PAPER

Buffer Sharing in Conflict-Free WDMA Networks

Ming CHEN[†], *Nonmember and* Tak-Shing Peter YUM[†], *Member*

SUMMARY A Wavelength Division Multiaccess (WDMA) network with buffer sharing among stations is studied. All stations in the network are connected to a passive optical star coupler and each station has a different fixed wavelength laser for transmitting packets. Each station in the network reports its packet backlog to a scheduler which computes and then broadcasts a transmission schedule to all the stations through a control channel in each time slot. A transmission schedule includes two types of assignments: 1) assign a maximum number of stations for conflict-free transmissions, and 2) assign the relocation of packets from congested stations to uncongested relaying stations through idling transceivers for distributed buffer sharing. The first assignment aims at maximizing throughput and the second assignment aims at minimizing packet loss. Simulation results show that as much as 75% of the buffers can be saved with the use of buffer sharing when 50% of the packets are of the non-sequenced type.

key words: *WDMA networks, transmission scheduling, buffer sharing*

1. Introduction

Protocols for Wavelength division multiaccess (WDMA) networks have been extensively studied [1]-[9]. These protocols in general fall into three types: Fixed Assignment, Demand Assignment, and Random-Access. An example of fixed assignment protocol was proposed in [3] where different wavelength channels are slotted and are assigned to stations in a time-divisioned fashion. Fixed assignment protocols are simple and have no destination conflict problem. The maximum network throughput, however, is limited as transmission slots are permanently assigned to all stations. Random-access protocols [6], [7] are known to have lower network throughput, and suitable only for LANs due to the degrading effects of the propagation-delay to message-transmission-time ratio [12]. The dynamic-time WDMA (DT-WDMA) protocol [4] is an example of demand assignment protocols. Here a station broadcasts its transmission intention before actual transmission so that the destination station will know which wavelength to use to receive the packet. The network throughput, however, is limited by the retransmissions when packet destination conflicts occur. The protocol proposed in [5], uses a near optimal scheduling algorithm to avoid packet destination conflicts by scheduling only those packets with different destination addresses for transmissions in each slot. This protocol can therefore achieve a higher network throughput than that in [4]. Another similar protocol [8] uses a random selection scheme for determining the schedule. Recent survey papers [10], [11] give more general and detail review on these and other WDMA protocols.

In this paper, we propose a protocol with both conflict-free and buffer sharing features for WDMA networks. The network has a scheduler which collects packet backlog information and computes transmission schedules. These schedules are then broadcasted to the stations in the network for conflict-free transmissions. A distributed design of the conflict-free network is possible [5]. The centralized version presented in this paper has the advantage of lower station complexity as a high speed processor for the computation of the transmission schedules is needed only at the scheduler. It, however, introduces one more station-to-station propagation delay for broadcasting the transmission schedules.

Packet blocking occurs when a station has used up all its buffers. This need not be the case when some other stations have spare buffers. The second feature of the protocol is the ability to share buffers among stations. The trick is to transmit some packets in congested stations (those lacking buffers) to uncongested stations (those having spare buffers). This, however, has to be done without sacrificing the network throughput. As we will show later this dynamic buffer sharing mechanism can actually give a slight increase of throughput because a more efficient transmission schedule can be obtained from a more balanced packet load distribution on the stations. The major benefit, however, is the reduction of packet loss due to buffer overflow. Results in Sect. 6 show that as much as 75% of the buffers can be saved with buffer sharing when 50% of the packets are of the non-sequenced type.

2. Network Architecture

Figure 1 shows an optical WDMA network with N

Manuscript received October 12, 1993.
Manuscript revised March 22, 1994.

[†] The authors are with the Department of Information Engineering, The Chinese University of Hong Kong, Shatin NT, Hong Kong.

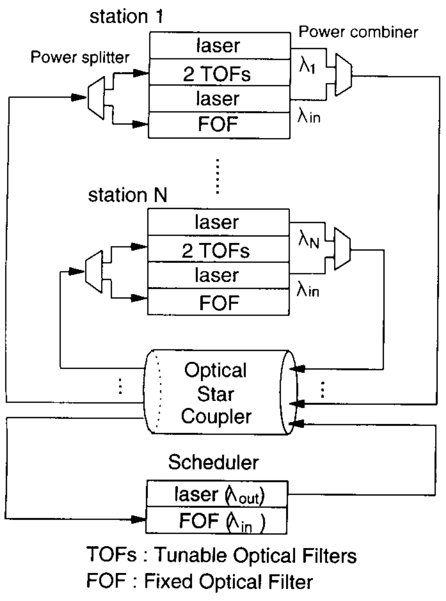


Fig. 1 A centrally controlled WDMA network.

