Path Accommodation Methods for the Bidirectional Ring with Optical Compression TDM

Kazuhiro Gokyu,
Ken-ichi Baba
and
Masayuki Murata
Osaka University

In this paper, we propose path accommodation methods for bidirectional rings based on an optical compression TDM (OCTDM) technology. We first derive a theoretical lower bound on the number of slots and frames, which is necessary to allocate paths among all nodes. The relationships between the lower bound and such parameters as the compression rate and the numbers of transmitters/receivers are then discussed. Three path accommodation algorithms are next proposed to achieve the lower bound as close as possible. Through numerical examples, we show that each of three algorithms has best parameter regions achieving the best results. Thus, our recommendation for off-line path accommodation is therefore that three algorithms are all performed, then the best one is chosen. In on-line permutation, the method must be planed in order to re-accommodate the all paths as soon as possible. Finally, we analyze the packet delay time for given number of slots and frames that our algorithms decided. Numerical examples show the characteristics of packet delays in various conditions.

Categories and Subject Descriptors: []

Additional Key Words and Phrases: Optical Compression TDM, Path Accommodation Method, Optical Bidirectional Ring Network, Theoretical Lower Bound, Analysis of Packet Delay

1. INTRODUCTION

According to a rapid growth of a user population and multimedia applications on the Internet, traffic volume on backbone networks has been dramatically increasing. In the metropolitan area, the backbone ring is promising to build MAN (Metropolitan Area Network), which inter-connects LANs (Local Area Networks) of companies, laboratories, ISPs (Internet Service Providers), and so on. An optical-electric exchange was conventionally used in routers or switches for the access from LAN to MAN.

An earlier version of this paper appeared in the *Proceedings of Workshop on Optical Networks 2000* (Jan. 2000) by CD-ROM.
Name: K. Gokyu
Affiliation: Dept. of Informatics and Mathematical Science, Graduate School of Engineering Science, Osaka University
Address: 1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan
Name: K. Baba and M. Murata
Affiliation: Cybermedia Center, Osaka University
Address: 5-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan

However, an all-optical access without the optical-electric exchange is now essential to realize high-speed MANs (see, e.g., [1; 2]).

A packet-switched ring with all-optical access can be realized by optical wavelength-division multiplexing (WDM) or optical time-division multiplexing (OTDM) techniques. In WDM, high speed transmission over 10 Gbps on a single wavelength can be obtained, but a limited number of wavelengths inhibits many-to-many communication among nodes at the same time. In the OTDM system, on the other hand, it is easier to control communication facilities than in the WDM system, and a recent development of an optical pulse compression/expansion technology makes it more attractive [3; 4], by which the backbone network with one to two magnitudes of larger capacities can be provided. In [3], for example, the backbone ring with an optical pulse compression/expansion technology, called OCTDM (Optical Compression TDM), is now being produced experimentally, where LANs with 622 Mbps are interconnected by the backbone OCTDM ring with one to tens of Gbps. In their proposal, when the optical node of the OCTDM ring receives the packet from LAN, bit intervals are shortened to fit the transmission rate of the backbone ring. The packet is then lengthened at the destination node. See Section 2 for more details.

In OCTDM, we need some routing policy on how each slot within the frame is used by nodes. In conventional TDM, one possible and natural way to accommodate the traffic on the ring is to assign the slot as follows; the number N of nodes on the ring is numbered from 0 to $N-1$. Then, ith slot within the frame (consisting of N slots) is assigned to ith source node, so that ith source node always transmits the packet on ith slot within destination address. In this scheme, the destination node may receive at most $N-1$ packets during the frame time. In OCTDM, however, the number of slots that the node can access within the frame is limited by the number of costly receivers [5]. Since the number of transmitters is also limited, another method that ith slot is reserved for ith destination node is not adequate for the OCTDM ring as well. Accordingly, we need a path accommodation method suitable to the OCTDM ring by taking account of those facts. Here, by "path", we mean the assigned slot(s) for a pair of source/destination nodes.

In the path accommodation method considered in this paper, paths between every source/destination pair are determined a priori for given traffic load matrix. In the receiver-oriented approach in which ith slot is reserved for ith destination node described above, some contention resolution mechanism among the source nodes is necessary. It must introduce an unacceptable overhead. Another approach also exists; a medium access control method such as a token passing method (DQDB or FDDI) might be applied to the OCTDM ring. However, the delay of control signals including the propagation delays between nodes much degrades performance of the OCTDM ring with very high-speed capacity. We thus do not consider such approaches in the current paper.

As a related work, the path design method for the WDM ring is reported in [6], where the cost effective design method is proposed for accommodating the *wavelength path* for every node pair. In their method, the number of wavelengths is a limited resource. In their companion paper [7], the time needed to accommodate all paths with the given number of wavelengths is also obtained. They consider the fixed packet length, and therefore, the time is slotted in the WDM system. Thus, their system becomes similar to our OCTDM ring. Actually, we will borrow their idea in one of path accommodation methods that we will examine in this paper. Further, we extend their method in [7] to make it possible to treat the heterogeneous traffic load for node pairs, while the original method assumed the homogeneous traffic load. In [8], they refer to the heterogeneous one for SONET/WDM rings. In this paper, however, the method for OCTDM rings is discussed. Also, the theoretical lower bound presented in Section 3 is an extension of the approach described in [7].

The rest of the paper is organized as follows. In Section 2, we briefly describe the OCTDM ring structure and our model. In Section 3, we develop the lower bound on the number of frames necessary to accommodate all paths for given parameters (the numbers of transmitters/receivers and the compression rate), and investigate the relationship between the lower bound and system parameters. In Section 4, three path accommodation algorithms are considered. The effectiveness of those algorithms is then compared based on the theoretical lower bounds shown in Section 3. In Section 5, we analyzed the packet delay time, which showed the importance of the effective path accommodation. Conclusions and future works are summarized in Section 6.

2. THE STRUCTURE OF OCTDM RING NETWORKS

2.1 Optical Pulse Compression/Expansion Technique

An optical pulse compression/expansion technology is promising to realize the very high-speed backbone ring [5]. When the packet is put on the optical line, a bit interval is compressed by using the fiber delay loop (Figure 1). Since the compression rate with one loop is limited, high compression rate can be achieved by using several steps if it cannot be realized at a time. A semiconductor optical amplifier (SOA) and the switch (SW) are inserted on the loop to compensate the loss on the fiber delay loop. Then, the packet is transmitted onto the ring. When the packet is received from the optical line to LAN, bit expansion is performed as a reverse procedure of bit compression. More details of the optical pulse compression technique are described in [9; 10; 11].

2.2 Access Method to Rings

We consider a bidirectional ring consisting of two unidirectional, working fiber links; one clockwise and the other counter-clockwise.

Each ring of two directions is time-slotted. The packet arriving at the node is divided into *mini-packet*s at the source node. The mini-packet with additional header is put in the slot pre-assigned for the source/destination pair. The assignment method of slots is our objective of the current paper, which will be described in the following sections. The mini-packet is transmitted to the ring after it is optically compressed with K times, and each frame consists of K slots. That is, the compression rate K is identical to the number of slots within the frame.

There are N nodes on the ring. The nodes are numbered clockwise from 0 to $N-1$. Nodes i and $mod(i+1, N)$ is connected by link i. See Figure 2. The node structure is shown in Figure 3. We assume that node i has the number T_i of transmitters and R_i of receivers. Each transmitter can send only one mini-packet per frame in either direction. Each receiver can receive only one mini-packet per frame as well. That is, transmitters and receivers are shared in both directions. The number of mini-packets that each node can transmit (receive) in the frame is thus limited by the number of transmitters (receivers). Noting that each node cannot receive more packets than the number of the receivers in one frame, an optical memory is unnecessary. It is attractive since the optical memory technology is still not mature for packet buffering.

Since the compression rate K is usually much smaller than the number of nodes N (which is a natural assumption from the current and probably future technologies), slots for all source/destination node pairs cannot be put within the frame. We will call a *super-frame* for the number of frames which is necessary to accommodate all paths (i.e., the number of slots for every pair of source/destination nodes). In Section 3, we will derive a theoretical lower bound for the length of the super-frame. Then, in Section 4, we will propose three path accommodation algorithms to assign slots to all nodes for given traffic load matrix. The effectiveness of those path accommodation algorithms is then investigated by comparing the theoretical lower bound.

3. DERIVATION OF THEORETICAL LOWER BOUNDS

3.1 Introduction of Notations

Let $\mathcal{T} = \{T_0, T_1, \cdots, T_{N-1}\}$ and $\mathcal{R} = \{R_0, R_1, \cdots, R_{N-1}\}$ be sets of the transceivers and receivers, respectively.

