, in which signals from a newly assigned TX will not overlap with those from the old one. To achieve 100% throughput under a heavy load, it is also necessary to ensure that no slot is wasted during each channel reassignment, which means j does not empty its queue before T_j in Fig. 11. If the system is under a heavy load, the release of CH_n from j takes place as soon as j occupies more channels than the expected number. Note that the weight of j is rounded with H_j; it is easy to see that a large enough value for H_j will guarantee a seamless reassignment. However, a large threshold introduces a long delay, which makes it necessary to determine the lower bound of H_j. Since the bandwidth of each channel is 1/N, we only need to ensure that:

$$H_j\geq(\delta_j+(T_j-T))/N=(\delta_j+\max(\delta_i+d_i,\delta_j+d_j)-d_j)/N. \quad (12)$$

Note that (12) is obtained assuming a time-continuous channel, while the channel slots are actually periodic, and thus the lowest threshold guaranteeing 100% throughput in WSCS is:

$$H_j^* = 1 + (\delta_j + \max_{i \in V} (\delta_i + d_i) - d_j)/N, \quad j \in V, \quad (13)$$

which is called the critical threshold.

Using a threshold lower than the critical threshold reduces the waiting time for a TX to obtain a channel assignment and improves the delay performance. Note that the throughput is not deteriorated under a light load, and thus adjusting the threshold adaptively to the network load is a good choice. In WSCS, the load can be reflected by the maximum unified queue length:

$$\rho^*(t) = \max_{i \in V} \frac{Q_i(t)}{N \cdot R_i^*}, \quad (14)$$

and threshold adjustment is carried out like a Schmidt trigger, as shown in Fig. 12. The value of H_i is initialized to 1, then changed to H_i^* once ρ^* increases over 0.7, and is set back to 1 if ρ^* drops below 0.3. It is shown in Section 4 that this approach yields good delay performance, as well as 100% throughput.

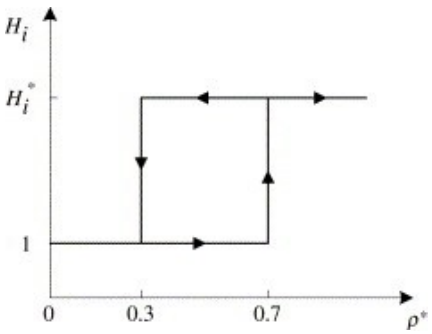


Fig. 12 Adaptive adjustment of the threshold

3.7. Scheduling algorithm

The scheduler is activated in each slot to perform multi-cycle scheduling, where each cycle searches for a pair of TXs with the maximum and minimum weight and determines whether to reassign a channel from the latter to the former. The detailed operations are:

(1) Mark each locked channel as unlocked if its new occupant has started transmission;(2) Perform threshold adjustment according to
(2) the system load;

(3) Calculate the state and weight of each TX;

(4) Find a TX i with the maximum weight; if there are multiple candidates with the same weight, choose one of them randomly;

(5) Find a TX j with the minimum weight and an unlocked channel CH_n; if there are multiple candidates, choose one of them randomly. If such a TX is not available, exit; otherwise go to the next step; and

(6) If the reassignment of CH_n from j to i does not comply with the policies in Section 3.5, the algorithm exits; otherwise issue the reassignment command, update the states and weights of i and j, and return to Step (4).

As illustrated in Fig. 13, the state of a TX is changed by both scheduling and traffic arrival/departure. By performing channel reassignment, the scheduler tries to maintain a balance between the arrivals and departures for each TX. For instance, a large volume of arrivals tends to push a TX into state HIGH; meanwhile, the scheduler tries to drag the TX into state FAIR by assigning it more channels. Several features of the algorithm are listed below:

• A TX with no assignment has the highest priority to obtain a channel once its queue length exceeds the threshold, and thus starvation is avoided;

• Under a light load, all the TXs tend to remain in FAIR1 or LOW1, which is similar to fixing each TX to a certain channel, and thus good delay performance is achieved, since a new arrival is transmitted without

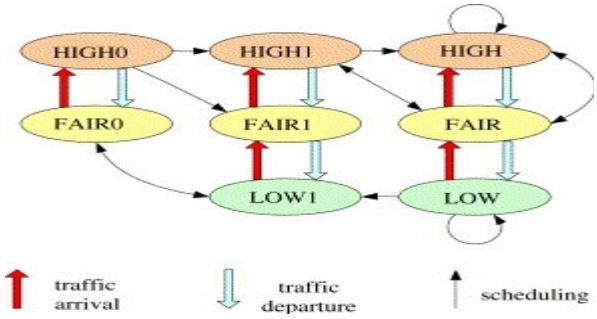


Fig.13 State transfer of a single TX

requesting the scheduler for channel reassignment; and

• A TX with a heavy load tends to obtain more service without blocking lightly loaded ones, which achieves high bandwidth utilization and fairness guarantee simultaneously.