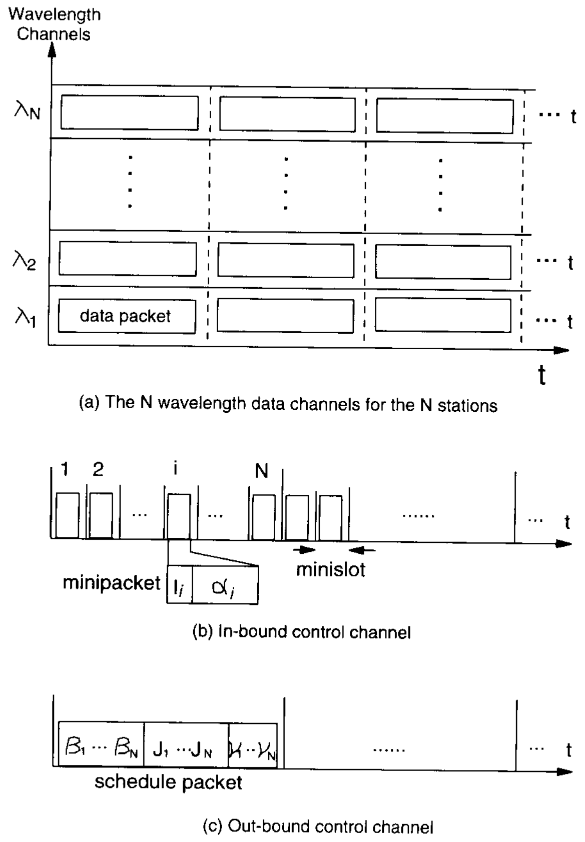

Wavelength Channels
λN
λ2
λ1
data packet
t
(a) The N wavelength data channels for the N stations
1 2 i N
minipacket
I_i α_i
minislot
(b) In-bound control channel
$\beta_1 \cdots \beta_N$ $J_1 \cdots J_N$ $\gamma_1 \cdots \gamma_N$
schedule packet
(c) Out-bound control channel

Fig. 2 Data channel and control channel formats.

stations and a scheduler connected by an $(NH) \times (N+1)$ star coupler. Station i $(i=1, 2, \cdots, N)$ has a laser diode for transmitting data packets at wavelength λ_i and another laser diode at wavelength λ_{in} ("*in*" denotes in-bound to the scheduler) for reporting packet backlog information to the scheduler. In addition, a station also has two tunable optical filters for selecting the desired data channel and a fixed optical filter at λ_{out} for receiving transmission schedules from the scheduler. The two tunable filters work alternately so that the tuning of one filter can be performed while the other is receiving a packet. The scheduler has one laser diode operating at wavelength λ_{out} for broadcasting transmission schedules and one fixed filter at λ_{in} for receiving backlog information. Channels at $\lambda_1, \lambda_2, \cdots, \lambda_N$ are called data channels, and channels at λ_{in} and λ_{out} are called the in-bound channel and the out-bound channel of the scheduler respectively.

The scheduler maintains synchronization among the stations through its out-bound channel. Let channels be slotted such that each slot can accommodate one data packet (Fig. 2). We assume that the propagation delay between stations is adjusted to be identical among all stations and is equal to one time slot. Each slot in the in-bound channel is further divided into N minislots and are assigned one to each of the N stations.

The network classifies packets into the *sequenced* and the *non-sequenced* types. Packet sequence needs to be maintained for the sequenced packets having the same destination while the non-sequenced packets do not have such a requirement. As the packet sequence may be disrupted with the routing of packets from the source to a relaying station in buffer sharing, only non-sequenced packets are eligible for such relocation. Those non-sequenced packets that get relocated to relaying stations are called *relocated* packets.

3. Network Protocol

3. 1 Minipacket and Schedule Packet Format

Each station reports the arrival of a new packet to the scheduler by writing a *minipacket* into its assigned minislot in every slot (Fig. 2). A minipacket from station i contains two fields:

• a 2 bit *type* field I_i defined as

$$I_i = \begin{cases} (0,0) & \text{no packet arrival} \\ (1,0) & \text{a sequenced packet arrival} \\ (0,1) & \text{a non-sequenced packet arrival} \end{cases}$$

• a $\lceil \log_2 N \rceil$ bit *destination address* field "α_i".

