

# Delay Comparison of Automatic Repeat Request Assisted Butterfly Networks

Yang Qin and Lie-Liang Yang

School of ECS, University of Southampton, SO17 1BJ, United Kingdom

Tel: 0044-(0)23-8059 3364, Email: yq06r,lly@ecs.soton.ac.uk; http://www-mobile.ecs.soton.ac.uk

**Abstract**—The average burst delay and the standard deviation (SD) of burst delay in the context of the Butterfly networks are investigated and compared, when three types of automatic repeat request (ARQ) schemes, namely, the stop-and-wait ARQ (SW-ARQ), go-back- $N$  ARQ (GBN-ARQ) and the selective-repeat ARQ (SR-ARQ), are considered. We assume that packets in a burst arrive at a constant rate and that the packet error rate (PER) is the same for all the links. Our studies show that, among the three ARQ schemes, the SW-ARQ scheme yields the highest average burst delay and also the highest SD of burst delay, while the SR-ARQ scheme is capable of attaining the lowest delay and also the lowest SD of burst delay. At very low PER, the GBN-ARQ may achieve a similar delay performance as the SR-ARQ. However, as the PER increases, the delay performance of the GBN-ARQ converges to that of the SW-ARQ.

**Index Terms**—Network coding, automatic repeat request, delay.

## I. INTRODUCTION

Packet-level network coding has drawn a lot of research, since its invention [1, 2]. In recent years, the performance of communications networks employing network coding has been investigated widely, but, mainly under the assumption that packets are conveyed over networks without transmission errors [3]. In practice, however, communication channels are not ideal and transmission errors are always unavoidable. Hence, in communications systems, error-control techniques are usually required in order to make the error probability of received data as low as possible [2, 4]. However, when some error-control schemes, such as the ARQ scheme, are introduced, transmission delay may incur, which may also affect the quality-of-services (QoS).

In this contribution, we motivate to study by simulation the delay performance of the Butterfly network, when it is operated in non-ideal communications environments, where information is delivered under the protection of a certain ARQ error-control scheme [5]. Specifically, in this paper we consider three types of ARQ schemes, namely, the SW-ARQ, GBN-ARQ and the SR-ARQ [5]. We investigate both the average burst delay and the standard deviation (SD) of burst delay, both of which are critical for attaining high QoS of communications. For example, for the service of Voice over Internet Protocol (VoIP), it is well-known that high average delay of packets results in high latency of the speech being delivered to the receiver, while high jitter may result in both increased latency and packet loss [6]. The delay performance of the Butterfly network using the above-mentioned three types of ARQ schemes is compared. Our studies show that, for a given PER, the SR-ARQ scheme is

capable of attaining the lowest delay and also the lowest SD of delay, among the three ARQ schemes considered. Hence, the SR-ARQ scheme may be one of the appropriate schemes for supporting data transmission in the Butterfly-style networks. However, we should realize that the SR-ARQ scheme has the highest implementation complexity, while the SW-ARQ scheme has the lowest complexity, among the three ARQ schemes considered.

Previous researches related to this study can be briefly summarized as follows. In [7], network coding in networks with feedbacks has first been studied. In [8], the problems of applying ARQ techniques to network coding have been addressed. It has been suggested to add an extra network coding layer into the TCP/IP stack, in order to introduce network coding into the existing Internet infrastructure. Furthermore, in [9], a random network coding framework employing hybrid ARQ scheme has been proposed for real-time media broadcast over single-hop wireless networks.