3.8. Throughput performance

Theorem 2

WSCS achieves 100% throughput under arbitrary admissible traffic.

Proof

Consider the case for which $Q_j(t)\to\infty$; according to (14), the system is under a heavy load and all the TXs use the critical thresholds. In this case, channel reassignment is without loss, in that each slot is filled with a super-packet, which means the aggregated service rate of the queues is 1. Under admissible traffic, the aggregated arrival rate $\sum_{i\in N}\alpha_i \le 1$

Theorem 3

With N TXs, WSCS guarantees each one a bandwidth of $\frac{1}{N}$ under arbitrary traffic.

Proof

Similar to the proof of Theorem 2, consider the case in which $Q_j(t)\to\infty$; according to WSCS, TX j is assigned at least one channel, and thus between two continuous slots there must be:

$$\lim_{Q_j(t)\to\infty} E\{Q_j(t+1) - Q_j(t) \mid Q_j(t)\} \le \alpha_j - \frac{1}{N},$$
$$\forall j \in V.$$

In the case of $\alpha_j \le \frac{1}{N}$, the queue length is under control and all the arrivals to TX j will be fully served. When $\alpha_j > \frac{1}{N}$, the allocated service depends on the load of other TXs, but a minimum bandwidth of $\frac{1}{N}$ is always guaranteed. This completes the proof.

3.9. Comparison with the distributed scheduling in TWIN

As briefly described in Section 1.1, two distributed algorithms are presented in SBS and DBS. In SBS, each source node decides its own transmissions independently, thus their signals may collide at the destination, which results in packet loss and throughput degradation. SBS is similar to slotted aloha in that the probability of collision can be 66% under heavy load or even higher in case of overload. On the other hand, DBS avoids collisions at destinations by granting only one source node for transmission. However, the contentions at source nodes still reduces the network throughput significantly. The analysis and simulations in show that the throughput of a 10-node network is approximately 65% under heavy load. In addition, each packet has to wait for the request-grant procedure before being sent out, which results in a packet delay that is at least three times of the propagation delay regardless of the load.

In contrast, the WSCS algorithm proposed in this paper achieves 100% throughput. At the same time, the delay is much lower than DBS since there is no waiting time for grants. Simulations in the next section show that the packet delay increases slowly from light to medium load, and the value under light load is dominated by the propagation delay.

4. Performance evaluation

This section evaluates the delay performance of WSCS using simulations. We choose a 10 Gb/s WDM network and set the slot size as 50 μs, which contains approximately 40 IP packets of the maximum size (1500 bytes). Since the performance of a wavelength tree is independent of the others, we only need to consider the performance of a single tree. Two models are considered in our simulations:

• Small network: contains 16 TXs and a receiver, and the propagation delays d_i and δ_i are randomly generated between 100 and 1500 μs. Since the light speed in the fiber is approximately 2×10^8 m/s, this model represents a typical metro network spanning a distance of approximately 300 km.

• Large network: contains 32 TXs and a receiver, and the propagation delays are randomly generated between 500 and 5000 μs. The network covers a much larger area (1000 km) than MAN and is used to verify the performance of WSCS in the case of a long propagation delay.

In our simulation, traffic arrivals to each queue are based on super-packets and are generated with a Bernoulli source. Two traffic distributions are adopted to examine uniform and non-uniform loads, respectively:

(1) Uniform: the network load ρ increases from 0.1 to 1, where the arrival rate of each queue is equally set to:

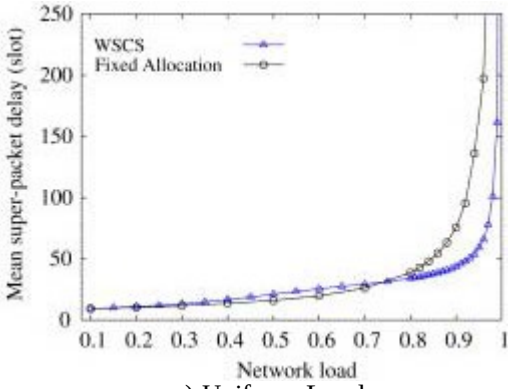
$$a_i = \frac{1}{N}\rho, \quad i \in V; \tag{15}$$