In each slot, the scheduler computes a transmission schedules and broadcasts them on the out-bound channel as a *schedule packet* to all stations (Fig. 2). The schedule packet has the format $[\beta_1 \cdots \beta_N, J_1 \cdots J_N, \gamma_1 \cdots \gamma_N]$, where

• β_i is the address to which station i should make a transmission,

• J_i is a 2-bit indicator of the packet type to be transmitted by station i

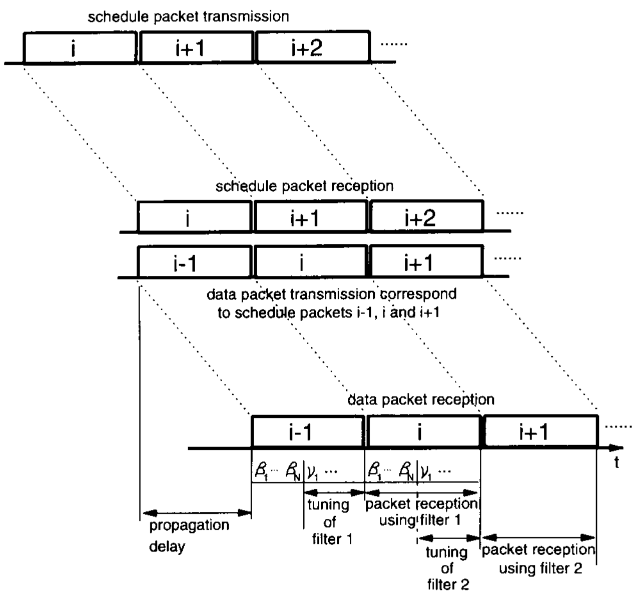

schedule packet transmission
i
i+1
i+2
schedule packet reception
i
i+1
i+2
i-1
i
i+1
data packet transmission correspond to schedule packets i-1, i and i+1
data packet reception
i-1
i
i+1
t
$\beta_1 \cdots \beta_N | \gamma_1 \cdots$
$\beta_1 \cdots \beta_N | \gamma_1 \cdots$
propagation delay
tuning of filter 1
packet reception using filter 1
tuning of filter 2
packet reception using filter 2

Fig. 3 Pipelining operation of the protocol.

$$J_i=\begin{cases}(0,0) & \text{null packet, or no transmission}\\(1,0) & \text{a sequenced packet}\\(0,1) & \text{a non-sequenced packet}\\(1,1) & \text{a relocated packet}\end{cases}$$

• γ_i is the destination address of the relocated packet from station i and is undefined otherwise.

Upon receiving a schedule packet, all stations transmit and/or receive packets in their data channels accordingly. The transmission and reception on the in-bound, the out-bound and the data channels as well as the processing of backlog information are all done slot by slot in a *pipelining* operation (Fig. 3).

3. 2 Procedures at the Scheduler

The scheduler maintains two *backlog matrices* $\boldsymbol{B}=[b_{ij}]_{N\times N}$ and $\boldsymbol{C}=[c_{ij}]_{N\times N}$, where b_{ij} and c_{ij} are the current number of sequenced and non-sequenced packets respectively at station i destining to station j. The scheduler performs three tasks in each slot simultaneously.

1. Updating backlog matrices and formulating a transmission schedule
   Remark: Matrices $\boldsymbol{B}$ and $\boldsymbol{C}$ are updated in every time slot by adding the number of newly arrived packets to and subtracting the number of packets scheduled for transmission from the corresponding elements in the two matrices. They are then used as inputs to the scheduling algorithm (to be described in Sect. 4) to formulate a schedule packet.
2. Receiving minipackets
   Remark: The scheduler receives minipackets in its in-bound channel. These minipackets will be used to update the backlog matrices in the next slot.
3. Broadcasting a schedule packet
   Remark: The scheduler broadcasts a schedule packet formulated in last slot to all stations through its out-bound channel.

3. 3 Procedures at the Stations

Each station performs the following three tasks simultaneously in every slot.

1. Transmitting a minipacket
   Remark: Each station formulates a minipacket and transmit it in the assigned minislot of the in-bound channel.
2. Receiving a schedule packet
   Remark: Each station receives a schedule packet from the scheduler to learn whether there is a packet transmission and/or reception scheduled for it. A station learns about the packet destinations for the next slot from the first part of the received scheduled packet (i.e. the β_i's). It immediately tunes one of its idle tunable filter to the assigned wavelength channel for data reception. (The other tunable filter is receiving a data packet as shown in Fig. 3.) This kind of push-pull arrangement avoids the filter tuning time overhead of a single tunable filter design.
3. Transmitting and/or receiving data packets
   Remark: After receiving a schedule packet, a local station, say station i, transmits and/or receives packets accordingly as follows:
   - $J_i=(1,0)$: Transmitting a sequenced packet to station β_i;
   - $J_i=(0,1)$: Transmitting a non-sequenced packet to station β_i;
   - $J_i=(1,1)$: transmitting a relocated packet with destination γ_i to a relaying station β_i.
   - Upon matching a β_k with the local address, the local station tunes a filter to wavelength λ_k to receive a packet after a fixed interval of one station-to-station propagation delay. If $J_k=(1,1)$, put the received packet into the output queue.