## II. BUTTERFLY NETWORK AND OPERATIONS

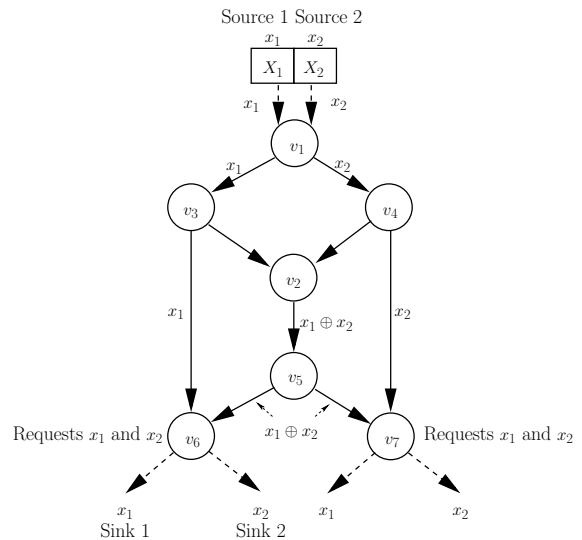


Fig. 1. A butterfly network with one source node  $v_1$  multicasting messages  $x_1$  and  $x_2$  to both the sink nodes  $v_6$  and  $v_7$  using linear network coding.

The system considered in this contribution is a typical Butterfly network as shown in Fig. 1. Node  $v_1$  is a source node, where two information sources  $X_1$  and  $X_2$  generate the packets transmitted simultaneously to the sink nodes  $v_6$  and  $v_7$ . Each of the sink nodes is attached with two sinks, for

example, Sink 1 and Sink 2 to the sink node  $v_6$ , as shown in Fig. 1. Node  $v_2$  is a two-input-single-output (2ISO) coding node with packet-level network coding. Node  $v_3$ ,  $v_4$  and  $v_5$  are the intermediate nodes, which forward incoming packets to the designated outgoing link(s) without network coding. The function of the intermediate nodes is identical to the conventional “store-and-forward” nodes. Finally, nodes  $v_6$  and  $v_7$  are two sink nodes expecting information from both sources  $X_1$  and  $X_2$ . For convenience of description, let us define the link set  $\mathcal{L} = \{l_{1,3}, l_{1,4}, l_{2,5}, l_{3,2}, l_{3,6}, l_{4,2}, l_{4,7}, l_{5,6}, l_{5,7}\}$ , where  $l_{i,j}$  is the link transmitting packets from node  $v_i$  to node  $v_j$ . We assume that packets are transmitted over the communication links based on the ARQ strategy. In this paper, three types of ARQ schemes, namely the SW-ARQ, GBN-ARQ and the SR-ARQ, are studied in the context of the Butterfly networks. In our analysis and simulations, the following assumptions are adopted.

- All the links  $\{l_{i,j} | l_{i,j} \in \mathcal{L}\}$  of the Butterfly network employ the same ARQ scheme, SW-ARQ, GBN-ARQ or SR-ARQ.
- The  $n$ th packet pair simultaneously generated by  $X_1$  and  $X_2$  are denoted as  $x_1(n)$  and  $x_2(n)$ , respectively, where  $n$  is defined as the “generation” number. Let  $x(n) = (x_1(n), x_2(n))$  denote the  $n$ th packet pair. We assume that  $x(n)$  arrives at  $v_1$  immediately after it is generated.
- Each of the links in  $\mathcal{L}$  is divided into two channels: a forward channel and a feedback channel. The forward channel is assumed to be a binary symmetric channel (BSC). The probability of detectable packet errors of the forward channel is the same for every link, which is denoted as  $p_e$  and simply referred to as the PER. This assumption means that the probability of undetectable packet errors is small and can be ignored. This is usually true, since for most error-control codes adopted for ARQ, the probability of undetectable errors is very small in comparison with the probability of detectable errors. Furthermore, we assume that the link outage rate is zero, implying that no packet is lost during transmission. Additionally, the feedback channels are assumed ideal and do not generate erroneous feedbacks.
- The round trip time (RTT), denoted by  $T$ , represents the time duration between the time a node sends a packet and the time it receives a confirmation signal. We assume that half of a RTT, i.e.,  $T/2$ , is required for transmitting a packet from a transmit node to a receive node by the corresponding forward channel. Similarly, half a RTT is required for sending a confirmation signal from the receive node to the transmit node using the corresponding feedback channel.
- The duration of packets is assumed to be much shorter than  $T$  of the RTT and can be ignored. Furthermore, the packet processing time can also be ignored. Alternatively, we may view that the average duration of packets as well as the average processing time are included in the RTT.
- Each of the nodes is assumed to have an infinite buffer for storing the packets, whenever necessary.