The path from source node i to destination node $(i + s)$ is represented by (i, s), where s is the clockwise distance in hops between two nodes. A counter-clockwise distance is represented by the negative number. However, to make representation simple in the following analyses, we will allow to use some k $(\geq N)$ in representing the node number. In that case, k should be read as $mod(k, N)$. Similarly, the negative values of the node number and the node distance are also allowed. Namely, $k \leftarrow N - mod(|k|, N)$ if $k < 0$. For example, in the case of $N = 64$, node 68 means node 4, and node -7 does node 57. The node distance -13 shows the distance 51 if we consider the distance clockwise.

In this and next sections, we assume that the traffic load is expressed in integer values, i.e., the required number of slots for path (i, s) takes an integer value $c^{(i,s)}$. A N by N matrix $C = \{c^{(i,s)}\}$ is given as the traffic load matrix. Hereafter, we implicitly assume that the total sum of the traffic load does not exceed the backbone ring capacity, so that it is always possible to accommodate all of paths.

Since we consider the bidirectional ring, there is a freedom to choose the path for two nodes clockwise or counter-clockwise. We assume to use the shorter path. When the number of nodes is even, there are two shortest paths between node i and $(i + \frac{N}{2})$. In that case, we use the path for $(i, \frac{N}{2})$ as follows;

—If $0 \leq i \leq \lfloor \frac{N}{4} \rfloor - 1$, or $\frac{N}{2} \leq i \leq \lfloor \frac{N}{4} \rfloor + \frac{N}{2} - 1$, the path is set up clockwise.

—Otherwise, use the path counter-clockwise.

3.2 Derivation of Lower Bounds

In this subsection, we derive the lower bounds of the super-frame length for given N (the number of nodes), $\mathcal{T}$ (a set of the number of transmitters), $\mathcal{R}$ (a set of the number of receivers), K (a compression rate of OCTDM), and C (a traffic load matrix). We define it as $LB(N, \mathcal{T}, \mathcal{R}, K, C)$. Note that the theoretical lower bound in WDM rings was studied in [7] under the conditions (1) that the numbers of transmitters and receivers provided by all nodes are identical, and (2) that the traffic load is uniform. We extend the method presented in [7] for our OCTDM ring under the non-uniform traffic load.

(A) The case where $\mathcal{T}$ and $\mathcal{R}$ are infinite, and K is finite. We first consider the case where the compression rate K is finite, but the numbers of transmitters/receivers at every node are infinite. We denote the total number of clockwise (counter-clockwise)

paths on the link i by ${n_R}^{(i)}$ (${n_L}^{(i)}$), which can be determined from traffic load matrix, C, as:

$$
\begin{aligned}
{n_R}^{(i)} &= \sum_{j=(i+1)+\lceil \frac{N}{2} \rceil}^{i+N} \sum_{s=(i+N+1)-j}^{\lfloor \frac{N}{2} \rfloor} c^{(j,s)} \cdot r_d^{(j,s)}, && (1) \\
{n_L}^{(i)} &= \sum_{j=i+1}^{i+\lfloor \frac{N}{2} \rfloor} \sum_{s=\lceil \frac{N}{2} \rceil}^{i+N-j} c^{(j,s)} \cdot l_d^{(j,s)}, && (2)
\end{aligned}
$$

where $r_d(i,s) = 1$ if path (i,s) is set up clockwise; $r_d^{(i,s)} = 0$ if counter-clockwise. Similarly, $l_d^{(i,s)} = 0$ if path (i,s) is set up clockwise; $l_d(i,s) = 1$ if counter-clockwise.

Since each frame has K slots, the number K of paths can be set up in each frame on link i in either of two directions. It then requires $\lceil \frac{{n_R}^{(i)}}{K} \rceil$ frames to allocate all of clockwise paths on link i. It is also true for counter-clockwise paths. The theoretical lower bound of the super-frame length, $LB(N, \infty, \infty, K, C)$ is thus given as:

$$
LB(N, \infty, \infty, K, C) = \max_{0 \le i \le N-1} \left(\left\lceil \frac{{n_R}^{(i)}}{K} \right\rceil, \left\lceil \frac{{n_L}^{(i)}}{K} \right\rceil \right). \quad (3)
$$

(B) The case where K is infinite, and $\mathcal{T}$ and $\mathcal{R}$ are finite. The total number of paths from sender node i to the other receiver nodes is given by:

$$
{s_p}^{(i)} = \sum_{s=1}^{N-1} c^{(i,s)}. \quad (4)
$$

Similarly, the total number of paths from sender nodes except node i to the receiver node i is given by:

$$
{r_p}^{(i)} = \sum_{k=0}^{N-1} c^{(k,i-k)}. \quad (5)
$$

The infinite compression rate ($K = \infty$) means that the number of slots in each frame is infinite. The number of paths allocated for node i is bounded by the numbers of transmitters, (T_i) and receivers (R_i). That is, $LB(N, \mathcal{T}, \infty, \infty, C)$ and $LB(N, \infty, \mathcal{R}, \infty, C)$ are derived as;

$$
\begin{aligned}
LB(N, \mathcal{T}, \infty, \infty, C) &= \max_{0 \le i \le N-1} \left(\left\lceil \frac{{s_p}^{(i)}}{T_i} \right\rceil \right), && (6) \\
LB(N, \infty, \mathcal{R}, \infty, C) &= \max_{0 \le i \le N-1} \left(\left\lceil \frac{{r_p}^{(i)}}{R_i} \right\rceil \right). && (7)
\end{aligned}
$$

From two cases (A) and (B) above, we can obtain $LB(N, \mathcal{T}, \mathcal{R}, K, C)$ using Eqs. (3),(6),(7) as follows:

$$LB(N, \mathcal{T}, \mathcal{R}, K, C) = \max_{0 \le i \le N-1} \left(\left\lceil \frac{n_R^{(i)}}{K} \right\rceil, \left\lceil \frac{n_L^{(i)}}{K} \right\rceil, \left\lceil \frac{s_p^{(i)}}{T_i} \right\rceil, \left\lceil \frac{r_p^{(i)}}{R_i} \right\rceil \right). \tag{8}$$

From Eq. (8), we can observe that the length of the super-frame can become smaller if terms in Eq. (8) are uniformly distributed for given numbers of transmitters/receivers and the compression rate. Then we have the ring with high throughput. We will give quantitative examples in the following subsection.

3.3 Numerical Results and Discussions

In this subsection, we investigate the relationship between the lower bounds of the super-frame length and several system parameters including the number of nodes, the numbers of transmitters and receivers, and the compression rate. In the OCTDM ring, the increase of the number of transceivers leads to the shorter super-frame, which results in decreasing the lower bound. However, it needs the large optical buffer (realized by the optical lines) [12]. Even when the number of transceivers can be provided, the super-frame length is not decreased if the compression rate K is small. It is because the compression rate poses a limit on the number of slots assigned for each node. Then, the throughput cannot be improved.

We first consider the uniform traffic load condition. The traffic matrix C_1 has elements with all 1's, i.e., $c^{(i,s)} = 1$ except elements, $c^{(i,0)} = 0$. The number of nodes is set to be 64. See Figure 4(a). Every node on the ring has the identical number T of transmitters, i.e., $\mathcal{T} = \{T, T, \cdots, T\}$. Similarly, the number of receivers of all nodes is also identically set to be R. Figure 5 plots the lower bounds for traffic matrix C_1. From the figure, we can observe that as the numbers of transmitters and receivers are increased, the lower bound shows a gradual decrease, and finally becomes constant. For instance, in the case of the compression rate of 20 (i.e., $K = 20$), the lower bound becomes constant when $T = R \ge 3$. That is, the lower bound is not limited by the numbers of transmitters/receivers but the compression rate in this parameter region. If $T = R = 1$ (which is a current technological level), then the compression rate of $K \ge 9$ does not help improve the lower bound (and the total throughput of the ring).

We next investigate the case of the non-uniform traffic demand. As shown in Figure 4(b), the traffic demand to the receiver node 63 is increased to 2 in the traffic matrix C_2. Other demands are same as in the traffic matrix C_1 (Figure 4(a)). (Note that the traffic matrices C_3, C_4, C_5, and C_6 in Figures 4(c), 4(d), 4(e), and 4(f) will be used in the next section.) The lower bounds for C_2 are plotted in Figure 6. In obtaining this figure, the number of transmitters/receivers at every node is identically set. Figure 6 shows the substantial increase of the lower bounds when compared with Figure 5. For example, the length of the super-frame is increased from 32 to 64 for $T = R = 2$ and $K \geq 16$. It means that the maximum throughput of the ring is halved with a slightly increased traffic to the receiver node 63.

If we add the receiver at node 63, however, the length of the super-frame can be again decreased. The results are presented in Figure 7. As shown in the figure, the number of receivers at node 63 is increased by one, while the numbers of transmitters/-receivers at other nodes remain unchanged. Namely, the resource balance is very important to attain the short super-frame in the OCTDM ring.