In this protocol destination conflicts are avoided in every slot with the use of conflict-free transmission schedules. Packet queue lengths in individual stations are equalized by relocating some non-sequenced packets from congested stations that do not have any packet transmission scheduled to uncongested stations that do not have any packet reception scheduled. This load balancing operation allows the sharing of buffers among all stations in the network. As we will show in Sect. 4, such sharing can significantly reduce the number of buffers required.

4. Computation of Transmission Schedules

As mentioned before, the formulation of a transmission schedule includes the identification of a maximum set of conflict-free transmission-reception pairs and the identification of the remaining transmission-reception pairs for load balancing. In this section, we design an algorithm for each of the above objectives. These algorithms are to be executed in the scheduler.

4. 1 Throughput Maximizing Algorithm

Given $\boldsymbol{B}$ and $\boldsymbol{C}$, the scheduler can compute a maximum set of stations for conflict-free transmissions by using the System Distinct Representative (SDR) algorithm [13]. This optimal algorithm, however, requires too much computation for our purpose. The Maximum Remaining Sum (MRS) algorithm [5] which gives optimal solution most of the time but with complexity $O(N)$ is chosen for use in our application. A brief description of the algorithm is given in the Appendix. The output of the MRS algorithm is a binary transmission matrix $\boldsymbol{T}=[t_{ij}]$ where $t_{ij}=1$ indicates station i should transmit a packet to station j. $\{\beta_i\}$, $\{\gamma_i\}$ and $\{J_i\}$ are obtained as follows.

```
begin
    for $i, j$=1 to $N$ do
    begin
        if( $t_{ij}$=0 ) then $J_i : = (0,0)$;
        else
        begin
            if( $b_{ij}$>0 ) then $\beta_i : = j$; $J_i = (1,0)$;
            {station $i$ transmits a sequenced packet to station $j$}
            if( $b_{ij}$=0 and $c_{ij} > 0$) then $\beta_i : = j$; $J_i = (0,1)$;
            {station $i$ transmits a non-sequenced packet to station $j$}
        end;
    end;
end.
```

The above algorithm always gives a higher transmission priority to sequenced packets.

4. 2 Buffer Sharing Algorithm

Among all the stations with $J_i = (0,0)$ in the output of the Throughput Maximizing algorithm, the buffer sharing algorithm would check if some of the non-sequenced packets can be routed from congested stations to uncongested stations.

In designing the buffer sharing algorithm, we have two objectives: 1) to balance queue lengths among stations by routing packets from congested stations to uncongested stations, and 2) to reduce the number of zero elements in matrix $\boldsymbol{C}$ without sacrificing the first objective. The former aims at minimizing packet loss due to buffer overflow, and the latter aims at increasing the network throughput as more non-zero elements in $\boldsymbol{C}$ can usually give a more efficient transmission schedule. Define a queue size array as $\boldsymbol{q}=[q_1, q_2, \cdots, q_N]$ where $q_i=\Sigma_j(b_{ij}+c_{ij})$, $i=1, 2, \cdots, N$, and let $\boldsymbol{q}$ be updated with $\boldsymbol{B}$ and $\boldsymbol{C}$. Let I_T be the set of stations with idling transmitters and I_R be the set of stations with idling receivers. Let $r=|I_T|$ be the number of elements in I_T. Obviously $|I_R|=r$. The following is the buffer sharing algorithm:

```
begin
    "sort the queue lengths of the stations in $I_T$ in descending order, and call these stations $n_1$, $n_2, \cdots, n_r$;"
    "sort the queue lengths of the stations in $I_R$ in ascending order, and call these stations $m_1$, $m_2, \cdots, m_r$;"
    for $i$=1 to $r$ do
    begin
        if ( $n_i \neq m_i$ and $q_{n_i} > q_{m_i}$ ) then
        begin
            "find $e$ such that $c_{n_i e} \neq 0$ and $c_{m_i e}$ is a minimum;"
            $\beta_{n_i} := m_i$; $\gamma_{n_i} := e$; $J_{n_i} = (1,1)$;
            {station $n_i$ should transmit a non-sequenced packet with destination $e$ to a relaying station at address $m_i$}
        end;
    end;
end.
```

This algorithm successively assigns a station with the longest queue to transmit a non-sequenced packet to a relaying station having the shortest queue.

As an example, let

$$\boldsymbol{B}=\begin{bmatrix}0&0&1&0\\0&0&0&2\\0&0&0&0\\0&0&0&0\end{bmatrix},\quad \boldsymbol{C}=\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&2&0&0\\0&0&1&0\end{bmatrix},$$

$$\boldsymbol{B}+\boldsymbol{C}=\begin{bmatrix}0&0&1&0\\0&0&0&2\\0&2&0&0\\0&0&1&0\end{bmatrix}.$$

Then the schedule packet is computed to be

$$[\beta_1\cdots\beta_N, J_1\cdots J_N, \gamma_1\cdots\gamma_N]$$
$$=[3, 4, 2, 1, (1,0), (1,0), (0,1), (1,1), -, -, -, 3].$$

From the schedule packet, it is seen that stations 1 and 2 are each assigned to transmit a sequenced packet to stations 3 and 4 respectively. Station 3 is assigned to transmit a non-sequenced packet to station 2 and station 4 is assigned to transmit a non-sequenced packet with destination 3 to station 1 for relaying.