Based on the above assumptions, the operations of the pack-

ets over the Butterfly network of Fig. 1 are carried out as follows.

As shown in Fig. 1, we assume that two unit rate messages (packets)  $x_1(n), x_2(n)$  are generated by the information sources  $X_1$  and  $X_2$ , which are attached to the source node  $v_1$ . We assume for simplicity that both  $x_1(n)$  and  $x_2(n)$  take values in the field  $\mathbb{F} = GF(2)$ . As shown in Fig. 1, message  $x_1(n)$  is directly sent to the sink node  $v_6$  via the route  $X_1 \rightarrow v_1 \rightarrow v_3 \rightarrow v_6$ . Similarly, message  $x_2(n)$  is directly sent to the sink node  $v_7$  via the route  $X_2 \rightarrow v_1 \rightarrow v_4 \rightarrow v_7$ . As shown in Fig. 1, both the messages  $x_1(n)$  and  $x_2(n)$  are also sent to node  $v_2$  via the route  $X_1 \rightarrow v_1 \rightarrow v_3 \rightarrow v_2$  for  $x_1(n)$  and  $X_2 \rightarrow v_1 \rightarrow v_4 \rightarrow v_2$  for  $x_2(n)$ , respectively. At  $v_2$ , once  $x_1(n)$  and  $x_2(n)$  are successfully received, they are encoded to form the message  $x_1(n) \oplus x_2(n)$  based on the modulo-2 addition over the field  $\mathbb{F} = GF(2)$ . The output message  $x_1(n) \oplus x_2(n)$  is then sent to the sink nodes  $v_6$  and  $v_7$  via the route  $v_2 \rightarrow v_5 \rightarrow v_6$  and  $v_2 \rightarrow v_5 \rightarrow v_7$ , respectively. Finally, after the sink nodes  $v_6$  and  $v_7$  receive the message  $x_1(n) \oplus x_2(n)$ , the sink node  $v_6$  recovers  $x_2(n)$  by the operation  $x_2(n) = x_1(n) \oplus (x_1(n) \oplus x_2(n))$ , while the sink node  $v_7$  recovers  $x_1(n)$  based on the operation  $x_1(n) = x_2(n) \oplus (x_1(n) \oplus x_2(n))$ . Consequently, both of the sink nodes  $v_6$  and  $v_7$  can obtain the messages  $x_1(n)$  and  $x_2(n)$  generated by the sources  $X_1$  and  $X_2$ , respectively.

From the above description as well as Fig. 1, we can see that, in the Butterfly network, there are two types of packet transmission paths. The first type do not go through the coding node  $v_2$ , while the second type contains the coding node  $v_2$ . The data transmission on the paths without involving the coding node is the same as that on the conventional relay links, where each node just stores the packet received from the incoming link and then forwards it to the next node using its outgoing link. By contrast, the paths containing the coding node behave differently from conventional relay links. At the coding node, rather than just forwarding the incoming packets, for each outgoing link, the coding node needs to wait for all the required incoming packets involved in order to form a corresponding outgoing coded packet.

### III. SYSTEM SETTINGS AND ASSUMPTIONS

In the Butterfly network as shown in Fig. 1, the packets are conveyed from one node to another protected by an ARQ scheme. It is well-known that ARQ protocols include many different schemes [5, 10]. In this paper, we consider three representative ARQ schemes, which are the SW-ARQ, GBN-ARQ and SR-ARQ. The principles of these ARQ schemes can be found in many text books, such as in [10].