In this section, we have considered the theoretical lower bounds given by Eq. (8). However, Eq. (8) only provides the bound, and the path accommodation algorithm is necessary to actually allocate the path(s) (i.e., slot(s)) to all source/destination pairs of nodes and to determine the length of the super-frame, which will be described in the next section.

4. PATH ACCOMMODATION ALGORITHMS AND COMPARISONS

In this section, we first describe three path accommodation algorithms in Section 4.1. It is difficult to obtain an optimal allocation since examination on all combinations is necessary to find it. Three accommodation algorithms presented in the below are all heuristic. The degree of an optimality of those algorithms is investigated by comparing with the lower bounds developed by the previous section, which will be presented in Section 4.2.

4.1 Path Accommodation Algorithms

In the algorithm A1, the path with the largest distance is chosen first to allocate the path. The traffic loads on links and nodes are taken into account in the algorithm A2. The algorithm A3 is an extension of the one described in [7], where the paths are determined according to quadrilateral paths accommodation. In what follows, we will describe those algorithms in turn.

4.1.1 *Algorithm A1: The longer path is assigned first.*

Algorithm A1

```
the_super-frame_length = 1
while( every path cannot be set up ){
  if( a path cannot be set up at all )
    the_super-frame_length++
  for( s = ⌊N/2⌋; s ≥ 1; s-- )
    for( i = 0; i ≤ N-1; i++ )
      CEP( i, s )
      CEP( i, N-s )
}
Decision of the_super-frame_length
```

We first describe the algorithm A1, which attempts to assign slots to the longest path. It is simple that it does not consider the traffic load condition on every link and node. However, as will be demonstrated in the next subsection, the algorithm A1 outperforms other two algorithms in some parameter sets.

The algorithm A1 first finds the source/destination pairs requesting the path with longest distance (i.e., $s = \lfloor \frac{N}{2} \rfloor$). For those paths, the slot is assigned from source node 0 to $(N-1)$ if the source/destination pair requests such a path. The transmitter for the source node and the receiver for the destination node are also examined. In doing so, the path is examined clockwise and counter-clockwise alternately. Then, next longest paths with distance $s = \lfloor \frac{N}{2} \rfloor - 1$ are assigned. All paths are examined until paths with distance 1 are assigned slots.

See algorithm A1 for its procedure. The procedure CEP(i, s) (Check and Establish a Path) in the algorithm first checks to see if transmitters, receivers, and slots are available to set the path (i, s) when $c^{(i,s)} \geq 1$. If it is true, path (i, s) is actually set. Then, $c^{(i,s)}$ is decremented by one.

4.1.2 *Algorithm A2: The path with the highest traffic load is set first.*

Algorithm A2

```
the_super-frame_length = 1
while( every path cannot be set up ){
   CALC(n_R, n_L, s_p, r_p, C_W)
   while( a path can be set up ){
      CALC(max_cw_i, max_cw_s)
      if(CP(max_cw_i, max_cw_s)){
         EP(max_cw_i, max_cw_s)
      }
      else {
         w^(max_cw_i,max_cw_s) = 0
      }
   }
   the_super-frame_length++
}
Decision of the_super-frame_length
```

In the second algorithm A2, the path using most links, transmitters, and receivers is first set. More specifically, the algorithm works as follows. Let us introduce a $N \times N$ traffic weight matrix $C_W = \{w^{(i,s)}\}$. It shows the sum of the weighting factors on the link, transmitter, and receiver along path (i, s), the element of which is determined as follows:

—If path (i, s) is set clockwise because it has a smaller distance, then

$$w^{(i,s)} = \left\lceil \frac{\sum_{k=i}^{i+s-1} n_R{}^{(k)}}{K} \right\rceil + \left\lceil \frac{s_p{}^{(i)}}{T_i} \right\rceil + \left\lceil \frac{r_p{}^{(i+s)}}{R_{i+s}} \right\rceil . \tag{9}$$

—If path (i, s) is set counter-clockwise, then

$$w^{(i,s)} = \left\lceil \frac{\sum_{k=i+s}^{i+N-1} n_L{}^{(k)}}{K} \right\rceil + \left\lceil \frac{s_p{}^{(i)}}{T_i} \right\rceil + \left\lceil \frac{r_p{}^{(i+s)}}{R_{i+s}} \right\rceil . \tag{10}$$

—$w^{(i,s)} = 0$ if $c^{(i,s)} = 0$.

The setup is first tried for the path with a maximum value of C_W. During the algorithm execution, the traffic weight matrix C_W should reflect the changes of the traffic load matrix C every frame such that the paths having been already set up are excluded.

See algorithm A2. The variables `max_cw_i` and `max_cw_s` are the row and column numbers of the maximal element of C_W, respectively. The function `CP(`i, s`)` (Check a Path) checks utilizations of transceivers, receivers and slots if path (i, s) can be set up when $c^{(i,s)} \geq 1$. It returns ‘true’ if those resources can be used; otherwise it returns ‘false’. The procedure `EP(`i, s`)` (Establish a Path) sets path (i, s) and then execute $c^{(i,s)} = c^{(i,s)} - 1$ if $c^{(i,s)} \geq 1$. The procedure `CALC(`$a_1, a_2, \cdots$`)` calculates the parameters $a_1, a_2, \cdots$ from C at a time.

4.1.3 *Algorithm A3: an extended CADS*

CADS Algorithm (N is even)

Step 1:
`for` $s = \frac{N}{2}$ `(a special case)`
`for` $i = 0, 1, \cdots, \frac{N}{4} - 1$
`Two paths in the following`
`are set up at a time if possible.`
$(i, \frac{N}{2}), (\frac{N}{2} + i, \frac{N}{2})$
Step 2:
`for` $s = \frac{N}{4}$ `(a special case)`
`for` $i = 0, 1, \cdots, \frac{N}{4} - 1$
`Four paths in the following`
`are set up at a time if possible.`
$(i, \frac{N}{4}), (\frac{N}{4} + i, \frac{N}{4}),$
$(\frac{N}{2} + i, \frac{N}{4}), (\frac{3N}{4} + i, \frac{N}{4})$
Step 3:
`for` $s = 1, 2, \cdots, \frac{N-2}{4}$ `(general case)`
`for` $i = 0, 1, \cdots, \frac{N}{2} - 1$
`Four paths in the following`
`are set up at a time if possible.`
$(i, s), (i + s, \frac{N}{2} - s),$
$(\frac{N}{2} + i, s), (\frac{N}{2} + i + s, \frac{N}{2} - s)$

The algorithm A3 is a modified and extended version of the CADS algorithm, which was originally proposed for the WDM ring [7]. Note that in [7], the CADS algorithm is applied for the case where the number N of nodes is even and the CATS algorithm is for N odd, but those algorithms are essentially same. Therefore, we only consider the CADS algorithm here. We first show a part of the CADS algorithm where paths are set up clockwise. In the CADS algorithm, two paths between two nodes, which are located on the diagonal of rings are chosen at the same time. Those are then set up along the same direction. It is performed in Step 1. Then, four paths corresponding to edges of the rectangle are set up at the same time along the same direction in Steps 2 and 3.

The CADS algorithm can only be applied to the case where the traffic load is uniform. It chooses a combination of the long-distance and short-distance paths to construct the rectangle. Without loosing the basic philosophy of the algorithm, we extend it to be applicable to any traffic load matrix. It is the algorithm A3. When the traffic load is non-uniform, every paths are not always set up at the same time. Thus, each path should be set up independently. Note that the algorithm A3 works exactly same as the CADS algorithm when the traffic load is uniform.

Algorithm A3

```
the_super-frame_length = 1
while( every path cannot be set up ){
  if( a path cannot be set up at all )
    the_super-frame_length ++
  for(i = 0 i <= N/4 - 1; i++){ (Step 1)
    CEP(i, N/2)
    CEP(i + N/2, N/2)
  }
  for(i = 0; i <= N/4 - 1; i++){ (Step 2)
    //clockwise
    CEP(i, N/4)
    CEP(i + N/4, N/4)
    CEP(i + N/2, N/4)
    CEP(i + 3N/4, N/4)
    //counter-clockwise
    CEP(i, 3N/4)
    CEP(i + 3N/4, 3N/4)
    CEP(i + 6N/4, 3N/4)
    CEP(i + 9N/4, 3N/4)
  }
  for(s = 1; s <= (N-2)/4; s++){ (Step 3)
    for(i = 0; i <= N/2 - 1; i++){
      //clockwise
      CEP(i, s)
      CEP(i + s, N/2 - s)
      CEP(i + N/2, s)
      CEP(i + N/2 + s, N/2 - s)
      //counter-clockwise
      CEP(i, N - s)
      CEP(i + N - s, N - (N/2 - s))
      CEP(i + 3N/2, N - s)
      CEP(i + 5N/2 - s, N - (N/2 - s))
    }
  }
}//while
Decision of the_super-frame_length
```

4.2 Comparisons of Three Path Accommodation Algorithms

In this subsection, we compare three algorithms A1, A2 and A3 presented in the previous subsection. The number of nodes, N, is fixed at 64. The numbers of transmitters and receivers per node are assumed to be identical; i.e., $T = R$. For the traffic load matrix, we will consider C_3 (Figure 4(c)), C_4 (Figure 4(d)), C_5 (Figure 4(e)), and C_6 (Figure 4(f)) in addition to previously used C_1 and C_2. Characteristics of those matrices are summarized as follows.