(2) Non-uniform: the first queue is heavily loaded with a factor of h(h ∈ [0,1]), and the arrival rate for each queue is:

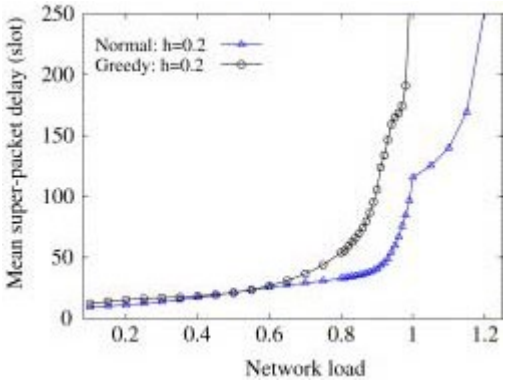
$$a_i = \begin{cases} h\rho + \dfrac{1-h}{N}\rho & \text{if } i = 1 \\ \dfrac{1-h}{N}\rho & \text{if } i \neq 1 \end{cases}, \quad i \in V. \tag{16}$$

In this case, TX 1 is subject to greedy traffic, while TXs 2–N experience normal traffic. Since the arrival rate for normal traffic may be less than 1/N, even if the network is overloaded, our simulations are also performed under the condition ρ>1 to examine the overload case.

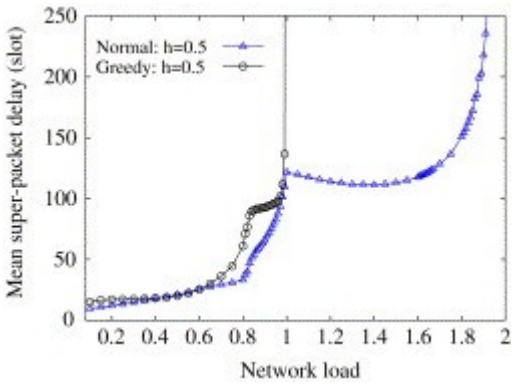
To show the delay performance of WSCS clearly, the fixed part, propagation delay d_i ($i \in V$), is removed from the results. Under uniform traffic, WSCS is compared with a fixed channel allocation where each CH_i is dedicated to TX i. Under non-uniform traffic, greedy and normal traffic are measured separately and the results under h=0.2,0.5,0.8 and 1 are presented. It is worth noting that h=1 means only TX 1 has a non-zero load and there is no curve for normal traffic. Fig. 14 shows the results of the small network and Fig. 15 is obtained from the large network.



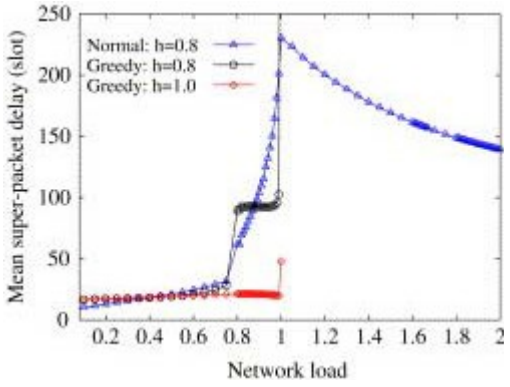
a) Uniform Load



(b) Non-uniform load, h=0.2.

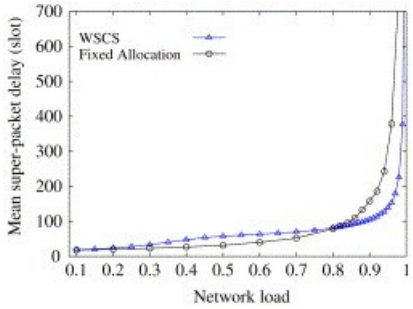


(c) Non-uniform load, h=0.5.

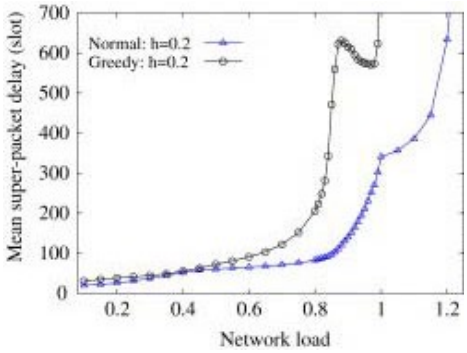


(d) Non-uniform load, h=.8, 1.

