PAPER

FEC-Based Reliable Transmission for Multiple Bursts in OBS Networks

Satoshi ARIMA[†], *Nonmember*, Takuji TACHIBANA[††a)], Yuichi KAJI[††b)], *and* Shoji KASAHARA[†††c)], *Members*

SUMMARY In this paper, we consider consecutive burst transmission with burst loss recovery based on Forward Error Correction (FEC) in which redundant data is transmitted with multiple bursts. We propose two burst generation methods: Out-of Burst Generation (OBG) and In-Burst Generation (IBG). The OBG generates a redundant burst from redundant data, while the IBG reconstructs a burst from an original data block and a part of the redundant data. For both methods, the resulting bursts are transmitted consecutively. If some bursts among the bursts are lost at an intermediate node, the lost bursts can be recovered with the redundant data using FEC processing at the destination node. We evaluate by simulation the proposed methods in a uni-directional ring network and NSFNET, and compare the performances of the proposed methods with the extra-offset time method. Numerical examples show that the proposed methods can provide a more reliable transmission than the extra-offset time method for the OBS network where the maximum number of hops is large. Moreover, it is shown that the end-to-end transmission delay for our proposed methods can be decreased by enhancing the FEC processor or by increasing the number of FEC processors.

key words: *optical burst switching, multiple bursts transmission, FEC, extra offset, burst failure rate, burst loss recovery*

1. Introduction

Optical burst switching (OBS) has received considerable attention as one of the most promising technologies for the next-generation Internet over wavelength division multiplexing (WDM) networks [1]–[3]. Currently, several research projects for the OBS network are on going [4], [5]. In Jumpstart project, the implementation of Just-In-Time (JIT) protocol which is a simple hardware-based signaling protocol has been performed [4]. In Japan, the OBS network testbed has been developed [5], and the optical code (OC)-based one-way protocol is used as a signaling protocol for the project. In the near feature, it is expected that the OBS networks will be world-widely deployed.

Various applications such as Voice over IP (VoIP) and Video on Demand (VoD) services are expected over the OBS network [6]–[8]. Therefore, the reliable transmission methods such as extra-offset time method [3], [9], [10], segmentation method [11], and preemptive method [12], [13] have been proposed.

Grid computing is also considered as desired applications over OBS networks [14], [15]. In the Grid over OBS, each Grid job is encapsulated into a burst, and then it is transmitted to a remote site with its corresponding control packet containing requirements such as deadline and necessary storage. The Grid over OBS can be utilized more effectively than lightpath-based optical grid if one of the following conditions holds; (1) the job size is small (on the order of a few megabytes), (2) the job-arrival time is highly unpredictable, and (3) the location of job submissions is highly unpredictable [16]–[18].

One of the important requirements for Grid computing is simultaneous data transmission in which multiple Grid jobs are simultaneously transmitted to a remote processing site. In order to provide the simultaneous data transmission in the OBS network, multiple bursts should be transmitted to a destination node. In addition, the reliable transmission of the multiple bursts is also indispensable in order to transmit the multiple grid jobs to the remote sites preferentially.

In the conventional OBS, however, an optical burst is generated from multiple IP packets with the same OBS destination, and it is not necessary to transmit multiple bursts to a destination node simultaneously. The conventional service differentiation schemes such as extra-offset time method [3], [9], [10], segmentation method [11], and preemptive method [12], [13] don't take into account the reliable transmission for the simultaneous multiple bursts.

In this paper, in order to transmit multiple bursts simultaneously and reliably, we propose two burst transmission methods based on forward error correction (FEC); Out-of-Burst Generation (OBG) [19] and In-Burst Generation (IBG). In both the methods, redundant data is generated from multiple original bursts with FEC encoding. Then, the original bursts are transmitted to a destination node along with the redundant data. When some bursts among the transmitted bursts are lost at some intermediate node, the lost bursts can be recovered with FEC decoding. The FEC encoding and decoding are performed only at source and destination, respectively, and the proposed methods do not require any improvement of OBS core nodes.

Here, the two proposed methods are different in how the redundant data is included into bursts. Suppose the number of original data blocks is β. Note that in Grid computing, a data block corresponds to a Grid job. In the OBG, a new burst containing the whole redundant data is generated as

Manuscript received October 30, 2006.

Manuscript revised May 1, 2007.

[†] The author is with Kyocera Corporation, Kyoto-shi, 612-8501 Japan.

[††] The authors are with Nara Institute of Science and Technology, Ikoma-shi, 630-0192 Japan.

[†††] The author is with Kyoto University, Kyoto-shi, 606-8503 Japan.

a) E-mail: takuji-t@is.naist.jp

b) E-mail: kaji@is.naist.jp

c) E-mail: shoji@i.kyoto-u.ac.jp

DOI: 10.1093/ietcom/e90–b.12.3541

the $\beta + 1$st burst [19]. In the IBG, on the other hand, each burst is generated with an original data block and a part of its redundant data, and hence the number of bursts to be transmitted is equal to that of original data blocks.

We evaluate by simulation the performances of the proposed methods in a unidirectional ring network and NSFNET. Two performance measures are considered: the failure rate and the mean FEC processing time. (In [19], we considered the worst-case performance of OBG by the maximum FEC processing time instead of the mean FEC processing time.) The failure rate is defined as the ratio of the number of failure requests to the number of overall transmission requests. In numerical examples, we show the effectiveness of the proposed methods by comparing the FEC processing time and extra offset time.

This paper is organized as follows. In Sect. 2, we describe the consecutive multiple-bursts transmission. Then, we explain the OBG and IBG methods in Sect. 3 and Sect. 4, respectively. Numerical examples are presented in Sect. 5, and finally, conclusions are presented in Sect. 6.

2. Consecutive Multiple-Bursts Transmission

In this section, we consider how multiple bursts constructing a large amount of data should be transmitted to its destination node in the OBS network.

Figure 1 shows an example of the grid computing in the OBS network. In the grid computing, a grid job is encapsulated into a burst and the burst is transmitted to its remote processing site. As shown in Fig. 1, when multiple grid jobs are processed at a remote processing site, multiple bursts are transmitted to the processing site. In this case, the multiple bursts should be transmitted simultaneously to their OBS egress-edge node.

One of the easiest ways for transmitting multiple bursts simultaneously is to transmit multiple bursts at the same time, using the same number of wavelengths as that of original bursts. In general, however, the number of wavelengths multiplexed into an optical fiber is not large, and hence this transmission method is not feasible.

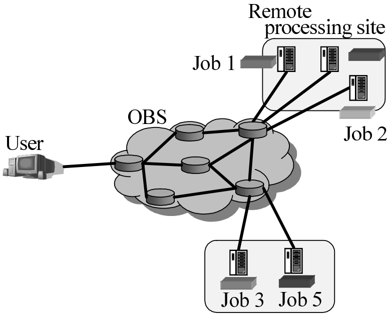


Fig. 1 The grid computing over the OBS network.

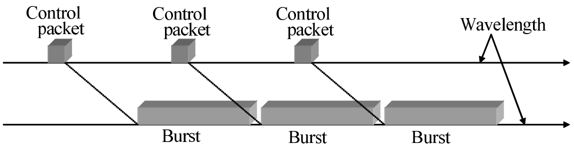


Fig. 2 The consecutive burst transmission.