—C_1: a uniform traffic load.

—C_2: all paths except the ones with receiver node 63 are uniform. The load of paths from any source node to receiver node 63 is two times larger than that of others.

—C_3: similarly to C_2, all paths except the ones with receiver node 63 are uniform. The load of any node to receiver node 63 is three times larger than that of others. That is, non-uniformity of the traffic load of C_3 is higher than C_2.

—C_4: all paths except the ones with receiver nodes 33 and 63 are uniform. The load of paths from any node to receiver nodes 33 and 63 is three times larger than that of others.

—C_5: the load of each paths is decided from 0 through 3 at random.

—C_6: each of all paths is decided from 0 through 7 at random.

Figure 8 compares the theoretical lower bounds on the length of the super-frame (labelled by '*LB*') derived in Section 3 and the results of three algorithms, dependent on the compression rate K. Figures 8(a) through 8(f) are for the traffic matrices C_1, C_2, C_3, C_4, C_5, and C_6, respectively. In these figures, the number of frames achieving the smallest value among three algorithms is shown with bold-face. From Figure 8(a), we can observe that the algorithm A3 is the best choice for the uniform matrix C_1. However, for matrices C_2, C_3, C_4, C_5, and C_6, the algorithm A2 also exhibits good results as the algorithm A3. It is especially true for small values of K. It is a good feature of algorithm A2 if we consider the current technological level of the optical compression rate. We also note that algorithm A1 becomes best in rather extreme cases including the cases of $T = R = 8$ and $K = 1$ for matrix C_4.

A more close look at the results is presented in Figures 9 through 14 using matrices C_1, C_2, C_3, C_4, C_5, and C_6, respectively. In each of those figures, results of the algorithm A1, A2 and A3 are plotted. In obtaining all figures, the numbers of transmitters and receivers (T and R) are identically set and varied from 1 to 12, and the compression rate K are from 1 to 64. The numbers achieving the theoretical lower bound are represented by '■', and the best result among three algorithms is shown by '$\times$'. As mentioned in the above, the algorithm A3 is appropriate for matrix C_1, excluding the case of $T = R = 1$. On the other hand, best results are often provided by the algorithm A2 for matrices C_2, C_3, C_4, C_5, and C_6. It means that the algorithm A2 considering traffic loads is effective as we expect.

In summary, the algorithm A3 provides better results than others especially when the traffic load is uniform. If the traffic load is non-uniform, however, the algorithm A2 also works well and often provides better results than the algorithm A3. The effectiveness of the algorithm A1 is limited, but it gives best result in several cases. That is, we could not find the best algorithm working best in all parameter regions. However, these comparisons showed the decided tendency, which helps us to select the algorithm for the most parameter. Thus, our conclusion and recommendation is as the following itemize.

—When we apply these algorithms to set up the connections before communicating among nodes (i.e. off-line path accommodation), we can select the most effective algorithms for a little long time, compared with the on-line permutation. Thus, the all three algorithms should be performed to find the best solution.

—If we must dynamically change the path scheduling for the traffic rearrangement (i.e. on-line path accommodation), the architecture is required to complete the accommodation as soon as possible. Accordingly, we have no time to find the best algorithm of the three. Thus, we make a proposal as the one of the solutions for this example; the Algorithm A3 is selected for the uniform traffic load, and the Algorithm A2 is selected for the non-uniform traffic load.

5. ANALYSIS OF PACKET DELAY TIMES

In Sections 3 and 4, we have considered that the traffic load is given in unit of slot times. Then, we considered the path accommodation algorithms which determine the super-frame length to accommodate all traffic. In doing so, we have assumed that the total traffic load does not exceed the backbone ring capacity. In this section, we again assume it to derive the delay time of packets arriving at the node.

We first assume the uniform traffic load. By assuming it, only one slot is assigned within the super-frame for given source/destination pair. When the packet consisting of the multiple mini-packets arrives, each mini-packet is transmitted using the slot. That is, after the packet with n mini-packets reaches the head of the queue at the source node, it takes the super-frame length (in time) multiplied by n to transmit all mini-packets.

In the following analysis, we will assume that packets arrive at the source node according to the Poisson distribution. The number of mini-packets contained in the packet follows the general distribution. Then, we can utilize the result of the M/G/1 queueing system to derive the packet delay time. We note here that the packets are first buffered at the LAN side, and therefore, we do not need optical buffer to store the packets at the source node.

In what follows, we will consider the source/destination pair of the path (i, s) to derive the packet delay. The uniform traffic load is first treated in Section 5.1. The result presented in Section 5.2 provides the approximate packet delay for the non-uniform traffic load.

5.1 Analysis of Packet Delay Times for the Uniform Traffic Load Case

By letting the LAN capacity be B_L [bps], and the unidirectional ring capacity B_R [bps], we have the relation, $B_R = K \cdot B_L$. One slot time denoted by t [s] is given by

$$t \;=\; \frac{(S_h + S_p)\cdot 8}{B_R}, \tag{11}$$

where S_h [byte] and S_p [byte] are the header and payload sizes of the mini-packet. The propagation delay between nodes i and $(i+s)$ is denoted by $W_p^{(i,s)}$ [s]. Further, the number of frames in the super-frame is represented by r, which has been determined by our path accommodation algorithms in the previous section. Then, the number of slots contained in the super-frame, D, is given by $K \cdot r$ where K is a compression rate of the optical ring.

We assume that at source node i, packets arrive according to a Poisson distribution with rate $\lambda^{(i,s)}$ destined for node $(i+s)$. Hereafter, we will derive the mean packet delay for this stream. The packet length in bytes has a general distribution with probability function f, and we represent its mean by P_B [byte]. The traffic load (in bps) for path (i,s) is then given by

$$B_f^{(i,s)} = \frac{\lambda^{(i,s)} \cdot P_B \cdot 8}{t}. \tag{12}$$

Further, we introduce the random variable P_m, representing the number of mini-packets in the packet. Its probability function, $g(n)$ $(n = 1, 2, \ldots)$, is given by

$$g(n) = \mathrm{Prob}[P_m = n] = \sum_{x=S_p(n-1)+1}^{S_p \cdot n} f(x). \tag{13}$$

Our objective is to derive the packet delay time $W^{(i,s)}$ [s] on path (i,s), which consists of four components;

$$W^{(i,s)} = \left[\frac{D}{2} + W_q^{(i,s)} + \left(E[T_F] - (D-1)\right)\right] \cdot t + W_p^{(i,s)}. \tag{14}$$

In what follows, we will consider each term of the right hand side of the above equation. The fourth term, $W_p^{(i,s)}$, is the propagation delay from source node i to destination node $(i+s)$. The first term in braces is necessary because we consider the random arrival of packets, and the packet should wait the half of the super-frame in average so that the first mini-packet can be put on the slot assigned to that path.

We next examine the third term in braces. The random variable T_F [slots] in the term shows the mean time to transmit all mini-packets contained in the packet from the time when the designated packet reaches the head of the queue. Since it needs the number $E[P_m]$ of super-frames, the following equation holds;

$$E[T_F] = D \cdot E[P_m]. \tag{15}$$

The subtraction of $D-1$ from $E[T_F]$ is necessary since we consider the time interval until the last mini-packet is put onto the ring in this term.

The second term of the right hand side in Eq. (14), $W_q^{(i,s)}$, corresponds to the queueing time at the source node buffer until the packet reaches the head of the queue. By applying the Pollaczek-Khinchin formula, it can be obtained by

$$W_q^{(i,s)} = \frac{\lambda^{(i,s)} E[T_F^2]}{2(1-\lambda^{(i,s)} E[T_F])}, \tag{16}$$

where $E[T_F^2]$ is given by $D^2 E[{P_m}^2]$.

By rewriting Eq. (14), we finally have

$$W^{(i,s)} = \left[\frac{\lambda^{(i,s)} D^2 E[{P_m}^2]}{2(1-\lambda^{(i,s)} D E[P_m])} + D\left(E[P_m] - \frac{1}{2}\right) + 1\right] \cdot t + W_p^{(i,s)}. \tag{17}$$

5.2 Extensions to the Case of Non-Uniform Traffic Load

In the case of non-uniform traffic load, two or more slots may be assigned within a single super-frame for the source/destination pair. The positions of assigned slots depend on the path accommodation algorithm, and the intervals of slots may be irregular. Those make it impossible to derive the packet transmission time in a generic form as in the previous subsection.