5. Computational Complexity of a Transmission Schedule

To form a transmission schedule involves the updating of $\boldsymbol{B}$ and $\boldsymbol{C}$, the execution of the throughput maximizing and the buffer sharing algorithms. For an easier operation, let the scheduler maintains matrices $\boldsymbol{B}$ and $\boldsymbol{B}+\boldsymbol{C}$ instead of $\boldsymbol{B}$ and $\boldsymbol{C}$. In addition, let the scheduler keep a binary array $\boldsymbol{u}=[u_1, u_2, \cdots u_N]$, where $u_i=1$ indicates that station i is assigned to receive a packet and $u_i=0$ otherwise. Let the complexity of algorithm X be denoted as C_X.

5. 1 The Complexity for Updating Backlog Matrices, C_B

Adding new arrivals to $\boldsymbol{B}$ and $\boldsymbol{B}+\boldsymbol{C}$ needs at most $2N$ scalar additions and $2N$ scalar assignments. Subtracting the transmitted packets (including those to relaying stations) from $\boldsymbol{B}$ and $\boldsymbol{B}+\boldsymbol{C}$ requires at most N scalar comparisons, N scalar additions, and N scalar assignments. In addition, the updating of $\boldsymbol{q}$ with that of $\boldsymbol{B}+\boldsymbol{C}$ requires at most N additions and N assignments. Therefore

$$C_B \leqq [9N]_{\text{SCALAR OPERATIONS}}.$$

5. 2 The Complexity of the Throughput Maximizing Algorithm, C_{TM}

The complexity of the MRS algorithm as given in [5] is

$$C_{MRS} \leqq [10N]_{\text{VECTOR OPERATIONS}} + [3N]_{\text{SCALAR OPERATIONS}}.$$

β_i, γ_i and J_i can be obtained directly by changing step 5 of the MRS algorithm to

```
$\beta_p := q$;   $u_{\beta_p} := 1$;
    **if** $(b_{pq} > 0)$ **then**
        $J_p = (1, 0)$
    **else**
        $J_p = (0, 1)$;
```

The resulting complexity is

$$C_{TM} \leqq [10N]_{\text{VECTOR OPERATIONS}} + [6N]_{\text{SCALAR OPERATIONS}}.$$

5. 3 The complexity of the Buffer Sharing Algorithm, C_{BS}

Operations in the buffer sharing algorithm include at most the followings:

- N scalar comparisons (with $\beta_1, \cdots, \beta_N$) and N scalar assignments in finding idling transmitters.
- Sorting q_i needs $(7r/6)\log(r)$ comparisons and exchange operations by the *Quicksort* algorithm [14].
- Finding idling receivers needs N scalar comparisons (with $u_1, \cdots, u_N$) and N scalar assignments. Sorting the queue lengths needs another $(7r/6)\log(r)$ scalar operations.
- To find destination address e, we first compute a set of N integers as follows:

  $$\{(b_{n_t1})^+(b_{m_t1}+1),\ (b_{n_t2})^+(b_{m_t2}+1), \cdots, (b_{n_tN})^+(b_{m_tN}+1)\},$$

  where $(x)^+$ is defined as $(x)^+=1$ for $x \geqq 1$ and $(x)^+=0$ otherwise for any integer x. This can be done in 3 vector operations. Choosing a smallest non-zero integer among them requires another vector operation [5]. Therefore we have a total of 4 vector operations in this step.

Set r equal to its maximum value $N-1$ and add up the above, we have

$$C_{BS} < \left[\frac{7N}{3}\log N + 4N\right]_{\text{SCALAR OPERATIONS}} + [4N]_{\text{VECTOR OPERATIONS}}$$

Therefore the total computational complexity of a transmission schedule is bounded by:

$$C_{\text{TOTAL}} < \left[\frac{7N}{3}\log N + 19N\right]_{\text{SCALAR OPERATIONS}} + [14N]_{\text{VECTOR OPERATIONS}}$$

For simplicity, define the right hand side of the above inequality as $f(N)$.