In this paper, we investigate and compare the delay performance of the Butterfly networks, when the above-mentioned three types of ARQ schemes are invoked. Both the average burst delay and the standard deviation (SD) of the burst delay are interested. In this contribution, the systems simulated use the following assumptions and settings.

We assume that there is a packet burst of size 32, as shown in the figures, to be transmitted. The burst is divided into two packet bursts of each with 16 packets, which are then sent from the source nodes to the corresponding sink nodes. One of the

packet bursts is sent from information source  $X_1$  to the Sink 1 and the other packet burst is sent from information source  $X_2$  to the Sink 2. In the two parallel bursts, the packets with the same generation serial number are assumed to arrive at the system at the same time. Furthermore, we assume that in each burst the packets arrive at the input of the system with a constant inter-arrival time and there is no input before or after each burst.

Let  $U$  denote the inter-arrival time between two adjacent packets and let  $N$  denote the number of packets in one burst. Then, the time duration of the burst is  $(N - 1) \times U$ , since the packet length is ignored. Additionally, let  $\lambda$  denote the arrival rate of the packets. Then, we have  $\lambda = 1/U$ . Note that, this input pattern is a setting intimating the traffic generated by a constant-bit-rate (CBR) voice codec, e.g. the ITU G.711 a-Law codec [11], where the packet rate is constant and the codec switches off during silence [12, 13].

In our simulations, the ARQ schemes are set up as follows. For both the GBN-ARQ and SR-ARQ schemes, we assume that the transmitter is able to send out four packets during one RTT. Since for both the GBN-ARQ and SR-ARQ schemes, larger queueing buffer results in higher throughput [14], hence, for fairness of comparison, the transmission window for both the GBN-ARQ and SR-ARQ schemes is set to a length of eight packets.

In our simulations, the observation starts at  $t = 0$  and ends at the moment when all the packets are successfully received by the sinks. The main concern in this section is the time spent by each of the two bursts to go through the Butterfly network. This time is defined as the delay, which is the time from the moment that a packet arrives at the source node to the moment that the packet is successfully received by a specific sink.

The average burst delay of the Butterfly network employing the SW-ARQ, GBN-ARQ and SR-ARQ schemes is obtained from the delay  $\tau$  of a burst, which is a random variable. Let  $\tau_1, \tau_2, \dots, \tau_M$  denote the  $M$  observations of  $\tau$ , and let  $E[\tau]$  denotes the average delay. Then, the average burst delay  $E[\tau]$  can be approximately evaluated by

$$E[\tau] \approx \frac{1}{M} \sum_{i=1}^M \tau_i \quad (1)$$

where  $M = 10000$  is used in our simulations. Note that, in a more complicated network, one packet may be multicasted to different information sinks. In this case, the delays corresponding to different paths may be different.

In addition to the average delay, the SD of the delay is also studied in this section. Let the SD of delay be denoted by  $\sigma$ . Then, we have

$$\sigma = \sqrt{\frac{1}{M} \sum_{i=1}^M (\tau_i - E[\tau])^2}. \quad (2)$$

Upon substituting (1) into (2), the SD can be approximately evaluated as

$$\sigma \approx \sqrt{\frac{1}{M} \sum_{i=1}^M \left[ \tau_i - \left( \frac{1}{M} \sum_{i=1}^M \tau_i \right) \right]^2}. \quad (3)$$

Let us now illustrate some performance results.