Here, we introduce the assumption that assigned slots are uniformly distributed within the super-frame. More specifically, the chance to transmit the mini-packet destined for destination node $(i+s)$ visits source node i every $D/c^{(i,s)}$ slots. Note that D and $c^{(i,s)}$ mean the number of slots of the super-frame and the number of slots assigned to path (i,s) during the super-frame, respectively.

Then, the mean packet delay for the case of non-uniform traffic load can be derived by modifying Eq. (17) as

$$W^{(i,s)} \approx \left[\frac{\lambda^{(i,s)} (D/c^{(i,s)})^2 E[{P_m}^2]}{2(1-\lambda^{(i,s)} (D/c^{(i,s)}) E[P_m])} + \frac{D}{c^{(i,s)}}\left(E[P_m] - \frac{1}{2}\right) + 1\right] \cdot t + W_p^{(i,s)}. \tag{18}$$

Table 1. Parameter Sets for the analysis of the packet delay time.

Parameter set	A	B	C	D	E	F	G	H	I
Traffic load matrix C	C_1			C_1			C_1		
The number of transmitters T	2			2			8		
The number of receivers R	2			2			8		
Compression rate K	8	16	32	16			64		
The super-frame length r	64	32	32	43	40	32	8		
Payload sizes S_p	53			53			53	256	530

5.3 Numerical Examples and Discussions

Noticing that our main purpose of this section is to investigate the effect of several system parameters on the packet delay, we simply assume that the distribution of the packet size, $f(x)$, follows the geometric function, i.e.,

$$f(x) = (1 - 1/P_B)^{x-1} \cdot 1/P_B. \tag{19}$$

In the following numerical examples, we will use the following values; the mean packet size P_B is set to be 500 [byte], the header size of the mini-packet S_h is 2 [byte], the LAN capacity, B_L, is fixed at 622 [Mbps], and the total length of the optical ring is 500 [Km]. For other parameters, we will use nine parameter sets shown in Table 1, which will be explained later in presenting figures.

We first investigate the effects of the compression rate, K, and the number of frames within the super-frame, r. For this, we use the parameter sets A, B and C shown in Table 1. For the super-frame length, we use the theoretical lower bound obtained in Section 3 to exclude the effect of the optimality level of the path accommodation method. The effect of the selection of the path accommodation method will be investigated in the below. While the parameter sets A ($K = 8$, $r = 64$) and B ($K = 16$, $r = 32$) give a same length of the super-frame ($D = 512$), Figure 15 clearly shows that the larger compression rate in the parameter set B leads to the smaller delay. Then, the maximum throughput becomes much larger. However, the larger compression rate does not always attain improved performance. It is illustrated by the parameter set C in which the compression rate of $K = 32$, twice of the one in the parameter set B, is used. The results of the parameter sets B and C are very close. That is, the number of frames r has a great impact on performance in this parameter region.

An example of the effect of path accommodation methods is shown by using parameter sets D, E and F in Table 1). By applying the Algorithms A1, A2 and A3, we obtained 43, 40 and 32 as the number of frames in those parameter sets. Figure 16 clearly shows that the number of frames much affects the mean packet delay time.

The effect of the mini-packet size is finally presented in Figure 17. Here, we set the mini-packet size, S_i, to be 53, 256 and 530 [byte] while the mean packet length remains unchanged. The different mini-packet sizes are considered in the parameter sets G, H and I of Table 1. The differences observed in Figure 17 are much affected by the mean and distribution of the packet length in the current case. Namely, we can observe that padding necessary for the last mini-packet cannot be ignored in obtaining high performance when the slot length is fixed as in the current case of the OCTDM ring. We need a further research work to determine the appropriate slot length by taking account of the actual packet size distribution.

6. CONCLUDING REMARKS AND FUTURE WORKS

In this paper, we have proposed and evaluated the path accommodation methods for the bidirectional OCTDM rings, which are expected as the new generation all-optical networks.

We have first derived the theoretical lower bound for the length of the super-frame, in which all paths among nodes are perfectly allocated. The relationships between the lower bound and such parameters as compression rate, transmitters and receivers are then investigated. Three path accommodation algorithms are proposed next to treat the non-uniform traffic load. Through numerical examples, we have shown that the result obtained by our proposed algorithms is close to the lower bound. The mean packet delay time has also been analyzed.

As future works, the reliability issue for the OCTDM rings should be addressed, which is an important feature of optical networks. Also, the optical compression TDM/WDM where the optical compression is applied to each of wavelengths in WDM must be an interesting research topic.

REFERENCES

[1] *Proceeding of the SPIE Conference on All-Optical Networking 1999: Architecture, Control, and Management Issues*, Sept. 1999.

[2] Special issue on “WDM fiber optic communications”, *IEEE Communications Magazine*, vol. 36, no. 12, Dec. 1998.

[3] A. Hasegawa and H. Toda, “A feasible all optical soliton based inter-LAN link using time division multiplexing,” *IEICE Transactions on Communications*, vol. E81-B, pp. 1681–1686, August 1998.

[4] N. S. Patel, K. L. Hall, and K. A. Rauschenbach, “Optical rate conversion for high-speed TDM networks,” *IEEE Photonics Technology Letters*, vol. 9, no. 9, pp. 1277–1279, September 1997.

[5] B. Y. Yu, P. Toliver, R. J. Runser, K. L. Deng, D. Zhou, I. Glesk, and P. R. Prucnal, “Packet-switched optical networks,” *IEEE Micro*, vol. 18, no. 1, pp. 28–38, January-February 1998.

[6] X. Zhang and C. Qiao, "On optimal scheduling and cost effective design in WDM rings," *IEEE/LEOS Broadband Optical Networks*, Paper TuB3, Aug. 1996.

[7] X. Zhang and C. Qiao, "On scheduling all-to-all connections and cost-effective designs in WDM rings," *IEEE/ACM Transactions on Networking*, vol. 7, no. 3, pp. 435–445, June 1999.

[8] X. Zhang and C. Qiao, "An effective and comprehensive approach for traffic grooming and wavelength assignment in SONET/WDM rings," *Proc. SPIE All-Optical Networking*, vol. 3531, pp. 221–232, November 1998.

[9] K. L. Deng, K. I. Kang, I. Glesk, P. R. Prucnal, and S. Shin, "Optical packet compressor for ultra-fast packet-switched optical networks," *Electronics Letters*, vol. 33, no. 14, pp. 1237–1239, July 1997.

[10] H. Toda, F. Nakada, M. Suzuki, and A. Hasegawa, "An optical packet compressor using a fiber loop for a feasible all optical TDM network," in *Proceedings of 25th European Conference on Optical Communication (ECOC'99)*, vol. Tu C3.7, September 1999.

[11] A. Hasegawa and H. Toda, "An optical packet compressor for a feasible all optical inter-LAN TDM network," in *Proceedings of Broadband Access and Technology, European Conference on Networks and Optical Communications (NOC'99)* (D. W. Faulkner and A. L. Harmer, eds.), pp. 233–238, IOS press, June 1999.

[12] K. L. Hall and K. A. Rauschenbach, "All-optical buffering of 40-Gb/s data packets," *IEEE Photonics Technology Letters*, vol. 10, no. 3, pp. 442–444, Mar. 1998.