The alternative method which both multiple bursts transmission and the efficient use of wavelength resource are achieved is the consecutive burst transmission. Figure 2 shows the consecutive burst transmission at a link. In the consecutive burst transmission, each burst is transmitted independently, that is, the control packet of each burst reserves a wavelength independently. This prevents a wavelength from being reserved by the multiple bursts for a long time. However, the drawback of this independent wavelength reservation is that some wavelength reservations for the bursts are likely to fail. In order to overcome the drawback, we apply FEC-based burst loss recovery to the consecutive burst transmission.

3. Out-of Burst Generation (OBG)

In this paper, we consider to use the Reed-Solomon (RS) codes for the FEC processing, but essentially any FEC codes are available in the proposed framework. The (n, k) RS code consists of n-byte length codewords, with k out of n-byte is the original data and the remaining $(n-k)$-byte is the *redundant data*. Note that the unit *byte* is used to denote one symbol of the code. In the case we choose $n = 2^8 - 1$, one symbol is represented in eight bits[†]. The (n, k) RS code can correct t errors and e erasures which occurred in one n-byte codeword if

$$2t + e \leq n - k. \tag{1}$$

If no error occurs, then the code can recover up to $(n-k)$-byte data loss [20]. We also note that, for RS codes, the parameters n and k can be chosen flexibly by puncturing some symbols in the code.

3.1 Burst Assembly Mechanism Based on FEC

Now, we consider the case where an application data is fragmented into β blocks with the same size. A burst is generated from each data block, and β bursts are transmitted to their destination node consecutively. For the simplicity of the explanation, we assume that the size of each data block is D bytes. The parameters of the RS code are chosen so that $n = (\beta + 1)\theta_O$ and $k = \beta\theta_O$ with θ_O a constant. The RS code is capable of recovering θ_O-byte data if the other $\beta\theta_O$-byte data is provided. By using the RS code, we construct one new burst which consists of redundant data of the other β bursts. Even if one of the $\beta + 1$ bursts is lost during the transmission, it is recoverable as far as the remaining β bursts are successfully delivered to the destination node.

Figure 3 shows the β original data blocks at a source node. In the OBG method, FEC encoding with $((\beta +$

[†] It is assumed in this paper that the RS code is defined over $GF(2^8)$ and one symbol of the RS code is represented in eight bits.

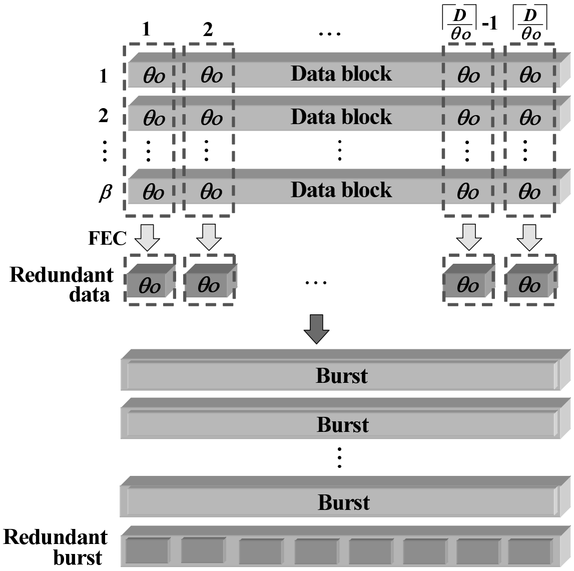

1
2
…
$\lceil \frac{D}{\theta_O} \rceil$-1
$\lceil \frac{D}{\theta_O} \rceil$
1
2
β
θo
Data block
FEC
Redundant data
Burst
Redundant burst

Fig. 3 Out-of burst generation (OBG).

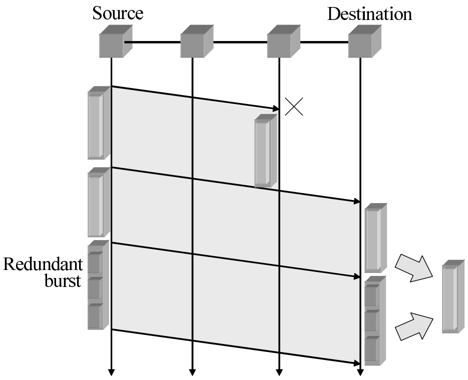


Fig. 4 Burst recovery using OBG for the consecutive multiple-bursts transmission.

$1)\theta_O, \beta\theta_O)$ RS code is performed at the source node. We collect θ_O-byte data blocks from each of the β bursts, and compute a θ_O-byte redundant data from the collected $\beta\theta_O$-byte data. In Fig. 3, this FEC encoding is illustrated using dot line. The redundant blocks are concatenated and one new burst is constructed. Because each data burst consists of D-bytes, we need to perform $\lceil D/\theta_O \rceil$ encoding operations, and each encoding operation generates θ_O redundant data, where $\lceil x \rceil$ is the smallest integer that is greater than or equal to x. Therefore the size of the redundant burst is $\lceil D/\theta_O \rceil \theta_O$-byte. These FEC encoding can be performed in parallel if there are $N_{FEC} = \lceil D/\theta_O \rceil$ FEC processors in the source node.

The β original bursts and the generated redundant burst are transmitted to their destination node. If one of the $\beta + 1$ bursts is lost at some intermediate node and the remainders are eventually transmitted to the destination node, the lost burst can be recovered at the destination node using FEC decoding according to (1). Note that in (1), $t = 0$, $e = D$, $n = (\beta + 1)D$, and $k = \beta D$.

Figure 4 shows the case where $\beta = 2$ original bursts and a redundant burst are transmitted from their source to destination. In this figure, the first original burst is lost at an intermediate node, and other bursts are eventually transmitted to the destination node. In this case, the lost original burst is recovered from the other original burst and redundant one. As a result, the transmission of the two original bursts succeeds.

3.2 FEC Processing Time

In the OBG method, FEC encoding and decoding are performed at the source and destination, respectively. Because the FEC decoding time is larger than the FEC encoding time, we focus on the FEC decoding time in the following.

Here, we consider the case where a redundant burst is generated from β original bursts whose sizes are D bytes as shown in Fig. 3. In this case, $(\beta + 1)D$-byte data is processed for the FEC decoding. Suppose that the number of FEC processors is N_{FEC}, and that the FEC processing speed of each FEC processor is L bps. Then, the FEC decoding time, T_{OBG} [s], is given by

$$T_{OBG} = \frac{8(\beta + 1)D}{N_{FEC} L}. \tag{2}$$

The FEC decoding process is performed only when a burst loss is recovered for a multiple bursts transmission. Let $T_{FEC}^{(OBG)}$ denote the random variable such that

$$T_{FEC}^{(OBG)} = \begin{cases} 0, & \text{if no burst loss occurs during} \\ & \text{the multiple bursts transmission,} \\ T_{OBG}, & \text{if the multiple bursts} \\ & \text{transmission succeeds with} \\ & \text{burstloss recovery.} \end{cases} \tag{3}$$

Then the mean FEC decoding time $E[T_{FEC}^{(OBG)}]$ is given by

$$\begin{aligned} E[T_{FEC}^{(OBG)}] &= T_{OBG} \\ &\times \Pr\{\text{one of } \beta + 1 \text{ bursts is lost} \mid \text{the transmission} \\ &\quad \text{of } \beta \text{ data blocks succeeds}\}. \end{aligned} \tag{4}$$

If burst losses rarely occur, the conditional probability in (4) is small, and even a FEC processor with a large decoding time is available.