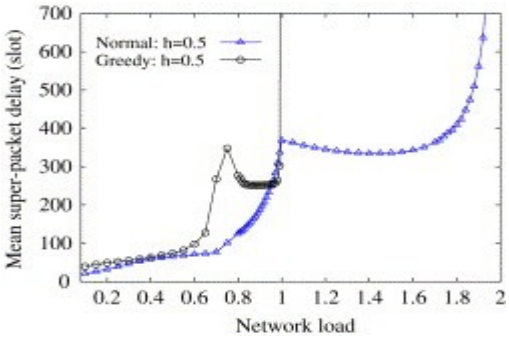
Fig. 14. Delay performance of a small network.



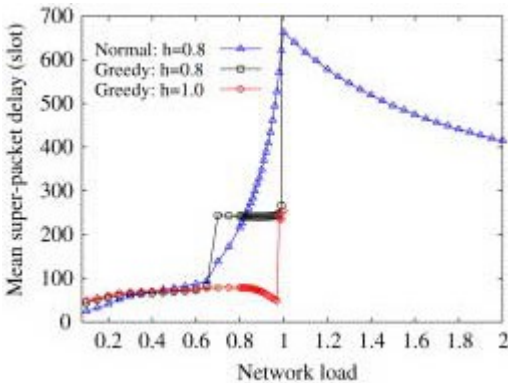
(a) Uniform load.



(b) Non-uniform load, h=0.2.



(c) Non-uniform load, h=0.5.



(d) Non-uniform load, h=.8, 1.

Fig. 15. Delay performance of a large network.

From the results several characteristics of the WSCS algorithm can be observed:

(1) Low delay is achieved under light to medium load (e.g., $\rho<0.6$) with both uniform and non-uniform traffic.

(2) When the network is heavily loaded with uniform traffic, WSCS outperforms fixed allocation, since the former performs adaptive bandwidth allocation according to the traffic fluctuation.

(3) Under a non-uniform load, normal traffic is well served and is guaranteed a 1/N bandwidth, even if the network is overloaded, e.g., with h=0.5, the delay approaches ∞ with $\rho\rightarrow 2$, where the arrival rate of normal traffic approaches 1/N. At the same time, greedy traffic is fully served only when $\rho<1$.

(4) The WSCS performance is stable for changes in network size.

In the case of a non-uniform load, as long as the thresholds are adjusted to be smaller than the critical values, normal traffic can easily accumulate enough packets to obtain a channel, even if its arrival rate is lower than for greedy traffic. This leads to a slow increase in delay for normal traffic under a medium network load, e.g., the curves in Fig. 15(b) with ρ changing from 0.6 to 0.88.

When the network load continues to increase, critical thresholds are used, which increases the waiting time for normal traffic to be assigned a channel, and the delay for normal traffic increases quickly. On the other hand, the delay for greedy traffic increases slowly, and may even decrease in the case of large critical thresholds, as shown in Fig. 15(b) when ρ changes from 0.88 to 0.98.

Under highly non-uniform traffic (e.g., h=0.8), the delay for normal traffic may decrease when $\rho>1$, as shown in Figs. 14(d) and 15(d). This can be explained as follows. When $\rho=1$, the arrival rate for normal traffic is much lower than the guaranteed bandwidth 1/N, and an increase in arrival rate remarkably reduces the time used to accumulate enough super-packets to trigger a channel assignment; thus, the mean delay is decreased to a certain degree.

A possible approach to improve the delay performance for normal traffic under a heavy non-uniform load is to start a timer once the queue for a TX is non-empty; upon timeout, a channel is assigned to the TX, even if its queue length is still below the critical threshold. In this case, the delay performance is improved with some sacrifice of the bandwidth utilization; the details of this issue are left for future work.

It is proved in Theorem 3 that WSCS provides fairness on bandwidth allocation by giving each TX 1/N of the total link capacity. This is also verified in Fig. 14 and Fig. 15, where each node is guaranteed of the bandwidth regardless of its load and propagation delay.

Our simulations also reveal that the algorithm provides fairness on delay performance in that the difference of the queueing delay among different nodes is trivial, where the queueing delay is equal to the total delay minus the propagation delay. This property means that nodes close to the destination or scheduler do not get any privilege over those located far away, compares the average queueing delay of the flows originated from all the 32 nodes, where the traffic distribution is uniform and the load is increased from 0.1 to 0.9. The propagation delays from each node to the destination and the scheduler are randomly generated. It can be seen that although the propagation delays are quite different, the queueing delays are roughly the same.