6. Performance Analysis by Simulation

In this section we study by computer simulation the savings of buffers at each station with the use of buffer sharing. Let packet arrival to each station be a Bernoulli process with rate p per time slot. Let f be the probability that a packet is of the non-sequenced type. For the following examples we let the requirement on packet blocking probability P_B be $P_B \leqq P_B^*$, $N=12$ and the station-to-station propagation delay through the star coupler be one slot. In case the time needed for control signalling or schedule computation is larger than the packet transmission time T_{pkt}, the slot size has to be made equal to $\max[f(N), N\tau]$, where $f(N)$ is the scheduling time upper bound and $N\tau$ is the transmission time of N minipackets on the control channel. As only T_{pkt} is used for packet transmission, the channel efficiency is reduced to $T_{pkt}/\max[f(N), N\tau]$. As both $f(N)$ and $N\tau$ increase with N, it is important to set the packet size T_{pkt} as large as $\max[f(N), N\tau]$ for the proposed buffer sharing scheme to work well. Alternatively, a limit on the network size N can also ensure that $\max[f(N), N\tau]$ will not be larger than T_{pkt}. In WDMA networks with a single optical star coupler, N cannot be made very large because of the constraints on power budget and receiver sensitivity. As an example illustrating the processing time requirement, assume a packet size of 2000 bits and a data rate of 45 Mbit/s. Then the slot size is 44 μs. For a typical value of $N=32$[15], $f(N)=1168$ operations. Assuming a processor speed of 50 Mflop, the computation time is merely 23 μs, well within the duration of a packet transmission time. For larger values of N, connections to multiple couplers are required [11]. The study of buffer sharing under this environment, however, is beyond the scope of this paper.

6. 1 Uniform Traffic

Let all stations have the same traffic rate. Figure 4 shows the required buffer size (in number of packets stored) in each station versus f, given $P_B^*=10^{-5}$ and $p=0.9$. Without buffer sharing, 59 buffers are needed to satisfy the blocking performance requirement. With the use of buffer sharing, it is seen that more than one-third reduction of the buffer size can be achieved for f as small as 0.2. For $f=0.5$ the total buffer size can be further reduced to 17 which is only 29% (17/59) of the original buffer requirement.

Figure 5 compares the mean end-to-end packet delay with and without buffer sharing. When p is small the mean packet delay is 7 slots which can be accounted to as 1 slot for backlog reporting, 1 slot for computing schedule, 1 slot for broadcasting schedule, 1 slot for packet transmission and 3 slots for the three times station-to-station propagation delay. When p is large, the packet delay is dominated by queueing at input stations. By equalizing queue size through buffer sharing there is a higher chance of obtaining a full transmission schedule. This fact is reflected in a significant reduction of mean end-to-end packet delay at large values of p. This delay difference diminishes when the packet transmission time is small compared to the network propagation delay.

6. 2 Non-uniform Traffic

Here, we divide stations into three groups of 4 stations each. Let each of the group 1 stations have arrival rate p, each of group 2 stations have arrival rate $p+\delta$ and each of the group 3 stations have arrival rate $p+2\delta$. Figure 6 shows the buffer size needed in each station given $P_B^*=10^{-5}$, $p=0.85$ and $\delta=0.05$. As before, for $f=0.2$, the buffer size can be reduced from 58 to 31 using buffer sharing whereas for $f=0.5$ it can be further reduced to 15, or only 26% of the original. This

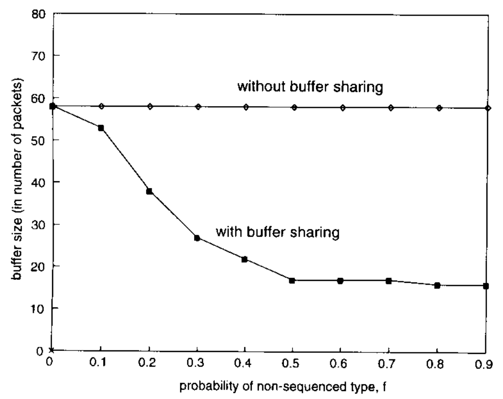


Fig. 4 Buffer size comparison, uniform traffic, $p=0.9$, $P_B^*=10^{-5}$.

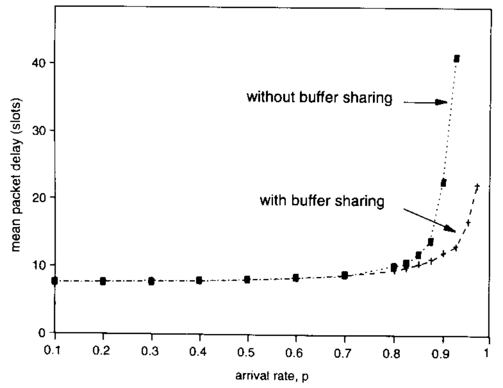


Fig. 5 Delay comparison, uniform traffic, $f=0.5$.