4. In-Burst Generation (IBG)

In the OBG method, a redundant burst is generated from multiple original bursts using FEC encoding. This increases the number of bursts to be transmitted and hence the transmission overhead becomes large. Furthermore, more number of bursts we may transmit, we surely have more risk that one or more bursts being lost. In this section, we consider the alternative method where the number of bursts to be transmitted does not change.

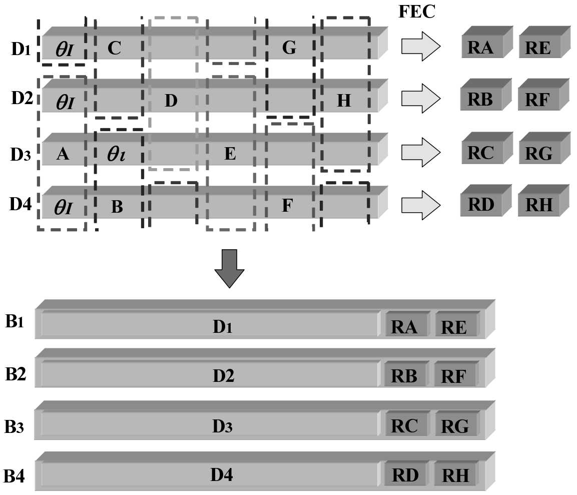


Fig. 5 In-burst generation (IBG).

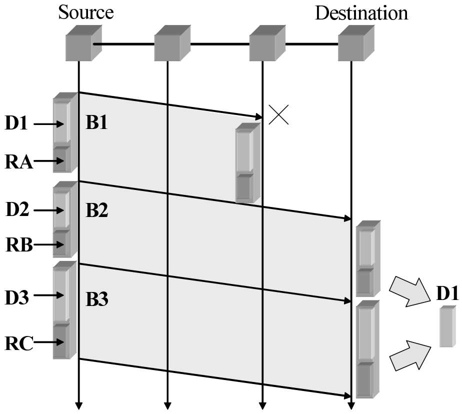


Fig. 6 Burst recovery using IBG for the consecutive multiple-bursts transmission.

4.1 Burst Assembly Mechanism Based on FEC

Similarly to Sect. 3, we consider the case where an application data is fragmented into β blocks. We assume that the size of each data block is D bytes.

The basic idea of the IBG is that the RS code is used to encode data from $\beta - 1$ bursts, and to append the computed redundant data to the remaining one burst. Let the parameters of the RS code be $n = \beta\theta_I$ and $k = (\beta - 1)\theta_I$ with θ_I a constant. The RS code generates θ_I-byte redundant data from $(\beta - 1)\theta_I$-byte data in $\beta - 1$ bursts.

Figure 5 shows the case of $\beta = 4$. Here, D_i denotes the ith data block and B_i the resulting burst corresponding to D_i. The essential point is that the encoding operation is performed to data from three out of four bursts. For example, RA is the redundant data obtained by encoding the first θ_I-byte blocks of D_2, D_3, and D_4 (the blocks denoted as "A" in the figure). RA is appended to the burst B_1 which did not offer a data block for the encoding. Another encoding is then performed for the first θ_I-byte blocks of D_1 and the second θ_I-byte blocks of D_3 and D_4 (the blocks "B" in the figure), and the computed redundant data is appended to D_2. In this way, each burst is appended with redundant data which are computed from data in the other bursts.

The most notable point here is that, the redundant data which is computed from a certain burst is not appended to that burst. See D_1 in Fig. 5 for example. Data blocks in the burst D_1 are used to compute RB, RC, RD, RF, RG and RH, and these redundant data blocks are not appended to D_1. Even if the burst B_1 is lost during the transmission, the data blocks in D_1 are recoverable because redundant data are kept safely in the other bursts. Remind that the RS code is capable of recovering θ_I-byte data if the other $(\beta - 1)\theta_I$-byte data is provided.

Here, the total number of FEC encodings is given by $\lceil \beta D/(\beta - 1)\theta_I \rceil$ where $\lceil x \rceil$ is the smallest integer that is greater than or equal to x. These FEC encodings can be performed simultaneously if there are $N_{FEC} = \lceil \beta D/(\beta - 1)\theta_I \rceil$ FEC processors in the source node.

The resulting number of generated burst is β, the same as the number of original data blocks. However, note that the size of the burst is larger than that of the original data block due to the redundant data. When $(\beta\theta_I, (\beta - 1)\theta_I)$ RS code is used, the resulting burst size is $\beta D/(\beta - 1)$ [bytes].

The β bursts are transmitted to their destination node. If one of the β bursts is lost at some intermediate node and the remainders are eventually transmitted to the destination node, the lost burst can be recovered at the destination node using FEC decoding according to (1). Note that in (1), $t = 0$, $e = D$, $n = \beta D$, and $k = (\beta - 1)D$.

Figure 6 illustrates the transmission of bursts B_1, B_2, and B_3. These are generated from original data blocks D_1, D_2, and D_3, and redundant data blocks RA, RB, and RC, respectively. In Fig. 6, B_1 is lost, however, the original data block D_1 is recovered using other two bursts. As a result, the transmission of the three original data blocks succeeds. Thus, the IBG method can recover the lost burst without increasing the number of bursts to be transmitted.

4.2 FEC Processing Time

In the IBG method, FEC encoding and decoding are performed at source and destination nodes, respectively, as well as the OBG. We also focus on the FEC decoding time.

Here, we consider the case as shown in Fig. 5. Because the size of a burst is $\beta D/(\beta - 1)$ from the previous subsection, $\beta^2 D/(\beta - 1)$-byte data is processed for the FEC decoding. Suppose that the number of FEC processors is N_{FEC}, and that the FEC processing speed of each FEC processor is L bps. Then, the FEC decoding time, T_{IBG} [s], is given by

$$T_{IBG} = \frac{8\beta^2 D}{(\beta - 1)N_{FEC}L}. \tag{5}$$

From Eqs. (2) and (5), we have

$$T_{IBG} - T_{OBG} = \frac{8D}{(\beta - 1)N_{FEC}L} \geq 0, \tag{6}$$

i.e., the FEC decoding time of the IBG method is larger than that of the OBG method. This is because the size of redundant data in the IBG method is larger than that in the OBG method.

As shown in Sect. 3.2, the FEC decoding process is performed only when a burst loss is recovered for a multiple bursts transmission. Therefore, the mean FEC decoding time $E[T_{FEC}^{(IBG)}]$ is given by

$$\begin{aligned} E[T_{FEC}^{(IBG)}] &= T_{IBG} \\ &\quad \times \Pr\{\text{one of } \beta \text{ bursts is lost} \mid \text{the transmission} \\ &\qquad \text{of } \beta \text{ data blocks succeeds}\}. \end{aligned} \tag{7}$$

5. Numerical Examples

In this section, we investigate the performances of the OBG and IBG methods in a unidirectional ring network and NSFNET.

