Real-Time Data Allocation Scheme Based on Dynamic Replacement in Burst Photonic Networks

Nattapong Kitsuwan, Eiji Oki, Naoto Kishi, and Tetsuya Miki
Department of Information and Communication Engineering
University of Electro-Communications
Tokyo, Japan
{kitsuwan,oki,kishi}@ice.uec.ac.jp, miki@office.uec.ac.jp

Abstract—This paper proposes a real-time allocation scheme for photonic networks that use wavelength division multiplexing (WDM) and optical time division multiplexing (OTDM) technologies in the backbone and ring regional networks, respectively. A frame that is used for transferring data and control information in a ring regional network takes only one round to complete data allocation processing, instead of two rounds as required by the prior reservation scheme, so that data can be transmitted immediately. Our challenge is to provide max-min fair share in terms of throughput with just one round. If no free space is left on the frame, the proposed scheme allows a group (some) of the newly requested data to replace some of the already allocated data to provide max-min fair share, in terms of throughput. Data replacement and de-fragmentation are processed in the optical domain. Simulations show that the proposed scheme maintains max-min fair share even in unbalanced traffic scenarios. The complexity of de-fragmentation depends on the number of delay lines needed to regenerate the original data groups. The maximum number of delay lines is determined.

I. INTRODUCTION

All-optical networks are a promising technology for near future implementation. Various services are now being offered via the Internet such as high-speed data transmission and real-time video conferencing. As a result, traffic amounts are rapidly increasing. A high-speed network is needed to deal with this. Using an optical-electronic-optical converter (O/E/O) for data payload transfer is not effective because of the electrical processing bottleneck.

Optical packet switching has the problem of data buffering at intermediate nodes when large amounts of data have to be transmitted [1]. It is difficult to create an optical buffer that can store large amounts of data at intermediate nodes on the backbone network. Therefore, we have to avoid using buffers in the backbone network. Many wavelengths have to be used since unique wavelengths are assigned to each pair of source and destination nodes [2]. One network architecture that can reduce the number of wavelengths is the burst photonic network; it combines optical time division multiplexing (OTDM) in the regional networks (RNs) with wavelength division multiplexing (WDM) in the backbone network. Called the OTDM-WDM-OTDM (TWT) burst photonic network, it was proposed by [3]. The number of wavelengths equals the number of RN pairs.

The TWT burst photonic network offers large bandwidth. It is among the more feasible structures in the near future since no connection establishment in the backbone network is required because the wavelength path of each network is fixed. There are two main parts to the network: the RN and the backbone network. Each RN links to the nodes within a regional area, such as a metropolitan area, using OTDM in a ring topology. A frame is used for transferring data and control information in this ring. Each frame, which consists of a data payload and a header, multiplexes data from each node. The number of timeslots that each node can use is calculated as its timeslot quota. The backbone network connects each RN pair by one or more unique wavelengths via WDM technology to send frames from one RN to another RN. Data allocation is managed only by the RN ring networks. Since the data is transmitted using OTDM, WDM, and OTDM networks in order, we call this structure the TWT burst photonic network.

The TWT burst photonic network has two main data allocation modes: reservation mode and real-time mode, see [3], [4]. In the reservation mode, a regional network node (RNN) uses a frame to collect the bandwidth requests of all edge nodes (ENs) in the first round, which it uses to determine the bandwidth allocations. Max-min fair share in terms of throughput [5] is achieved, since the RNN knows all bandwidth requests from all ENs. User data is sent to the destination RN via the second round frame. In the real-time mode, on the other hand, bandwidth allocation and data collection are performed in just one round. That is, data can be transmitted immediately in the first round. However, max-min fair share, in terms of throughput, is not assured. This is because none of the EN bandwidth requests are centrally processed. Our challenge is to provide max-min fair share in terms of throughput, which is an advantage of the reservation mode, with only one round, i.e. the real-time mode.

This paper proposes a real-time data allocation scheme, called the dynamic replacement (DR) scheme, for the TWT burst photonic network. Payload data can be replaced by data from later ENs on the ring, if no free space is left on the payload. The replaced data is not lost because it is stored in the local buffer of the source EN. This is likely to fragment the EN data. As it is impractical to demand that each EN receive fragmented data in the optical domain, data de-fragmentation is performed at the RNN in the optical domain by changing the delay time of each data fragment before releasing it from the source RNN to the backbone network. Simulations show
that the DR scheme offers max-min fair share for unbalanced traffic scenarios, while allowing data to be sent immediately in the first round.