5. Conclusion

A virtual fully connected (VFC) architecture is proposed for low-cost and high-performance WDM metropolitan area networks. VFC architecture avoids the use of expensive routers by providing single-hop transportation between node pairs; at the same time, it can be combined with a mesh physical topology to achieve high survivability under multiple failures. Medium access control and dynamic bandwidth allocation are realized by a scheduling algorithm called weighted slot-channel scheduling (WSCS), which provides 100% throughput and fairness guarantee under arbitrary traffic scenarios. Simulations show that a VFC network yields good delay performance under both uniform and non-uniform loads.

Based on the VFC architecture, there are several interesting issues to investigate in our future work. One such issue is to improve the efficiency by considering the multi-hop extension of the architecture with traffic grooming, such that when a node does not have much incoming traffic, its wavelength tree can be used to relay traffic to other nodes. Another issue is to increase the wavelength resource utilization. In the present architecture, the wavelength channels not occupied by the trees are unused until they are employed for failure recovery. By modifying the node architecture and network control schemes, such a resource may be used to carry traffic as well, which will definitely improve the resource efficiency. In addition, further investigation of the detailed protection/restoration scheme is also worthwhile such that the nodes and the schedulers can be coordinated for fast failure recovery, where certain proactive calculations and configurations are believed to be helpful.

References

[1] C.M. Assi, Y. Ye, S. Dixit and M.A. Ali, Dynamic bandwidth allocation for quality-of-service over Ethernet PONs, IEEE J. Select. Areas Commun. 21 (2003) (9), pp. 1467–1477. Abstract-INSPEC | Abstract-Compendex | Abstract-Compendex | Full Text via CrossRef | Abstract + References in Scopus | Cited By in Scopus

[2] H.J. Chao, K. Deng and Z. Jing, Petastar: a petabit photonic packet switch, IEEE J. Select. Areas Commun. 21 (2003) (7), pp. 1096–1112. Abstract-Compendex | Abstract-INSPEC | Full Text via CrossRef | Abstract + References in Scopus | Cited By in Scopus

[3] I. Chlamtac and A. Gumaste, Light-Trails: a solution to ip centric communiction in the optical domain, Proc. 2nd Intl. Workshop on Quality of Service in Multiservice IP Networks QoS-IP'03, Springer-Verlag, Heidelberg (2003), pp. 634–644. Abstract-INSPEC

[4] A. Demers, S. Keshav, S. Shenker, Design and analysis of a fair queueing algorithm, in: Proc. ACM SIGCOMM'89, September 1989, pp. 1–12.

[5] J. Geske, Y. Okuno, D. Leonard and J. Bowers, Long-wavelength two-dimensional WDM vertical cavity surface-emitting laser arrays fabricated by nonplanar wafer bonding, IEEE Photon. Technol. Lett. 15 (2003) (2), pp. 179–181. Abstract-Compendex | Full Text via CrossRef | Abstract + References in Scopus | Cited By in Scopus

[6] M.S. Goodman, H. Kobrinski, M.P. Vecchi, R.M. Bulley and J.L. Gimlett, The LAMBDANET multiwavelength network: architecture, applications and demonstrations, IEEE J. Select. Areas Commun. 8 (1990) (6), pp. 995–1004. Abstract-Compendex | Abstract-INSPEC | Full Text via CrossRef | Abstract + References in Scopus | Cited By in Scopus

[7] I. Jang and S. Lee, Simple approaches of wavelength registration for monolithically integrated DWDM laser arrays, IEEE Photon. Technol. Lett. 14 (2002) (12), pp. 1041–1135.

[8] J.P. Jue, M.S. Borella and B. Mukherjee, Performance analysis of the rainbow WDM optical network prototype, IEEE J. Select. Areas Commun. 14 (1996) (5), pp. 945–951. Abstract-Compendex | Abstract-INSPEC | Full Text via CrossRef

[9] D.R. Karger, P.N. Klein and R.E. Tarjan, A randomized linear-time algorithm to find minimum spanning trees, J. ACM 42 (1995) (2), pp. 321–328. Abstract-Compendex | Abstract-INSPEC | MathSciNet | Full Text via CrossRef | Abstract + References in Scopus | Cited By in Scopus

[10] G. Kramer, Interleaved polling with adaptive cycle time (IPACT): a dynamic bandwidth distribution scheme in an optical access network, Photonic Net. Commun. 4 (2002) (1), pp. 89–107. Abstract-INSPEC | Full Text via CrossRef | Abstract + References in Scopus | Cited By in Scopus

[11] M. Maier, Architecture and access protocol for a wavelength-selective single-hop packet switched metropolitan area network, Ph.D. dissertation, Tech. Univ. of Berlin, 2003.