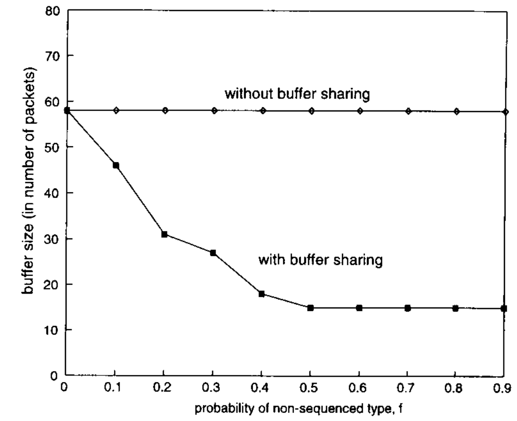


Fig. 6 Buffer size comparison, non-uniform arrival rates, $p=0.85$, $\delta=0.05$, $P_B^*=10^{-5}$.

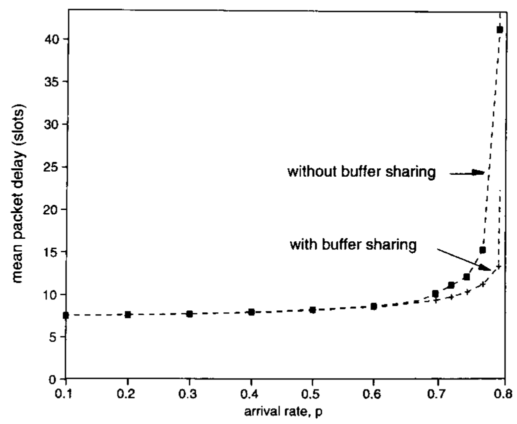


Fig. 7 Delay comparison, non-uniform arrival rates, $\delta=0.1$, $f=0.5$.

higher reduction rate of the buffer size shows that the buffer sharing algorithm proposed in this paper is more efficient in non-uniform traffic environment.

Figure 7 shows the mean end-to-end packet delay versus arrival rate p for $\delta=0.1$. Here, the network can accommodate a maximum traffic load of $p=0.8$ because $p+2\delta$ must not be larger than 1. Significant reduction of delay for large p is also observed when buffer sharing is employed.

7. Conclusions

In this paper we demonstrate that buffer sharing among stations is possible in an optical WDMA network. Such sharing of buffers amounts to a drastic reduction of buffer requirement while reducing the mean packet delay. The implementation of a conflict-free WDMA network with buffer sharing is currently under way. The challenge appears on the design of the scheduler. Generalization of the problems addressed here to variable size packets appears to be difficult.

Acknowledgement

This research is supported by a generous grant from KDD Hong Kong Ltd. The many helpful suggestions given by a reviewer is highly appreciated.

References

[1] Kobrinski, H. and Cheung, K. W., "Wavelength-tunable optical filters: applications and technologies," *IEEE Mag.*, vol. 27, no. 10, pp. 53-63, Oct. 1989.

[2] Henry, P. S., "High-capacity lightwave local area networks," *IEEE Mag.*, vol. 27, no. 10, pp. 20-26, Oct. 1989.

[3] Ganz, A. and Chlamtac, I.,"Path allocation access control in fiber optical communication systems," *IEEE Trans. Comput*, vol. 38, no. 10, pp. 1372-1382, Oct. 1989.

[4] Chen, M. S., Dono, N. and Ramaswami, R., "A media-access protocol for packet-switched wavelength division multiaccess metropolitan area networks," *IEEE JSAC*, vol. 8, no. 6, pp. 1048-1057, Aug. 1990.

[5] Chen, M. and Yum, T. S., "A conflict-free protocol for optical WDMA networks," *Proceedings of IEEE Globecom'91*, pp. 35.6.1-35.6.6.

[6] Habbab, I. M. I., Kavehrad, M. and Sundberg, C-E. W. "Protocols for very high speed optical fiber local area networks using a passive star topology," *IEEE J. Lightwave Tech.*, vol. LT-5, no. 12, pp. 1782-1794, Dec. 1987.

[7] Sudhakar, G. N. M., Georganas, N. D. and Kavehrad, M., "Slotted Aloha and reservation Aloha protocols for very high-speed optical fiber local area networks using passive star topology," *IEEE/OSA J. Lighwave Tech.*, vol. 9, no. 10, pp. 1141-1422, Oct. 1991.

[8] Chipalkatti, R., Zhang, Z. and Acampora, A. S., "High speed communication protocols for optical star coupler using WDM," *IEEE Infocom'92*, pp. 2124-2133.

[9] Cheung, N. K., Nosu, K. and Winzer G., (ed.), "Dense wavelength division multiplexing techniques for high capacity and multiple access communication systems," *IEEE JSAC*, vol. 8, no. 6, Aug., 1990.

[10] Mukherjee, B., "WDM-based local lightwave networks Part I: single-hop systems," *IEEE Network Mag.*, May 1992.