Each frame consists of data fragments arranged in a specific pattern. There are two approaches to de-fragmenting the data: numbered ordering and appearance ordering. The former orders fragments by source EN number. The latter orders them by the position within the received frame. Since de-fragmentation changes the delay time of each fragment, it increases overall delay. This delay is examined. Note that we do not consider queuing delay in each EN buffer.

The rest of this paper is organized as follow: Section II describes the TWT burst photonic network architecture. Section III defines the dynamic replacement scheme and describes how it works. Section IV shows the simulations and their results. Section V calculates the delay due to the de-fragmentation process. Finally, Section VI summarizes the key points.

II. TWT Burst Photonic Network

The TWT burst photonic network consists of two main networks; RNs that use OTDM technology and long haul backbone network (LHBN) that use WDM technology as illustrated in Fig. 1. The function of each part is described below:

A. Long Haul Backbone Network (LHBN)

LHBN is a core network that connects RN pairs. A data frame is transmitted from one RN to another RN using a unique wavelength. For example as shown in Fig. 1, there are $\lambda_{ij}$ links between $i$-RN and $j$-RN.

B. Regional Network (RN)

Client users, such as home users or company users, are linked to an EN. The ENs are connected to a local RNN in a ring network. The RNNs are connected by the LHBN. Frames are circulated along the paths that correspond to each of the wavelength paths to an RNN to assign bandwidth and to multiplex the data from each EN using an allocation scheme that is described in the next section. There are two main allocation schemes in TWT photonic networks:

1) Reservation mode: the bandwidth of each EN within the same RN is reserved in advance before transferring the data. The frame header in this mode consists of a bandwidth request field and timeslot addresses. Its procedure is explained below and in Fig. 2.

- Step 1: A frame that consists of a header and a data payload area is generated by the RNN for each wavelength path to a remote RNN. The frame is transmitted at the same frequency as will be used to connect to the remote RNN.
- Step 2: At each EN, the frame is received, and the EN’s bandwidth request, if any data is waiting to be sent to an EN in the remote RN indicated by the frame’s frequency, is entered into the header field. The entered data consists of source EN, destination EN, and data amount. The frame is then reinjected into the RN.
- Step 3: After the frame reaches the RNN, the bandwidth requests of all ENs are read and the bandwidth resources (timeslots in the frame) are assigned by the RNN to each EN.
- Step 4: The frame is re-circulated along the same path.
- Step 5: At each EN, timeslot addresses allocated to the EN are read from the header.
- Step 6: The data held by the EN is transferred to the timeslots assigned to the EN.
- Step 7: After the frame arrives back at the local RNN, it is transmitted by a unique wavelength to the destination RNN.

This mode, therefore, circulates the frame twice, once to acquire bandwidth requests and the second time to acquire the data. The frame is then transmitted to the destination RNN. Max-min fair share bandwidth assignment is achieved because the local RNN collects the bandwidth requests of all local ENs but data transfer is delayed by the two-round process.

2) Real-time mode: data can be immediately written into the payload when the frame arrives at an EN since timeslot assignment is performed at each EN. The frame is transmitted to the destination RNN after finishing its first circuit. The bandwidth allocation method of this mode is shown below
and in Fig. 3.

- Step 1: This step is the same as Step 1 in reservation mode.
- Step 2: At each EN, the available timeslot addresses are read from the header. Sufficient unused timeslot addresses are then calculated.
- Step 3: Revised timeslot address information is written back into the header.
- Step 4: The EN transfers as much of the data held by it as possible to the frame.
- Step 5: After the frame arrives at the local RNN, it is transmitted to the remote RNN.

Although fairness in term of throughput can not be guaranteed by the real-time mode described above, since the bandwidth requests of all ENs are unknown, the allocation process is finished within just one round.