5.1 Ring Network

In the ring network, we assume that the number of nodes is K and distance between adjacent nodes is 200 km. The number of wavelengths is eight at each link and the transmitting speed of a wavelength is 10 Gbps. Each node has full-range wavelength conversion capability, and the processing time of a control packet at each node is $\delta = 1.0$ ms.

We assume that the number of data blocks to be simultaneously transmitted is β, and that the size of data block is D bytes. Bursts are generated from the data blocks with our proposed methods. The bursts arrive at some node in the ring network according to a Poisson process with rate λ [request/ms]. The pair of source and destination nodes of the bursts is distributed uniformly, i.e., any pair is selected with the same probability.

As for background traffic, non-reliable background bursts arrive at the ring network according to a Poisson process with rate λ_{BG} [request/ms]. The size of a background burst is exponentially distributed with the mean D bytes. For both the multiple-bursts transmission and the background burst transmission, the offset time of each transmission is given by $\Delta = (H + 1)\delta$ ms when the number of transmission hops is H. The transmission interval between the burst and its corresponding control packet is given by the offset time Δ.

Note that multiple-bursts for β original data blocks are regarded as one transmission request. From this view point, a transmission request succeeds if β original data blocks are eventually transmitted to the destination with or without FEC recovery. Otherwise, the transmission request fails. We define the failure rate as the ratio of the number of failure requests to the number of overall transmission requests. In the following, the performance measure is the failure rate.

We compare the proposed methods with the extra-offset time method. In the extra-offset time method, β multiple bursts are consecutively transmitted as well as the proposed methods, but the extra offset time is applied to each burst transmission. When the number of transmission hops is H and the extra offset time is E ms, the transmission interval between the burst and its corresponding control packet is given by $(H + 1)\delta + E$.

5.1.1 Impact of FEC Processing

In this subsection, we investigate the efficiency of the FEC recovery in the ring network with $K = 6$ nodes. We compare the proposed methods with the consecutive multiple-bursts transmission without FEC. We assume that the number of original data blocks β is 10. From this assumption, the number of bursts for the OBG is 11, while that for the IBG is 10. We also assume that the burst size D is 1.0 Mbytes. The arrival rate of background traffic λ_{BG} is set to 2.0.

Figure 7 shows the transmission failure rates of three methods against the arrival rate of transmission requests. From Fig. 7, we can observe that the transmission failure rates of both proposed methods are smaller than that of the method without FEC regardless of the arrival rate λ. This is because the lost burst is effectively recovered for the proposed methods. A remarkable point is that the proposed methods can decrease the failure rate by 90%. This also implies that the code rates $\beta/(\beta + 1) = 10/11$ for the OBG and the code rate $(\beta - 1)/\beta = 9/10$ for the IBG are significantly effective for the proposed methods.

Moreover, we find that the transmission failure rate of the IBG method is smaller than that of the OBG method. This is because the number of bursts for the IBG is smaller than that for the OBG, resulting in the efficient use of wavelengths. Therefore, the IBG method is more effective than the OBG method.

Next, we investigate how the number of data blocks β affects the recovery performance of the proposed method. Note that β is directly related to the code rates of the OBG and IBG. Figure 8 shows failure rates for the proposed methods against β. Here, we set $\lambda = 1.5/\beta$, $\lambda_{BG} = 3.0$, and $D = 1.0$ Mbytes. λ is determined such as the offered load does not change against β.

We observe from Fig. 8 that the failure rate becomes large when β increases. This is because a large β makes the code rate large, resulting in the decrease of the amount of

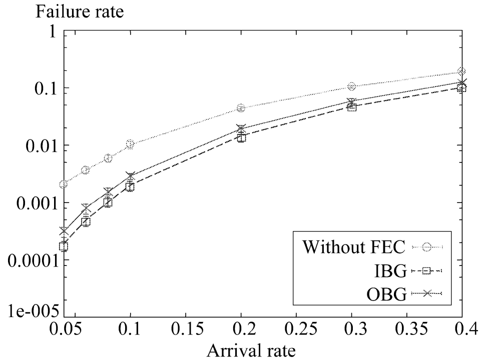


Fig. 7 Comparison of the consecutive burst transmission methods with and without FEC.

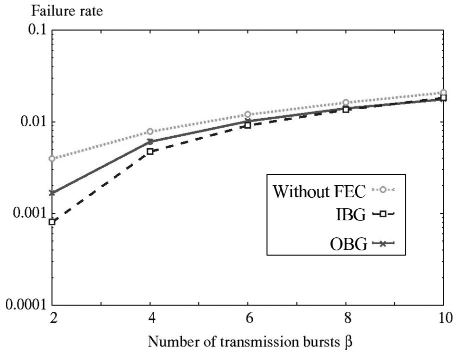


Fig. 8 Transmission failure rate vs. number of transmission bursts β.

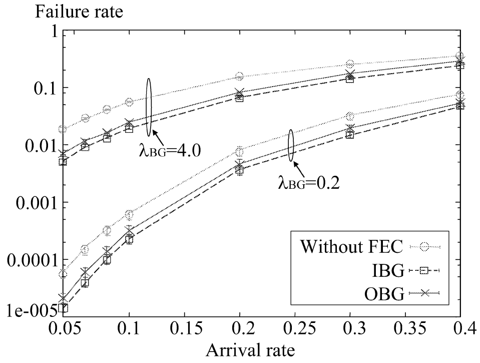
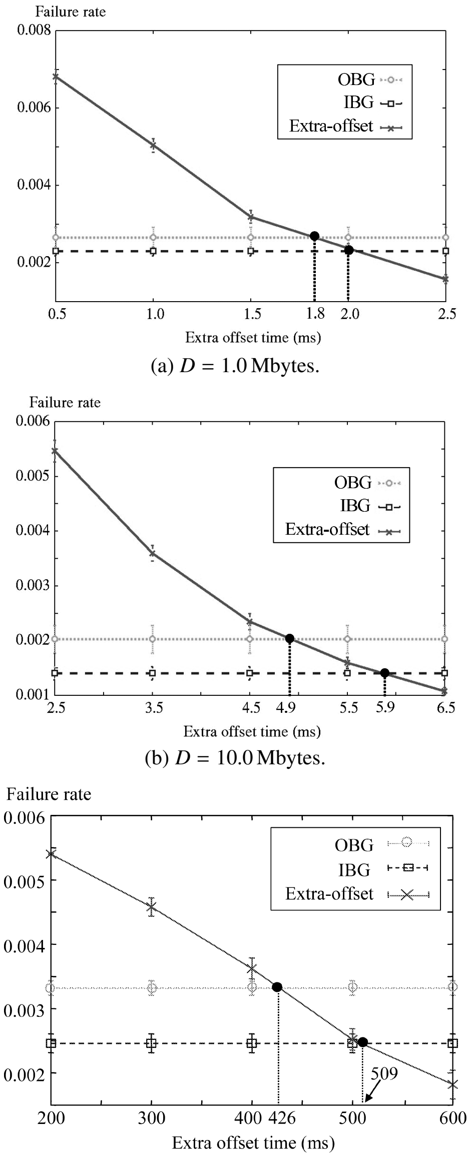


Fig. 9 Transmission failure rate vs. arrival rate λ in cases of λ_{BG} = 0.2 and 4.0.