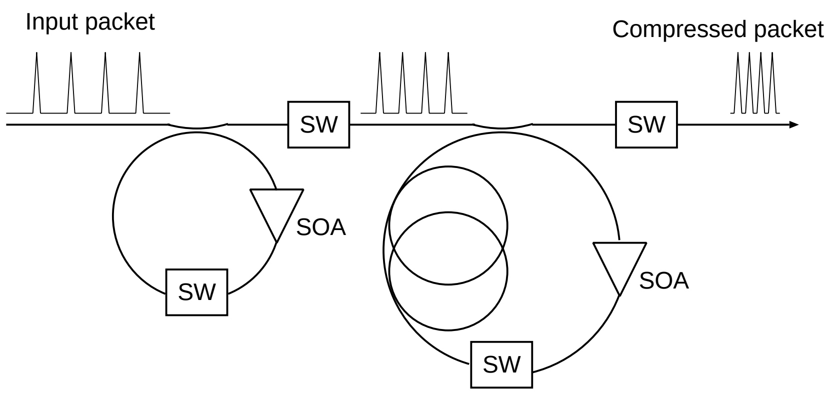


Fig. 1. Optical packet compressor

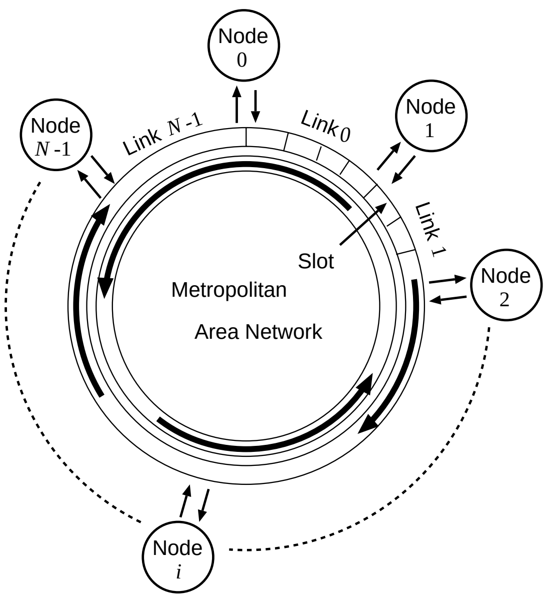


Fig. 2. A bidirectional ring with optical compression TDM

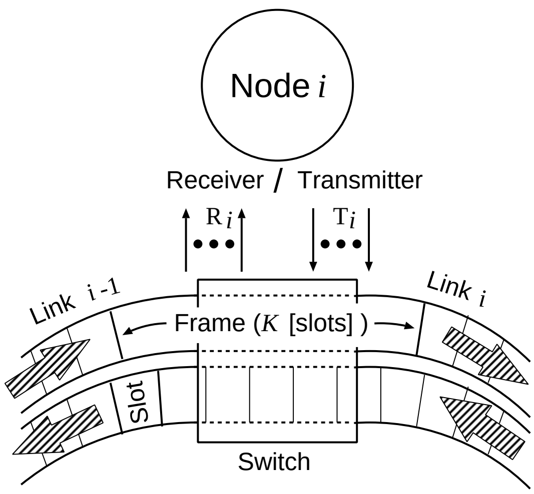
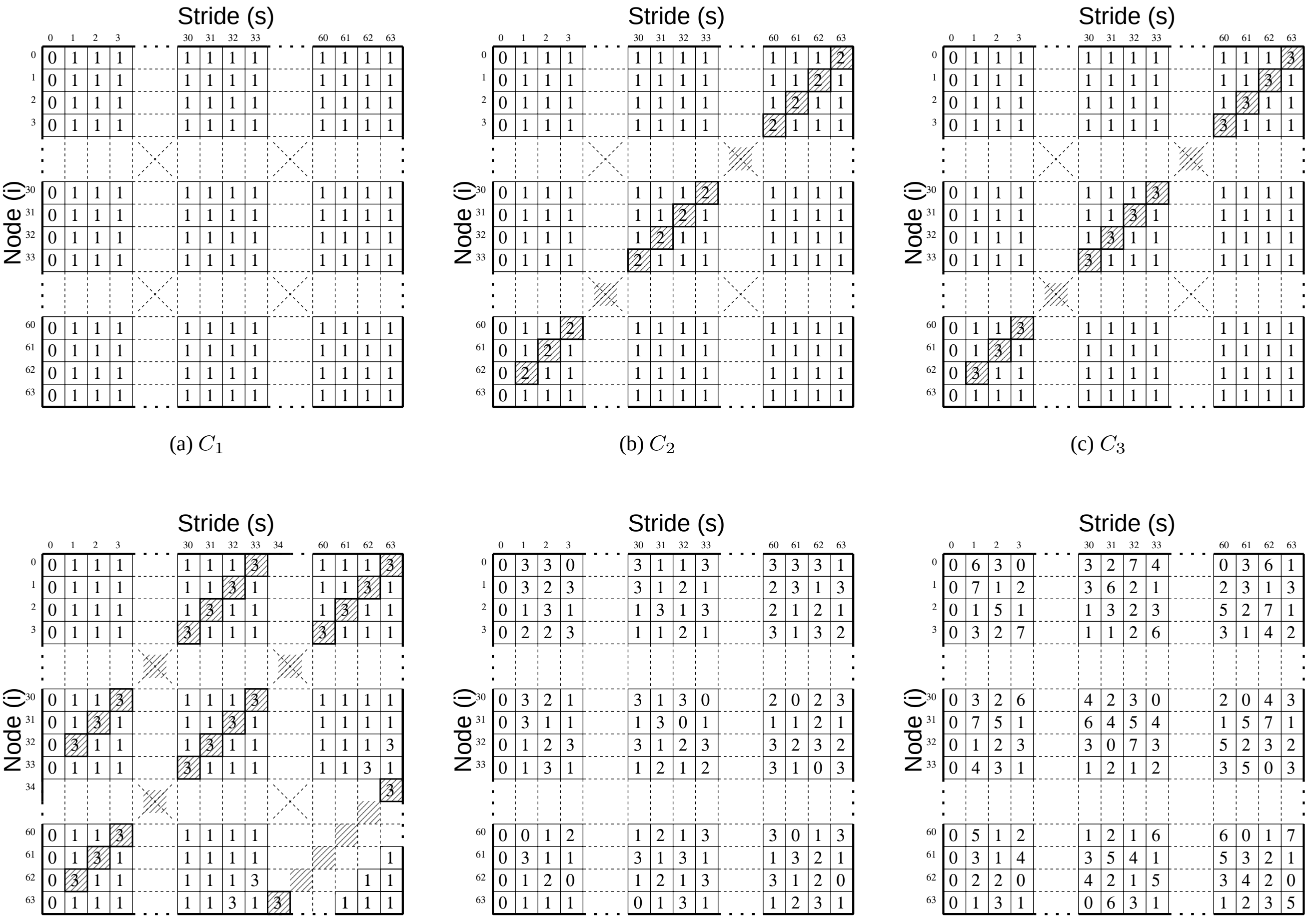


Fig. 3. Structure of node i



(a) C_1 (b) C_2 (c) C_3

(d) C_4 (e) C_5 (f) C_6

Fig. 4. Traffic matrices for numerical examples

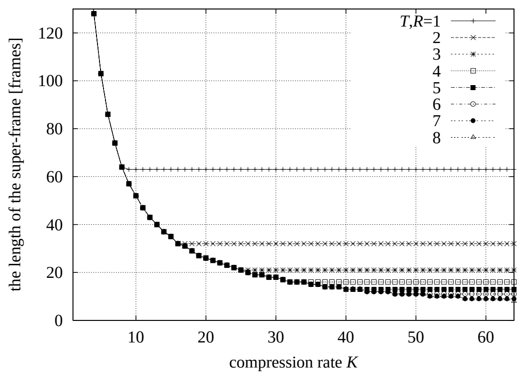


Fig. 5. $LB(64, T, R, K, C_1)$

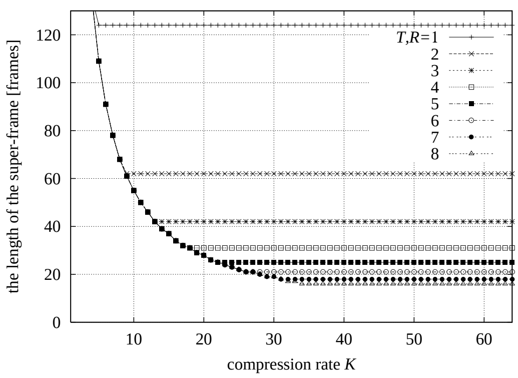


Fig. 6. $LB(64, T, R, K, C_2)$

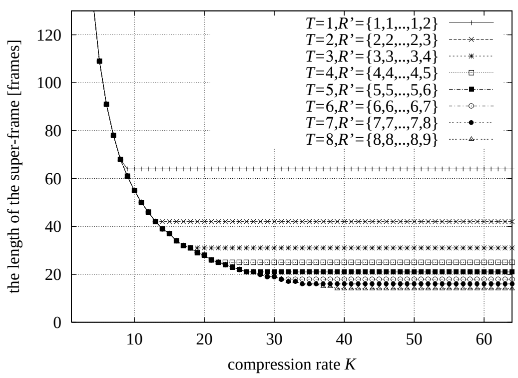


Fig. 7. $LB(64, T, R', K, C_2)$

	$T=R=1$				$T=R=2$			
K	LB	A1	A2	A3	LB	A1	A2	A3
1	512	527	530	**512**	512	525	526	**512**
2	256	266	267	**257**	256	264	264	**256**
4	128	139	138	**129**	128	132	132	**128**
8	64	85	81	**69**	64	70	68	**64**
16	63	**67**	**67**	69	32	43	40	**32**
32	63	67	**66**	69	32	33	33	**32**
64	63	**63**	**63**	69	32	**32**	34	**32**
	$T=R=4$				$T=R=8$			
K	LB	A1	A2	A3	LB	A1	A2	A3
1	512	525	527	**512**	512	525	526	**512**
2	256	263	265	**256**	256	263	263	**256**
4	128	132	132	**128**	128	132	132	**128**
8	64	66	67	**64**	64	66	66	**64**
16	32	34	34	**32**	32	33	33	**32**
32	16	20	20	**16**	16	17	17	**16**
64	16	17	17	**16**	8	10	10	**8**