III. DYNAMIC REPLACEMENT SCHEME

The DR scheme achieves max-min fair share within just one round. If the data payload is full, a data request from an EN can trigger replacement of some of the data already in the payload. Note that the dropped data is not lost since the originating EN keeps a copy of the data. Allocation proceeds as follows:

- Data that arrives at an EN is stored in a local FIFO buffer.
- The EN receives a frame that consists of a data payload and a header. The payload contains the data entered by upstream ENs. The header includes two pieces of information:
  - Amount of data successfully transferred from each EN in the previous round (entered by local RNN).
  - Amount of data entered by each of the upstream ENs.
- Data in the local buffer that was successfully sent in the previous round is deleted.
- If the unused space in the payload is larger than the whole data in the local buffer, the whole data is duplicated in the payload. Note that data duplication is employed, instead of transfer, because it is possible that some of the data will be replaced by data from downstream ENs.
- If no space is available or the free space is less than the whole data in the local buffer,
data de-fragmentation is shown in Fig. 5. Since A1 and A2 are separated by B, we need to pass the fragments across tunable delay lines to change their relative delay time. In this case, A1 is delayed by T seconds, and B by 2T seconds, while A2 is passed straight through. As a result, the data from the same EN is formed into a contiguous block.

To clarify the DR scheme, an example is illustrated in Fig. 6. Five ENs are assumed for simplicity. Each frame contains 1,000 timeslots.

EN1 requests 400 timeslots. This number is less than the unused capacity (1,000 timeslots) so all EN1 timeslots are duplicated in the payload. The free space is now 600 timeslots.

At EN2, 500 timeslots are requested. This is also less than the unused capacity, 600, so all are duplicated in the payload. The remaining free space is 100 timeslots.

At EN3, 300 timeslots are requested. Unfortunately, only 100 timeslots are available. First, the timeslot quota of each EN from EN1 to EN3 is calculated based on max-min fair share. The resulting quotas are 350, 350 and 300 timeslots for EN1, EN2, and EN3, respectively. Next, the data from EN3 replaces 50 EN1 timeslots and 150 EN2 timeslots. Each EN now has the quota calculated by EN3. The data of EN1 and EN2 that were replaced by EN3 data are not lost because they are still stored in the local buffer of the originating EN.

EN4 requests 300 timeslots but no free space is available. Again quotas are determined so as to realize max-min fair share; each EN gets 250 timeslots as its quota. 100 EN1 timeslots, 100 EN2 timeslots, and 50 EN3 timeslots are replaced by EN4 data. The remaining data of EN4 is stored in a buffer at EN4.

EN5 requests 400 timeslots. The quota of each EN is 200 timeslots according to the max-min fair share theory. EN5 data is placed into 50 timeslots of each EN area in the payload.

IV. SIMULATION AND RESULTS

Simulations were conducted to ensure that the DR scheme provides max-min fair share under unbalanced traffic scenarios. A single wavelength was assumed to offer 150 Gbps transmission speed. The frame length was 500µs. Hence, a frame can contain approximately 9.3 MB. We also assume 2,000 timeslots per frame. The number of bytes per slot is calculated as $500\times150\text{Gbps}/8\text{bit}/2,000\text{slots} = 4,687$ bytes. Based on measured Internet traffic patterns, incoming traffic follows the self-similar traffic pattern [6], [7].

A. Sample scenarios

Three sample scenarios were tested. Fig. 7 compare the throughput rates at each EN for our proposal and optimal throughput based on the max-min fair share theory.

In Fig. 7(a), the incoming traffic at EN4 to EN10 overload the link, while that at EN1 to EN3 do not. Optimal throughput is analytically calculated by considering the max-min fair share theory. EN1, EN2, and EN3 obtain full throughput, while the other ENs have the same restricted throughput. Throughputs set by the DR scheme at all ENs mostly equal the optimal throughputs. In Fig. 7(b), we set the opposite incoming traffic scenario. The incoming traffic at EN1 to EN7 overload the link, while that at EN8 to EN10 do not. Fig. 7(b) shows that, here again, the throughputs achieved by the DR scheme at all ENs mostly equal the optimal throughputs. In Fig. 7(c), the incoming traffic at EN1 to EN4 and EN7 to EN10 overload the loadlink, while that at EN5 and EN6 do not. The ENs requesting excessive traffic receive fair, but limited, throughputs. Fig. 7(c) shows that the throughputs obtained by the DR scheme at all ENs mostly equal the optimal throughputs. The results of all scenarios indicate that the DR scheme provides max-min fair share.