(a) D = 1.0 Mbytes.

(b) D = 10.0 Mbytes.

(c) D = 1.0 Gbytes.

Fig. 10 Comparison of the proposed and extra-offset time methods.

data to be recovered. This result implies that our proposed methods are effective when β is small, however, the overhead of FEC processing is large due to a small code rate.

5.1.2 Impact of Background Traffic

Next, we investigate the impact of the arrival rate of background traffic λ_{BG} on the performances of OBG and IBG. As is the case with Fig. 7, we set $K = 6$, $\beta = 10$, and D = 1.0 [Mbytes].

Figure 9 shows the transmission failure rates of OBG, IBG, and the multiple-bursts transmission without FEC in cases of λ_{BG} = 0.2 and 4.0. On the other hand, Fig. 7 shows the failure rates of the three methods in the case of λ_{BG} = 2.0.

From Figs. 7 and 9, we observe that these failure rates increase (decrease) as λ_{BG} becomes large (small), as expected. However, the differences among these failure rates are not significantly affected by the arrival rate of background traffic λ_{BG}. Therefore, the performances of our proposed methods are insensitive to the background traffic.

5.1.3 Impact of Extra Offset Time

In this subsection, we compare the proposed methods and the extra-offset time method. We investigate how the extra offset time improves the failure rate of the extra-offset time method in comparison with the proposed method. Here, we set K = 6 and β = 10. When the burst size is D = 1.0 Mbytes, we set λ = 0.1 and λ_{BG} = 2.0. On the other hand, when the burst size is D = 10.0 Mbytes, we set λ = 0.013 and λ_{BG} = 0.25. In addition, when the burst size is D = 1.0 Gbytes, we set λ = 0.0001 and λ_{BG} = 0.0022.

Figures 10(a), (b), and (c) show the transmission failure rates of our proposed methods and that of the extra-offset time method in cases of the burst size D = 1.0 Mbytes, 10.0 Mbytes, and 1.0 Gbytes, respectively. Note that the failure rates of the proposed methods are independent of the extra offset time and this results in the constant failure rates against the extra offset time.

From Fig. 10(a), we observe that the transmission failure rate of the extra-offset time method decreases as the extra offset time increases, and that when the extra offset time is larger than 1.8 ms, the transmission failure rate of the extra-offset time method is smaller than that of the OBG method. Moreover, when the extra offset time is larger than 2.0 ms, the transmission failure rate of the extra-offset time method is smaller than that of the IBG method. This implies that the OBG (IBG) is more effective than the extra-offset time method if the FEC processing time is smaller than 1.8 ms (2.0 ms) (see black circles in Fig. 10(a)).

On the other hand, from Fig. 10(b) where the burst size is 10.0 Mbytes, if the extra offset time is smaller than 4.9 ms (5.9 ms), the OBG (IBG) method is more effective than the extra-offset time method (see black circles in Fig. 10(b)). Moreover, when the burst size is 1.0 Gbytes, the OBG (IBG) method is more effective than the extra-offset time method if the extra offset time is smaller than 426 ms (509 ms) (see black circles in Fig. 10(c)). From these figures, we observe that our proposed methods are significantly effective when the extra-offset time is small. We also find that the IBG method is more effective than the OBG method.

5.1.4 Impact of the FEC Processing Speed

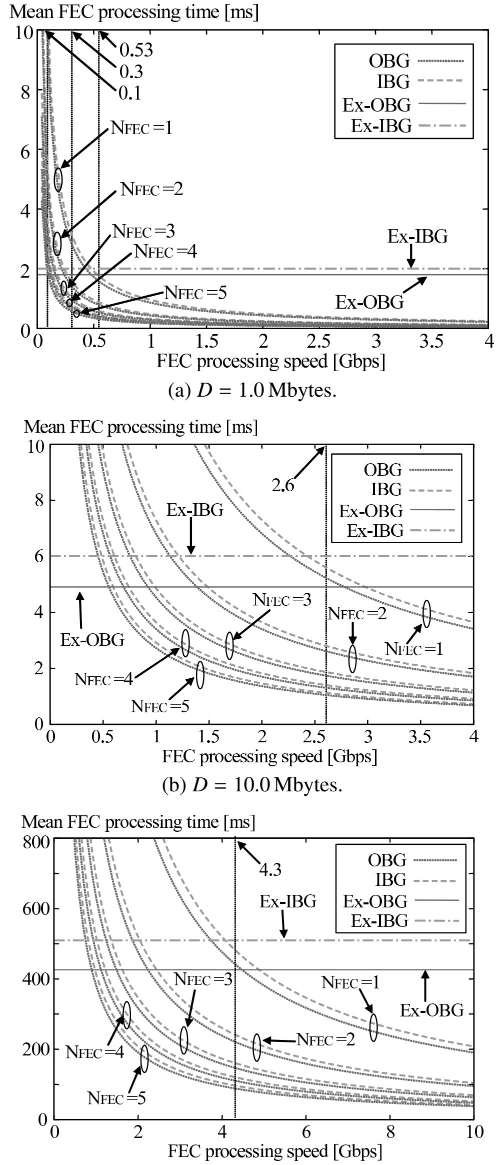
Remind that the FEC decoding process is performed only when a burst loss occurs for a multiple bursts transmission. Therefore, the mean end-to-end transmission delay for the OBG (IBG) method is greatly affected by the mean FEC processing time $E[T_{FEC}^{(OBG)}]$ ($E[T_{FEC}^{(IBG)}]$). On the other hand, in the extra-offset time method, the extra-offset time E is added to all bursts regardless of burst losses, and the mean end-to-end transmission delay depends on the extra-offset time. In this subsection, we compare the mean FEC processing time $E[T_{FEC}^{(OBG)}]$ ($E[T_{FEC}^{(IBG)}]$) with the extra-offset time.

With the parameter setting of Figs. 10(a), (b), and (c), we calculated $E[T_{FEC}^{(OBG)}]$ and $E[T_{FEC}^{(IBG)}]$ from Eqs. (4) and (7), respectively. Here, the conditional probabilities in (4) and (7) were obtained from simulation.

Figures 11(a), (b), and (c) show how the FEC processing speed L affects the mean FEC processing time in cases of $D = 1.0$ Mbytes, 10 Mbytes, and 1.0 Gbytes, respectively. In these figures, N_{FEC} represents the number of FEC processors at each node and the range of N_{FEC} is from one to five. Curved lines denote the mean FEC processing times for the OBG and IBG methods. Note that in these figures, the mean FEC processing time of the OBG is almost the same as that of the IBG.

The straight line Ex-OBG (Ex-IBG) denotes the extra-offset time such that the burst loss probability of the extra-offset time method becomes the same as that of the OBG (IBG). If the mean FEC processing time is smaller than the corresponding extra-offset time, our proposed methods can provide a smaller mean end-to-end delay than the extra-offset time method.