(a) The case of traffic matrix C_1

	$T=R=1$				$T=R=2$			
K	LB	A1	A2	A3	LB	A1	A2	A3
1	544	551	**546**	547	544	545	**544**	**544**
2	272	291	**276**	285	272	276	**273**	**273**
4	136	170	160	**149**	136	146	140	**138**
8	124	129	**126**	127	68	86	80	**70**
16	124	**126**	**126**	**126**	62	64	**63**	64
32	124	**126**	**126**	**126**	62	**63**	**63**	**63**
64	124	**126**	**126**	**126**	62	**63**	**63**	**63**
	$T=R=4$				$T=R=8$			
K	LB	A1	A2	A3	LB	A1	A2	A3
1	544	545	**544**	**544**	544	545	545	**544**
2	272	**272**	273	**272**	272	**272**	273	**272**
4	136	138	137	**136**	136	137	137	**136**
8	68	72	70	**68**	68	69	69	**68**
16	34	43	39	**35**	34	36	35	**34**
32	31	**32**	**32**	**32**	17	22	19	**18**
64	31	**32**	**32**	**32**	16	**16**	**16**	**16**

(b) The case of traffic matrix C_2

	$T=R=1$				$T=R=2$			
K	LB	A1	A2	A3	LB	A1	A2	A3
1	576	589	**577**	581	576	**576**	**576**	**576**
2	288	322	**299**	300	288	294	**288**	289
4	186	215	**192**	193	144	162	150	**146**
8	186	191	**189**	190	93	109	**97**	**97**
16	186	**189**	**189**	**189**	93	96	**95**	**95**
32	186	**189**	**189**	**189**	93	**95**	**95**	**95**
64	186	**189**	**189**	**189**	93	**95**	**95**	**95**
	$T=R=4$				$T=R=8$			
K	LB	A1	A2	A3	LB	A1	A2	A3
1	576	**576**	**576**	**576**	576	**576**	**576**	**576**
2	288	**288**	**288**	**288**	288	**288**	**288**	**288**
4	144	147	**145**	**145**	144	**144**	**144**	**144**
8	72	81	75	**73**	72	74	73	**72**
16	47	54	**48**	49	36	41	38	**37**
32	47	**48**	**48**	**48**	24	27	**24**	25
64	47	**48**	**48**	**48**	24	**24**	**24**	**24**

(c) The case of traffic matrix C_3

	$T=R=1$				$T=R=2$			
K	LB	A1	A2	A3	LB	A1	A2	A3
1	578	594	**586**	598	578	**578**	580	590
2	289	326	**309**	315	289	298	**291**	301
4	186	215	199	**194**	145	163	155	**150**
8	186	191	**189**	190	93	107	99	**97**
16	186	**189**	**189**	**189**	93	96	**95**	**95**
32	186	**189**	**189**	**189**	93	**95**	**95**	**95**
64	186	**189**	**189**	**189**	93	**95**	**95**	**95**
	$T=R=4$				$T=R=8$			
K	LB	A1	A2	A3	LB	A1	A2	A3
1	578	**578**	581	590	578	**578**	580	590
2	289	**290**	291	301	289	**290**	291	301
4	145	147	**146**	148	145	**145**	**145**	148
8	73	80	77	**74**	73	74	**73**	74
16	47	54	**49**	**49**	37	41	39	**38**
32	47	**48**	**48**	**48**	24	27	**24**	25
64	47	**48**	**48**	**48**	24	**24**	**24**	**24**

(d) The case of traffic matrix C_4

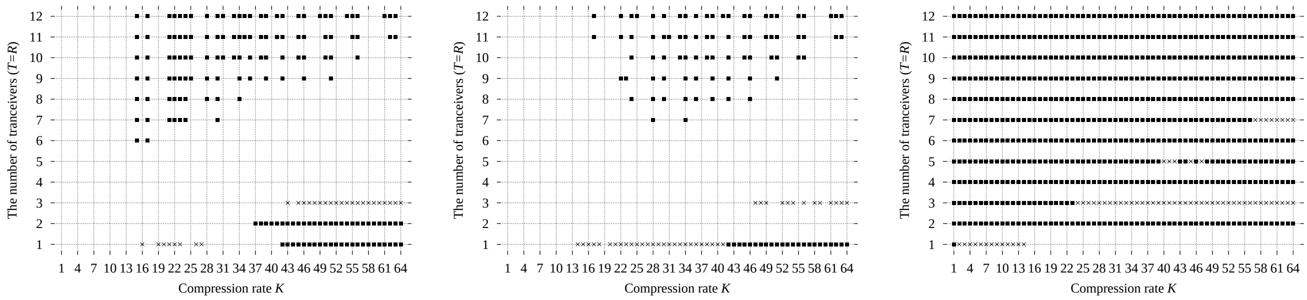
	$T=R=1$				$T=R=2$			
K	LB	A1	A2	A3	LB	A1	A2	A3
1	823	843	**834**	918	823	841	**830**	916
2	412	423	**418**	461	412	421	**416**	457
4	206	221	**215**	237	206	212	**208**	229
8	115	134	135	**130**	103	108	**107**	116
16	115	116	116	**115**	58	69	66	**64**
32	115	**115**	**115**	**115**	58	**58**	**58**	**58**
64	115	**115**	**115**	**115**	58	**58**	**58**	**58**
	$T=R=4$				$T=R=8$			
K	LB	A1	A2	A3	LB	A1	A2	A3
1	823	841	**833**	916	823	841	**830**	916
2	412	421	**415**	457	412	421	**415**	457
4	206	211	**208**	229	206	211	**208**	229
8	103	105	**104**	115	103	105	**104**	115
16	52	**54**	**54**	58	52	53	**52**	58
32	29	33	34	**31**	26	**27**	**27**	29
64	29	**29**	**29**	**29**	15	17	17	**16**

(e) The case of traffic matrix C_5

	$T=R=1$				$T=R=2$			
K	LB	A1	A2	A3	LB	A1	A2	A3
1	1936	1979	**1940**	2063	1936	1972	**1940**	2057
2	968	996	**971**	1039	968	989	**970**	1030
4	484	512	**496**	531	484	496	**486**	517
8	277	322	314	**291**	242	250	**249**	261
16	277	281	**279**	**279**	139	156	158	**142**
32	277	**279**	**279**	**279**	139	**140**	**140**	**140**
64	277	**279**	**279**	**279**	139	**140**	**140**	**140**
	$T=R=4$				$T=R=8$			
K	LB	A1	A2	A3	LB	A1	A2	A3
1	1936	1972	**1940**	2057	1936	1972	**1941**	2057
2	968	989	**970**	1027	968	989	**971**	1027
4	484	493	**485**	516	484	493	**485**	516
8	242	246	**243**	258	242	246	**243**	258
16	121	126	**124**	130	121	123	**122**	129
32	70	76	77	**70**	61	**62**	63	65
64	70	**70**	**70**	**70**	35	37	39	**36**

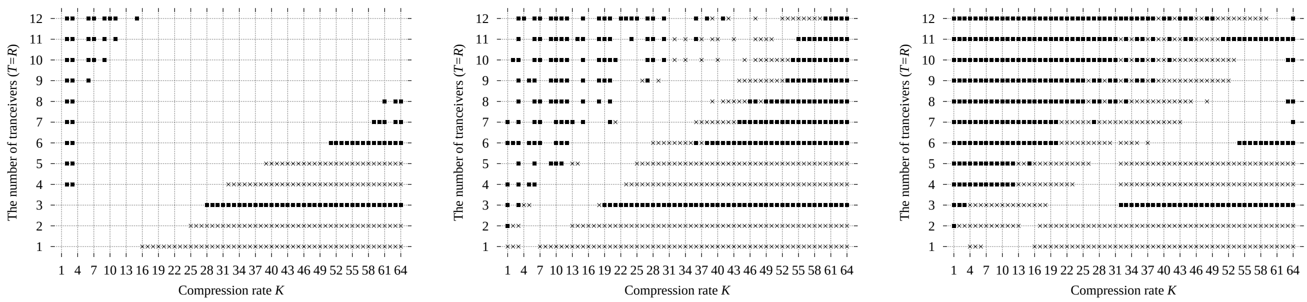
(f) The case of traffic matrix C_6

Fig. 8. Comparisons of lower bounds and the super-frame lengths obtained by three algorithms