B. Random scenarios

One hundred scenarios with randomly generated incoming traffic patterns were also examined. The throughputs obtained with the DR scheme were also similar to the optimal throughput, as in the case of the sample scenarios. To see whether the max-min fair share is achieved, the fairness index introduced in [8], [9] is used.

We confirmed via a simulation that the fairness index values in all random scenarios were 1.0 with a variance of just 1%. This indicates that the DR scheme does achieve fairness in terms of throughput.

V. DE-FRAGMENTATION DELAY ANALYSIS

As mentioned in Section III and the example in Fig. 5, data from a node might become fragmented on a frame. It is difficult for the destination to demultiplex the fragmented data in the frame. Moreover, every EN must support this rather complex and expensive process. Our solution is to de-fragment the data on the frame at the original RNN before transmitting the frame. The de-fragmentation can yield numbered ordering or appearance ordering. De-fragmentation is performed by using delay lines to impose the appropriate delays on the
fragments. The complexity of the de-fragmentation process depends on the number of delay lines. In this section, the maximum number of delay lines is determined to evaluate hardware costs.

Fig. 8 shows a block diagram of de-fragmentation. It consists of switches at RNN input and output ports and a group of delay lines. The data fragments are separated by the input switch. Each fragment is then passed to the appropriate delay line. The outputs of the delay lines are re-assembled at the output switch before the frame is transmitted by the RNN. The data in the released frame is de-fragmented. We note that each delay line can handle several fragments in turn if the fragments are sufficiently offset in the received frame. The delay lines have delay values of \( T, 2T, \ldots, D_{\text{max}}T \) where \( T \) is timeslot length and \( D_{\text{max}}T \) is the maximum delay. \( D_{\text{max}}T \) is the parameter representing hardware complexity.

To determine \( D_{\text{max}}T \): first, the required delay of each fragment has to be determined. Next, the required number of delay lines is estimated. To obtain the delay of each fragment, the position of the fragment before and after de-fragmentation is analyzed, see the next subsection. The distance between the position before and after de-fragmentation determines the amount of delay needed. Three position changes are possible. First, the fragment bypasses the delay lines. Second, the fragment has to be delayed against the others. Third, the fragment has to be advanced against the others.

This section details this process. First, the orders of the fragments before and after de-fragmentation are described. Next, the temporal position of each fragment is obtained. Finally, the maximum delay, which is the total number of delay lines required at the RNN, is determined by linear programming.

### A. Data before De-fragmentation

The terminologies used in this section are defined in the following:
- \( F \) : Number of timeslots in a frame.
- \( n \) : Number of ENs.
- \( d_{m,k} \) : Timeslot index number of the \( k \)th data fragment from \( m \)th EN.

Since data may be replaced to satisfy max-min fair share at every EN, the data fragments are likely to become separated. Note that since data replacement is always performed at the end of the fragment, fragments from ENs later in the RN are likely to appear ahead of fragments from ENs earlier in the RN. We assume that each EN has incoming traffic to simplify the explanation. The case that one EN has no incoming traffic is easily covered by slightly modifying the notations. The first partial group starts from \( d_{n,1} \) and the second partial group from \( d_{n,2} \).

The order of partial data groups before de-fragmentation is given as follows:
- The first fragment is \( (d_{1,1}) \). Its start time position is zero.
- Each partial data group, \( k \), runs from 1 to \( n - 1 \).
- The number of EN fragments in a sequence of partial data groups is decremented linearly i.e. \( d_{n,k}, d_{n-1,k}, \ldots, d_{m,k}, \ldots, d_{k+1,k} \).

### B. Data after De-fragmentation

There are ways to de-fragment the data: numbered ordering and appearance ordering. In numbered ordering, the merged fragments are ordered according to EN number, from first to last following the order of frame receipt. In appearance
ordering, the merged fragments are ordered according to the order of the first fragment from each EN in the frame before de-fragmentation.