From Fig. 11(a), we find that when L is larger than 0.53 Gbps, the mean processing times of the proposed methods are smaller than the extra-offset time regardless of the number of processors N_{FEC}. Therefore, if L is larger than 0.53 Gbps, our proposed methods are effective with only one FEC processor. As the FEC processing speed L becomes small, more FEC processors are required to provide a delay equivalent to the extra-offset time scheme. For example, when the FEC processing speed L is 0.3 Gbps, two FEC processors are required. Similarly, if the FEC processing speed L is 0.1 Gbps, five FEC processors are required.



(a) $D = 1.0$ Mbytes.

(b) $D = 10.0$ Mbytes.

(c) $D = 1.0$ Gbytes.

Fig. 11 Mean FEC processing time vs. FEC processing speed.

From Fig. 11(b), when the FEC processing speed L is 2.6 Gbps, we find that the IBG requires one FEC processor. On the other hand, the OBG method requires two FEC processors. Although the FEC processing time of the IBG is larger than that of the OBG from (6), the IBG requires a smaller number of FEC processors than the OBG because the failure rate of the IBG method is smaller than that of the OBG method. Moreover, in Fig. 11(c), the OBG (IBG) method requires two FEC processors (one FEC processor) when the FEC processing speed L is 4.3 Gbps.

Comparing these three figures, we observe that the

Table 1 Relationship between the required minimum FEC processing speed L and the number of the required FEC processors N_{FEC} for each data block size D (ring network).

D = 1.0	N_{FEC}	1	2	3	4	5
Mbytes	L for OBG	0.53	0.26	0.17	0.13	0.10
(Fig. 11(a))	L for IBG	0.43	0.21	0.14	0.11	0.08
D =10.0	N_{FEC}	1	2	3	4	5
Mbytes	L for OBG	2.97	1.48	0.99	0.74	0.59
(Fig. 11(b))	L for IBG	2.27	1.13	0.75	0.56	0.45
D =1.0	N_{FEC}	1	2	3	4	5
Gbytes	L for OBG	4.87	2.43	1.62	1.21	0.97
(Fig. 11(c))	L for IBG	3.74	1.87	1.24	0.93	0.74

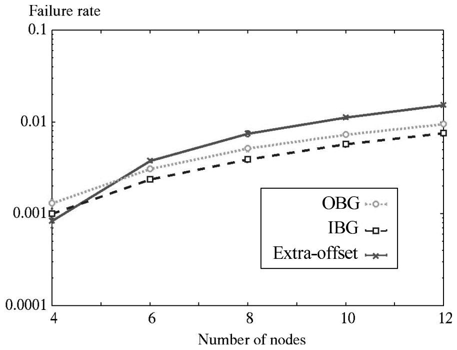


Fig. 12 Transmission failure rate vs. number of nodes.

number of the required FEC processors increases as the burst size becomes large. Nevertheless, it is possible to decrease the mean FEC processing time by enhancing the processing speed of the FEC processor or by increasing the number of FEC processors. The relationship between the required minimum FEC processing speed L and the number of the required FEC processors N_{FEC} is shown in Table 1.

5.1.5 Impact of the Number of Nodes

In this subsection, we investigate the impact of the number of nodes K in the ring network on the performances of the three methods. We set $D = 5.0$ Mbytes, $\lambda = 0.025$, and $\lambda_{BG} = 0.5$. The extra offset time is set to 3.0 ms.

From Fig. 12, we observe that when the number of nodes is larger than five, the failure rate of the IBG method is smaller than that of the extra-offset time method. As the number of nodes becomes larger than six, the failure rate of the OBG method is also smaller than that of the extra-offset time method. This is because the advantage of the extra offset time becomes small as the number of hops the control packet has passed through is large. Therefore, the proposed methods are significantly effective for the OBS network with large number of hops.

5.2 NSFNET

In this section, we evaluate the performances of the OBG and IBG in NSFNET with 14 nodes (see Fig. 13). The number of wavelengths is four and the transmitting speed of a wavelength is 10 Gbps. Each node has full-range wavelength conversion capability. The distances between adjacent nodes are from 300 km to 2,800 km and those are depicted in Fig. 13. A static route between source and destination nodes is chosen according to the minimum hop routing. In this case, the maximum number of hops is three.

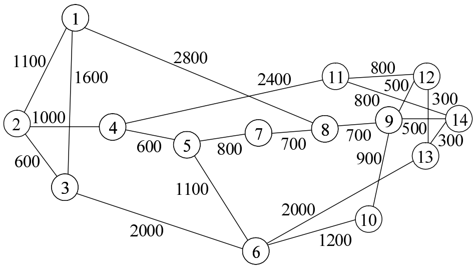


Fig. 13 NSFNET with 14 nodes.

Here, the number of data blocks is $\beta = 10$ and the size of data block is D bytes. Bursts are generated from the data blocks with our proposed methods. The bursts arrive at a node according to a Poisson process with rate λ [request/ms]. The pair of source and destination nodes of the bursts is also distributed uniformly.

As for background traffic, non-reliable background bursts arrive at the network according to a Poisson process with rate λ_{BG} [request/ms]. The size of a background burst is exponentially distributed with the mean D bytes. The processing time of a control packet at each node is δ =1.0 ms.

5.2.1 Comparison of Failure Rate

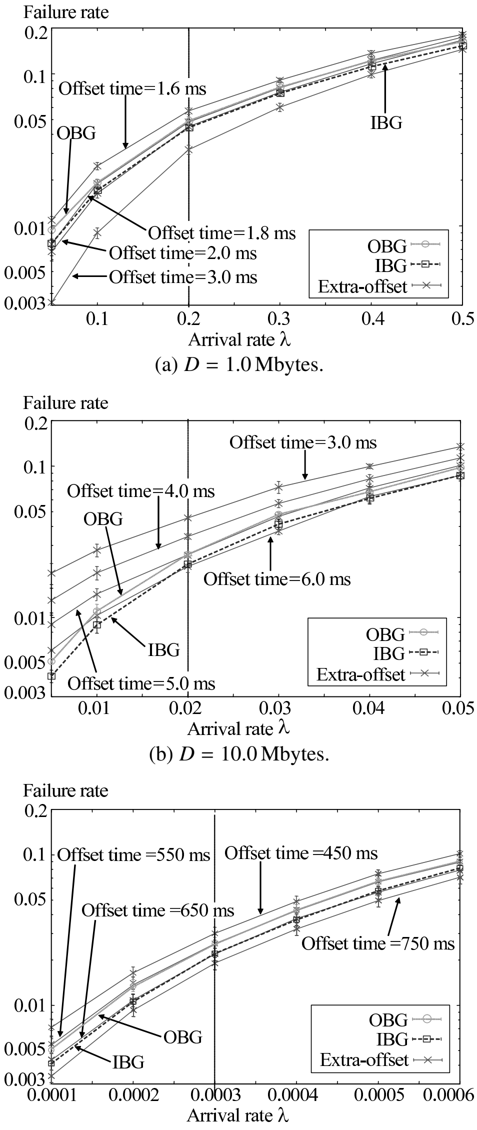
Figures 14(a), (b), and (c) show the transmission failure rates of the OBG, IBG, and extra-offset time methods in the cases of D = 1.0 Mbytes, 10 Mbytes, and 1.0 Gbytes, respectively. Here, we set λ_{BG} to 2.0 in Fig. 14(a), 0.27 in Fig. 14(b), and 0.0022 in Fig. 14(c). The extra offset time is set to 1.6, 1.8, 2.0, and 3.0 ms in Fig. 14(a), 3.0, 4.0, 5.0, and 6.0 ms in Fig. 14(b), and 450, 550, 650, and 750 ms in Fig. 14(c).