[11] Ramaswami, R., "Multiwavelength lightwave networks for computer communication," *IEEE Communications Mag.*, vol. 31, no. 2, pp. 78-88, Feb. 1993.

[12] Rubin I. and Baker, J. E., "Media access control for high-speed local area and metropolitan area communications networks," *Proc. of The IEEE*, vol. 78, no. 1, pp. 168-203, Jan. 1990.

[13] Hall, Jr, M., *Combinatorial theory*, Second Edition, John Wiley & Son, 1986.

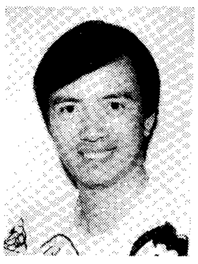
[14] Schildt, H., *C: The Complete Reference, second edition*, Osborne McGraw-Hill, 1990.

[15] Green, Jr, P. E., *Fiber Optic Networks*, p. 73. Prentice-Hall, 1993.

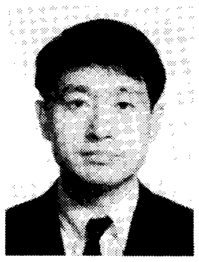
Appendix

Let $\boldsymbol{D}=[d_{ij}]$ be a binary matrix of which $d_{ij}=(b_{ij}+c_{ij})^+$ and where $x^+=1$ if $x>0$ and $x^+=0$ otherwise. Let n_i and m_i be the ith row sum and column sum of $\boldsymbol{D}$ respectively. The MRS algorithm given below would produce a transmission matrix $\boldsymbol{T}=[t_{ij}]$.

```
1. Find $[n_1, n_2, \cdots, n_N]$ and $[m_1, m_2, \cdots, m_N]$ from $\boldsymbol{D}$;
2. Find a smallest element in $[n_1, n_2, \cdots, n_N]$ and a smallest element in $[m_1, m_2, \cdots, n_N]$. If the smaller of the two elements is a row sum, denote its position as $p$ and goto 3; else denote the position of the column sum as $q$ and goto 4;
3. Find all $k$ such that $d_{pk} \neq 0$ and denote them as $k_1, k_2, \cdots$. Find $q$ such that $m_q \leqq m_i$ for all $i \in \{k_1, k_2, \cdots\}$ goto 5;
4. Find all $k$ such that $d_{kq} \neq 0$ and denote them as $k_1, k_2, \cdots$. Find $p$ such that $n_p \leqq n_i$ for all $i \in \{k_1, k_2, \cdots\}$;
5. $t_{pq} := 1$;
   $[n_1, n_2, \cdots, n_N] := [n_1, n_2, \cdots, n_N] - [d_{1q}, d_{2q}, \cdots, d_{Nq}]$;
   $[m_1, m_2, \cdots, m_N] := [m_1, m_2, \cdots, m_N] - [d_{p1}, d_{p2}, \cdots, d_{pN}]$;
   $[d_{1q}, d_{2q}, \cdots, d_{Nq}] := 0$;
   $[d_{p1}, d_{p2}, \cdots, d_{pN}] := 0$;
   $n_p := 0$; $m_q := 0$;
6. If $n_i = 0$ for all $i$, STOP.
   else goto 2.
```



Ming Chen received the B.Eng. and M.Eng. degrees in telecommunications engineering from Beijing University of Posts and Telecommunications. He is currently finishing the Ph.D. in electrical engineering at the University of Ottawa, Canada. Prior to his study in Canada, he worked for Singapore Telecom, Singapore, conducting projects of a Teleview gateway connecting ISDN and DECnet networks, and an ATM trial network in Singapore. Frtm 1989 to 1991, he was a research fellow in information engineering at The Chinese University of Hong Kong, Hong Kong, where he was a recipient of the KDD Research Fellowship and designed protocols for WDMA and video delivery networks. After his undergraduate studies, he was with Sichuan Research Institute of Posts and Telecommunications, China, as a network engineer. Mt. Chen's research interests include design and performance evaluation of protocols for high-speed networks, internetworking, multimedia synchronization, and ATM networks.



Tak-Shing Peter Yum received his BSc, MSc, MPh and PhD from Columbia University in 1974, 75, 77 and 78 respectively. He worked in Bell Telephone Laboratories, USA for two and a half years and taught in National Chiao Tung University, Taiwan, for 2 years before joining The Chinese University of Hong Kong in 1982. He has published original research on packet switched networks with contributions in routing algorithms, buffer management, deadlock detection algorithms, message resequencing analysis and multiaccess protocols. In recent years, he branched out to work on the design and analysis of cellular network, lightwave networks and video distribution networks. He believes that the next challenge is designing an intelligent network that can accommodate the needs of individual customers. Peter claims himself to be an existentialist. He enjoys playing bridge and badminton.