[12] H. Masuura, Y. Watanabe, M. Kagawa, H. Kazami, K. Ida and N. Sato, An optical combiner module for DWDM systems, Furukawa Rev. (2002) (21), pp. 17–21.

[13] M.P. McGarry, M. Maier and M. Reisslein, Ethernet PONs: a survey of dynamic bandwidth allocation (DBA) algorithms, IEEE Netw. 42 (2004) (8), pp. S8–S15. Abstract-Compendex | Full Text via CrossRef | Abstract + References in Scopus | Cited By in Scopus

[14] I.D. P802.3ah/D1.0TM, Media access control parameters, physical layers and management parameters for subscriber access networks. IEEE 802.3ah Task Force, Aug. 2002.

[15] A.K. Parekh and R.G. Gallager, A generalized processor sharing approach to flow control in integrated services networks: the single node case, IEEE/ACM Trans. Networking 1 (1993) (3), pp.

344–357. Abstract-Compendex | Abstract-INSPEC | Full Text via CrossRef | Abstract + References in Scopus | Cited By in Scopus

K.V.S.S.S.S.SAIRAM (s5kanduri@rediffmail.com) is working as Senior Associate Professor, ECE Department, Bharat Institute of Engineering & Technology, Mangalpally, Ibrahimpatnam, Hyderabad, Andhra Pradesh State, IDIA. He was previously worked as Lecturer and Assistant Professor in Dr. M.G.R. Deemed University, Chennai. He is pursuing his Ph.D (Optical Communications) under the guidance of Dr. P.V.D Somasekhar Rao and Dr. T. Janardhana Rao, UGC-ASC Director, J.N.T.University, Kukatpally, Hyderabad - 72 & Professor &HOD of the ECE Department, Sridevi Women's Engineering College, V.N.Pally, Gandipet, Hyderabad- 75. He got his Bachelors Degree in ECE from Karnataka University, Dharwad in 1996 and Masters Degree from Mysore University, Mysore in 1998.His research interests are Optical Communication, Networking, Switching and Routing and Wireless Communication. He was published 30 PAPERS in IEEE Communication Magazine, IEEE Potentials, International and National Conferences. He is an IEEE REVIEWER and EDITORIAL MEMBER for Optical Society of America, Journal on Photonics and IEEE Journal on Quantum Electronics and IASTED.

Dr.P.V.D.SomasekharRaoB.E.(SVU), M.Tech.(IIT, Kharagpur), Ph.D. (IIT, Kharagpur. Professor and Head of the Department & UGC-ASC Director Specialized in Microwave and Radar Engineering. His research interests include Analysis and design of Microwave circuits, Antennas, Electro Magnetics, and Numerical Techniques. He published 20 research papers in National and international Journals and Conferences. He is presently guiding two Ph.D. students. He prepared the source material for School of Continuing and Distance Education, JNTU, in the subjects such as computer programming & Numerical Techniques, Radar Engineering, Antennas and Propagation and Microwave Engineering. He has more than 20 years of teaching and research experience, which include R&D works at Radar Centre, IIT Kharagpur and at Radio Astronomy centre and TIFR. He is a Senior Member of IEEE, Fellow of IETE. He delivered a number of invited lectures. He is a reviewer for the Indian Journal of Radio & Space Physics from 1991. He is the recipient of the IEEE -USA outstanding Branch Counselor/Advisor award for the year 1993-94. He had completed a number of projects aided by AICTE. He has been a visiting faculty at Assumption University, Bangkok, during 1997-99.

Dr. T. Janardhana Rao is working as Professor and Head of the Department in Sridevi women's Engg College, V.N.Pally, Gandipet, Hyderabad, Andhra Pradesh State, INDIA. His research interests include Optical Networks, Digital Electronics, Biomedical Engg., & Power Electronics. He published 15 papers in national and international Journals and Conferences. Professor Rao was a former member of the faculty of S.V.University with a teaching experience of about 45 years. He is a life member of ISI and ISTE.