From Figs. 14(a), (b), and (c), we find that the transmission failure rate of the IBG method is smaller than that of the OBG method, as expected. Therefore, even in the NSFNET, the IBG method is more effective than the OBG method in terms of the transmission failure rate.

From these three figures, we find that the extra-offset time method requires a large extra offset time as the burst size D increases. This is because a burst with large size uses wavelengths for a long time and the effectiveness of the extra offset time becomes small. Therefore, in terms of the transmission failure rate, our proposed methods are more effective than the extra-offset time method when the burst size is large.

5.2.2 Comparison of Mean Burst Transmission Delay

From Fig. 14(a) in the previous subsection, when λ is equal to 0.2, we find that the transmission failure rate of OBG (IBG) is almost the same as that of the extra-offset time

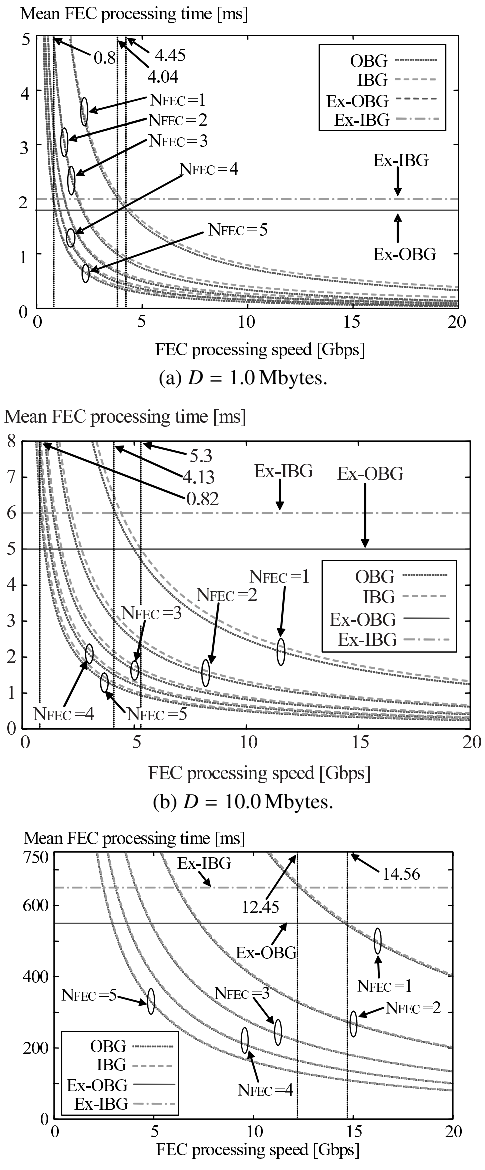


(a) D = 1.0 Mbytes.

(b) D = 10.0 Mbytes.

(c) D = 1.0 Gbytes.

Fig. 14 Transmission failure rate vs. arrival rate λ.



(a) D = 1.0 Mbytes.

(b) D = 10.0 Mbytes.

(c) D = 1.0 Gbytes.

Fig. 15 Mean FEC processing time vs. FEC processing speed.

method whose extra offset time is 1.8 ms (2.0 ms). On the other hand, in the case of $\lambda = 0.02$ in Fig. 14(b), the transmission failure rate of the OBG (IBG) is almost the same as that of the extra-offset time method whose extra offset time is 5.0 ms (6.0 ms). Moreover, in the case of $\lambda = 0.0003$ in Fig. 14(c), the transmission failure rate of the OBG (IBG) is almost the same as that of the extra-offset time method whose extra offset time is 550 ms (650 ms). In these cases, if the mean end-to-end transmission delay of the OBG (IBG) is smaller than that of the extra-offset time method, the OBG (IBG) is more effective than the extra-offset time method.

In this subsection, we compare the mean FEC processing times of the OBG and IBG methods $E[T_{FEC}^{(OBG)}]$ and $E[T_{FEC}^{(IBG)}]$ with the extra offset time. As is the case with the Sect. 5.1.4, we computed the mean FEC processing times $E[T_{FEC}^{(OBG)}]$ and $E[T_{FEC}^{(IBG)}]$ according to (4) and (7). Here, the corresponding conditional probabilities were obtained from the simulation results in Fig. 14, and T_{OBG} and T_{IBG} were calculated from (2) and (5), respectively.

Figures 15(a), (b), and (c) show how the FEC processing speed L affects the mean FEC processing times of the OBG and IBG in cases of D = 1.0 Mbytes, 10 Mbytes, and 1.0 Gbytes, respectively. In these figures, the range of N_{FEC} is from one to five.

From Fig. 15(a), we find that when L is larger than 4.45 Gbps, the mean FEC processing times of the proposed methods are smaller than the extra-offset time regardless of the number of FEC processors N_{FEC}. However, if L becomes smaller than 4.04 Gbps, our proposed methods become ineffective than the extra-offset time method using one FEC processor. On the other hand, as the number of FEC processors increases, the required FEC processing time becomes small. If the number of FEC processors is five, the

Table 2 Relationship between the required minimum FEC processing speed L and the number of the required FEC processors N_{FEC} for each data block size D (NSFNET).

D =1.0	N_{FEC}	1	2	3	4	5
Mbytes	L for OBG	4.45	2.22	1.48	1.11	0.89
(Fig. 15(a))	L for IBG	4.04	2.02	1.34	1.01	0.80
D =10.0	N_{FEC}	1	2	3	4	5
Mbytes	L for OBG	5.30	2.65	1.76	1.32	1.06
(Fig. 15(b))	L for IBG	4.13	2.06	1.37	1.03	0.82
D =1.0	N_{FEC}	1	2	3	4	5
Gbytes	L for OBG	14.56	7.28	4.85	3.64	2.91
(Fig. 15(c))	L for IBG	12.45	6.22	4.15	3.11	2.49

required FEC processing speed is more than 0.8 Gbps.

On the other hand, from Fig. 15(b), the FEC processor speed L larger than 5.3 Gbps (4.13 Gbps) is needed in the OBG (IBG) method when the number of FEC processor is one. However, if multiple FEC processors are available, the FEC processors with smaller FEC processing time can be used. For example, when five FEC processors are available, the required FEC processing time for the OBG (IBG) is 1.06 Gbps (0.82 Gbps).

In addition, from Fig. 11(c), the FEC processor speed L larger than 14.56 Gbps (12.45 Gbps) is needed in the OBG (IBG) method when the number of FEC processor is one. However, if multiple FEC processors are available, the FEC processors with smaller FEC processing time can be used. Therefore, even in the NSFNET, it is expected that the proposed methods are effective using multiple FEC processors. The relationship between the required minimum FEC processing speed L and the number of the required FEC processors N_{FEC} is shown in Table 2.

6. Conclusions

In this paper, we proposed two burst transmission methods in order to transmit multiple bursts reliably and simultaneously. In the OBG method, a redundant burst is generated from multiple bursts, and the multiple bursts and the redundant one are consecutively transmitted. On the other hand, the IBG method generates the same number of bursts as that of original data blocks.