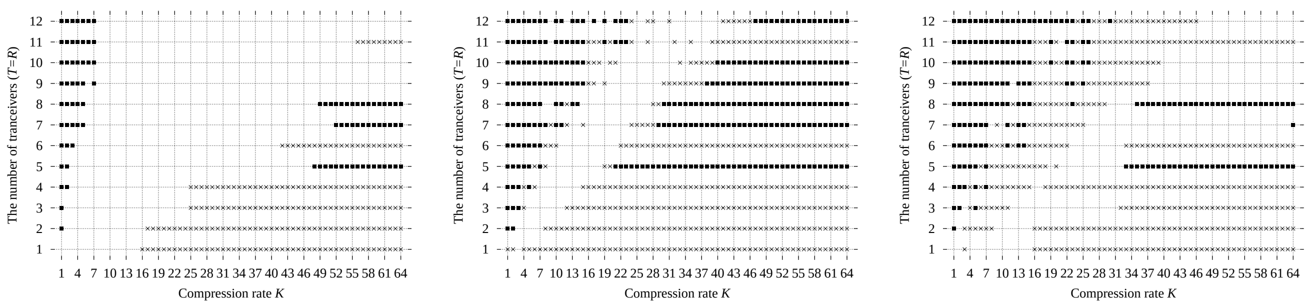
(a) Algorithm A1 for C_1 (b) Algorithm A2 for C_1 (c) Algorithm A3 for C_1

Fig. 9. Precise comparisons of results obtained by three algorithms for the traffic matrix C_1



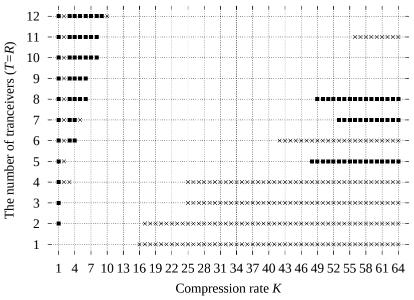
(a) Algorithm A1 for C_2 (b) Algorithm A2 for C_2 (c) Algorithm A3 for C_2

Fig. 10. Precise comparisons of results obtained by three algorithms for the traffic matrix C_2

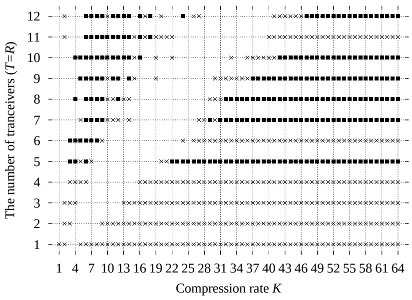


(a) Algorithm A1 for C_3 (b) Algorithm A2 for C_3 (c) Algorithm A3 for C_3

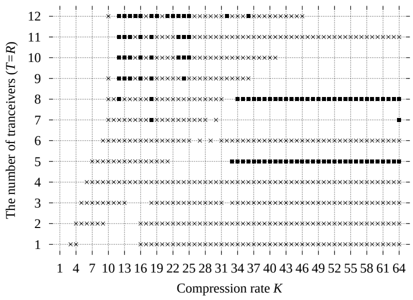
Fig. 11. Precise comparisons of results obtained by three algorithms for the traffic matrix C_3



(a) Algorithm A1 for C_4

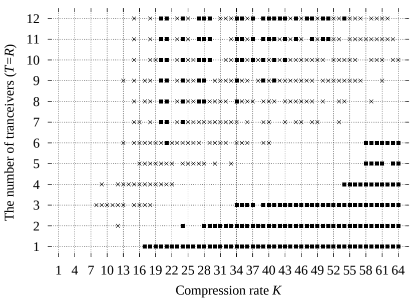


(b) Algorithm A2 for C_4

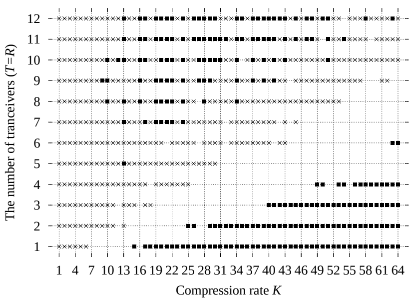


(c) Algorithm A3 for C_4

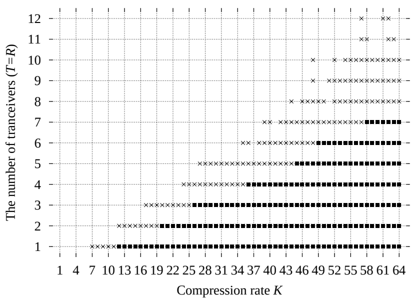
Fig. 12. Precise comparisons of results obtained by three algorithms for the traffic matrix C_4



(a) Algorithm A1 for C_5

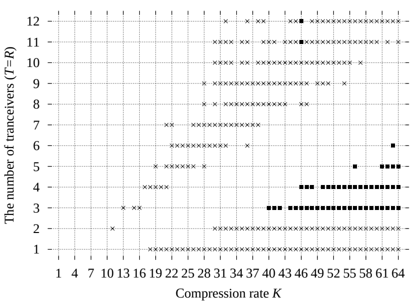


(b) Algorithm A2 for C_5

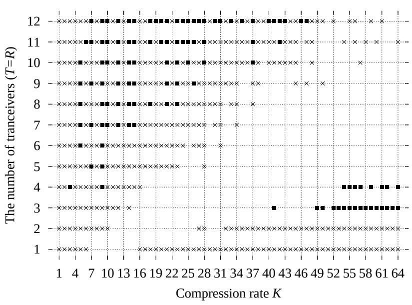


(c) Algorithm A3 for C_5

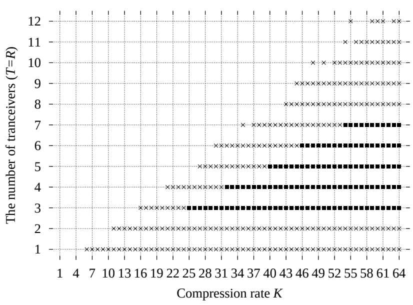
Fig. 13. Precise comparisons of results obtained by three algorithms for the traffic matrix C_5



(a) Algorithm A1 for C_6



(b) Algorithm A2 for C_6



(c) Algorithm A3 for C_6

Fig. 14. Precise comparisons of results obtained by three algorithms for the traffic matrix C_6

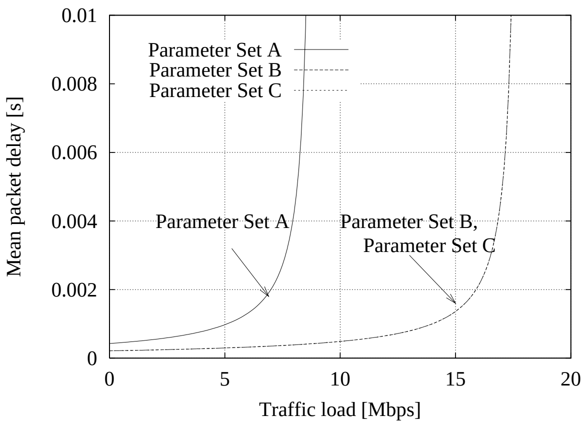


Fig. 15. The packet delay times for parameter sets A, B and C

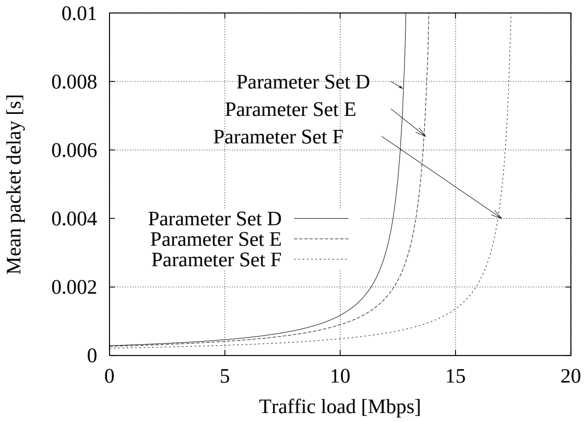


Fig. 16. The packet delay times for parameter sets D, E and F

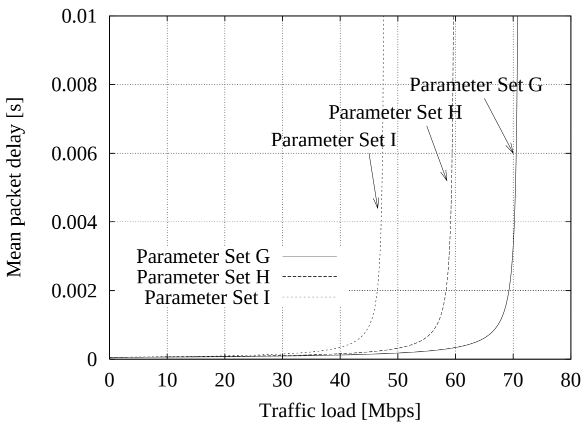


Fig. 17. The packet delay time for parameter sets G, H and I