1) Numbered Ordering: Numbered ordering is the order of merged fragments yielded by numbered ordering; \( m \) runs from 1 to \( n \). In the same EN data group, fragments are arranged in ascending order, i.e. \( k \) runs from 1 to \( m - 1 \).

2) Appearance Ordering: The merged fragments are ordered according to the order of the first partial groups in the frame before de-fragmentation. Since all ENs are assumed to have data to send, \( d_{1,1} \), the fragment from the first EN, always occupies the head of the frame and the next fragment is from the last EN, \( d_{n,1} \). It easy to see that the order before de-fragmentation is \( d_{1,1}, d_{n,1}, d_{n-1,1}, \ldots, d_{2,1} \). After de-fragmentation, the merged fragments are arranged in the matching order 1, \( n \), \( n-1 \), ..., 2. The pointer of fragment \( d_{1,1} \), is 0. Within each merged EN data group, the data fragment is arranged in ascending order.

C. Maximum Delay Analysis

Since the pointer of each data fragment before and after de-fragmentation have been found, the time offset of the \( k \)th data fragment of the \( n \)th EN, \( t_{m,k} \), is given. For both numbered and appearance ordering, \( t_{m,k} \) is always zero since \( d_{1,1} \) is always the first data group.

Numbered ordering:

\[
t_{m,k} = \sum_{i=2}^{n} \sum_{j=1}^{m} d_{i,j} - \sum_{i=1}^{n-m} d_{n-i,k} - \sum_{j=k+1}^{n} \sum_{i=1}^{m-1} d_{n-i+1,j} - \sum_{i=2}^{n} \sum_{j=1}^{m-1} d_{i,j} - \sum_{i=1}^{k-1} d_{m,i}
\]  (1)

Appearance ordering:

\[
t_{m,k} = \sum_{i=2}^{n} \sum_{j=1}^{m} d_{i,j} - \sum_{i=1}^{n-m} d_{n-i,k} - \sum_{j=k+1}^{n} \sum_{i=1}^{m-1} d_{n-i+1,j} - \sum_{i=1}^{n-m} d_{n-i+1,j} - \sum_{i=1}^{k-1} d_{m,i}
\]  (2)

Eq. (1) and Eq. (2) are the objective functions to be maximized for numbered ordering and appearance ordering, respectively. The coefficient of \( t_{m,k} \) might be plus or minus as follows:

- Plus: data fragment \( d_{m,k} \) is advanced relative to \( t_{m,k} \) to overtake the other fragments.
- Minus: data fragment \( d_{m,k} \) is delayed relative to \( t_{m,k} \) to achieve it's desired position.

A relative advance is achieved by delaying all other fragments so de-fragmentation delays frame transmission.

Since the objective function and constraints are expressed in linear form by decision variables, linear programming is used to determine the maximum delay, \( D_{\text{max}} \). By solving this linear programming problem, \( D_{\text{max}} \) is obtained as \( F-2 \) with numbered ordering and \( F-3 \) with appearance ordering. This means that the maximum delay is \( F-2 \) or \( F-3 \) depending on the type of ordering employed.

VI. Conclusion

A real-time allocation scheme based on dynamic replacement, called the DR scheme, was proposed. It combines the advantages of general allocation mode, reservation, and real-time. For example, the frame need be circulated just one time to assign bandwidth and multiplex the data into a frame while fairness is still assured. Bandwidth assignment is processed at each EN. Data placed in the frame by earlier EN can be partially replaced by the data from a later EN if the frame is full. Since the data from one EN is likely to become fragmented due to replacement, it is de-fragmented using an adaptive delay line at the local RNN before passing the frame to the backbone network in the optical domain. We verified the fairness of the DR scheme under imbalanced input traffic patterns. Simulations showed that it achieved fairness in terms of throughput. The delay imposed by de-fragmentation was investigated to determine the hardware amount for de-fragmentation. We introduce two different types of order, numbered and appearance ordering. Hardware complexity was evaluated by finding the maximum delay. The maximum delay as determined by linear programming was \((F-2)T\) and \((F-3)T\) for numbered ordering and appearance ordering, respectively. This indicates that \( F-2 \) and \( F-3 \) delay lines should be prepared, respectively to support those ordering schemes.

REFERENCES