Numerical results showed that the FEC recovery mechanism works quite well even with a small number of FEC processors. A remarkable point is that both the two proposed methods with a few FEC processors can achieve almost the same failure-rate performance as the conventional extra-offset time method. Moreover, we showed that the proposed methods are significantly effective for the OBS network where both the maximum number of hops and the processing time of a control packet are large. In our proposed methods, it is possible to decrease the end-to-end transmission delay by enhancing the FEC processor or by increasing the number of FEC processors.

References

[1] C. Qiao and M. Yoo, "Optical burst switching (OBS): A new paradigm for an optical Internet," J. High Speed Networks, vol.8, no.1, pp.36–44, Jan. 2000.

[2] T. Tachibana, T. Ajima, and S. Kasahara, "Round-robin burst assembly and constant transmission scheduling for optical burst switching networks," Proc. IEEE Globecom'03, pp.2772–2776, Dec. 2003.

[3] M. Yoo, C. Qiao, and S. Dixit, "QoS performance of optical burst switching in IP-over-WDM networks," IEEE J. Sel. Areas Commun., vol.18, no.10, pp.2062–2071, Oct. 2000.

[4] I. Baldine, G.N. Rouskas, H.G. Perros, and D. Stevenson, "Jump-Start: A Just-In-Time signaling architecture for WDM burst-switched networks," IEEE Commun. Mag., vol.40, no.2, pp.82–89, Feb. 2002.

[5] K. Kitayama, M. Koga, H. Morikawa, S. Hara, and M. Kawai, "Optical burst switching network testbed in Japan," Proc. OFC 2005, OWC3, March 2005.

[6] S. Oh, H.H. Hong, and M. Kang, "A data burst assembly algorithm in optical burst switching networks," ETRI Journal, vol.24, no.4, pp.311–322, Aug. 2002.

[7] Y. Sun, T. Hashiguchi, V. Minh, X. Wang, H. Morikawa, and T. Aoyama, "A burst-switched photonic network testbed: Its architecture, protocols and experiments," IEICE Trans. Commun., vol.E88-B, no.10, pp.3864–3873, Oct. 2005.

[8] Y. Sun, T. Hashiguchi, V. Minh, X. Wang, H. Morikawa, and T. Aoyama, "Design and implementation of an optical burst-switched network testbed," IEEE Commun. Mag., vol.43, no.11, pp.S48–S55, Nov. 2005.

[9] M. Yoo and C. Qiao, "Supporting multiple classes of services in IP over WDM networks," Proc. IEEE Globecom'99, pp.1023–1027, Dec. 1999.

[10] M. Yoo, C. Qiao, and S. Dixit, "Optical burst switching for service differentiation in the next-generation optical Internet," IEEE Commun. Mag., vol.39, no.9, pp.98–104, Feb. 2001.

[11] V.M. Vokkarane, J.P. Jue, and S. Sitaraman, "Burst segmentation: An approach for reducing packet loss in optical burst switched networks," Proc. IEEE ICC'02, pp.2673–2677, April/May 2002.

[12] T. Tachibana, M. Ueda, and S. Kasahara, "A preemptive scheme with two-way release message transmission in optical burst switching networks," Proc. Globecom'04, pp.1994–1998, Nov./Dec. 2004.

[13] L. Yang, Y. Jiang, and S. Jiang, "A probabilistic preemptive scheme for providing service differentiation in OBS networks," Proc. Globecom'03, pp.2689–2693, Dec. 2003.

[14] M. Leenheer, E. Breusegem, P. Thysebaert, B. Volckaert, F. Turck, B. Dhoedt, P. Demeester, D. Simeonidou, M. O'Mahoney, R. Nejabati, A. Tzanakaki, and I. Tomkos, "An OBS-based grid architecture," Proc. Globecom 2004 Workshop on High-Performance Global Grid Networks, pp.390–394, Nov. 2004.

[15] X. Mountrouidou, H. Perros, and M. Beshai, "Performance evaluation of an optical burst switching scheme for grid networks," Proc. GridNets 2005, Oct. 2005.

[16] T. Coutelen, H. Elviaze, and B. Jaumard, "Performance comparison of OCS and OBS switching paradigms," Proc. 7th International Conference on Transparent Optical Networks, pp.212–215, July 2005.

[17] F. Xue, S. Yoo, H. Yokoyama, and Y. Horiuchi, "Performance comparison of optical burst and switched networks," Proc. OFC 2005, OWC1, March 2005.

[18] M. Leenheer, P. Thysebaert, B. Volckaert, F. Turck, B. Dhoedt, P. Demeester, D. Simeonidou, R. Nejabati, G. Zervas, D. Klonidis, and M. O'Mahony, "A view on enabling-consumer oriented grids through optical burst switching," IEEE Commun. Mag., vol.44, no.3, pp.124–131, March 2006.

[19] S. Arima, T. Tachibana, and S. Kasahara, "FEC-based burst loss recovery for multiple-bursts transmission in optical burst switching networks," Proc. IEEE Globecom'05, pp.1994–1998, Nov./Dec. 2005.

[20] S.B. Wicker, Error Control Systems for Digital Communication and Storage, Prentice-Hall, 1995.

Satoshi Arima received the B.Eng. degree from Doshisha University, Japan, in 2003. He received the M.Eng. degree from the Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, Japan, in 2005. He joined Kyocera Corporation in 2005. His research interests include network architectures in optical networking.

Takuji Tachibana received the B.Eng. degree from the Department of Systems Engineering, Nagoya Institute of Technology, Japan, in 2000. He received the M.Eng. and Dr.Eng. degrees from the Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, Japan, in 2001 and 2004, respectively. From 2004 to 2006, he was an expert researcher in the Information and Network Systems Department, National Institute of Information and Communications Technology, Japan. Since 2006 he has been an assistant professor at the Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, Japan. His research interests include network architectures in optical networking and performance analysis of computer and communication systems. He is a member of the IEEE and the Operations Research Society of Japan.

Yuichi Kaji was born in Osaka, Japan, on December 23, 1968. He received the B.E., M.E., and Ph.D. degrees in information and computer sciences from Osaka University, Osaka, Japan, in 1991, 1992 and 1994, respectively. In 1994, he joined Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan. In 2003 and 2004, he visited the University of California Davis and the University of Hawaii at Manoa as a visiting researcher. His current research interests include the theory of error correcting codes, fundamental techniques for information security, and the theory of automata and rewriting systems. He is a member of IPSJ, SITA and IEEE.

Shoji Kasahara received the B.Eng., M.Eng., and Dr. Eng. degrees from Kyoto University, Kyoto, Japan, in 1989, 1991, and 1996, respectively. He was with the Educational Center for Information Processing, Kyoto University from 1993 to 1997 as an Assistant Professor. In 1996, he was a visiting scholar of University of North Carolina at Chapel Hill, NC, USA. From 1997 to 2005, he was with the Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology. Since 2005, he has been an Associate Professor of Department of Systems Science, Graduate School of Informatics, Kyoto University. His research interests include queueing theory and performance analysis of computer and communication systems. Dr. Kasahara is a member of the IEEE, the Operations Research Society of Japan, the Information Processing Society of Japan, and the Institute of Systems, Control and Information Engineers.