#### IV. PERFORMANCE RESULTS

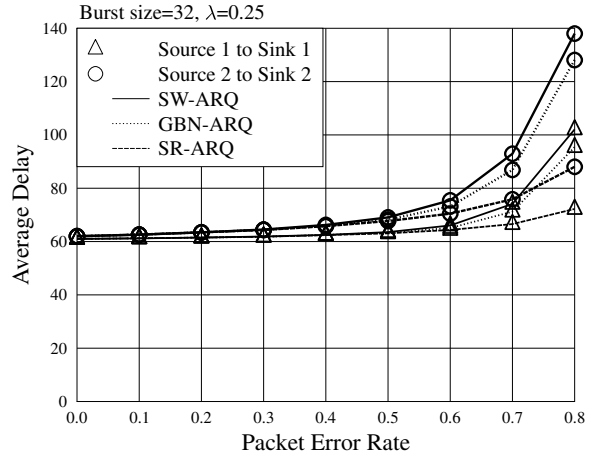


Fig. 2. Average delay versus packet error rate performance of the Butterfly network with different ARQ schemes.

Fig. 2 shows the average delay of both the path from source  $X_1$  to Sink 1 and the path from source  $X_2$  to Sink 2, when assuming  $\lambda = 0.25$ . The duration of the burst sent from each path can be calculated, which is

$$\frac{1}{\lambda} \times (N - 1) = 4 \times 15 = 60 \text{ (RTT)}. \quad (4)$$

When the PER is  $p_e = 0$ , each packet needs one RTT to be transmitted from source  $X_1$  to Sink 1, and each packet needs two RTTs from source  $X_2$  to Sink 2. Therefore, at  $p_e = 0$ , the delay for transmitting the burst from source  $X_1$  to Sink 1 is  $60 + 1 = 61$  RTTs, and the delay for transmitting the burst from source  $X_2$  to Sink 2 is  $60 + 2 = 62$  RTTs. As seen in Fig. 2, as the PER increases, for each path, all the three ARQ schemes yield a similar delay performance, when the PER is relatively low, such as, when  $p_e \leq 0.5$ . When  $p_e > 0.5$ , the average delay increases significantly, as  $p_e$  increases. In this region, for each of the two paths, the SW-ARQ scheme generates the highest delay, while the SR-ARQ scheme yields the lowest delay. Furthermore, for each of the ARQ schemes, the average delay of the path from source  $X_2$  to Sink 2 is longer than that of the path from source  $X_1$  to Sink 1.

Fig. 3 shows the average delay for both the path from source  $X_1$  to Sink 1 and the path from source  $X_2$  to Sink 2, when assuming the arrival rate of  $\lambda = 1$ . In this case, the duration of the burst sent by each path is

$$\frac{1}{\lambda} \times (N - 1) = 1 \times 15 = 15 \text{ (RTT)}. \quad (5)$$

Correspondingly, as seen in Fig. 3, at  $p_e = 0$ , the delay of the path from source  $X_1$  to Sink 1 is  $15 + 1 = 16$  RTTs, while the delay of the path from source  $X_2$  to Sink 2 is  $15 + 2 = 17$  RTTs. From Fig. 3, we can see that, as the PER increases, the delay of both paths with any considered ARQ schemes increases. For each of the ARQ schemes, the path from source  $X_1$  to Sink 1 has a better delay performance than the path from source

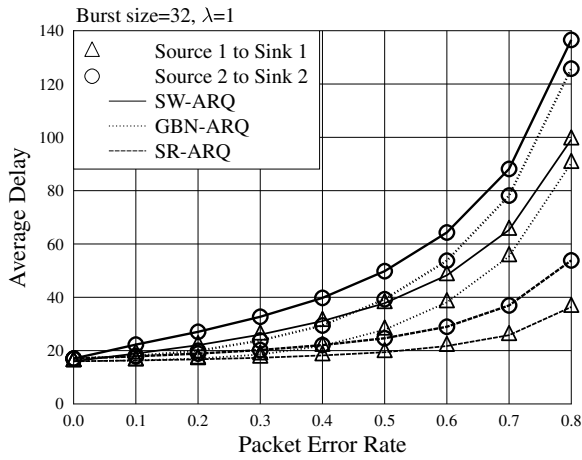


Fig. 3. Average delay versus packet error rate performance of the Butterfly network with different ARQ schemes.

$X_2$  to Sink 2. Among all the three ARQ schemes, the SR-ARQ scheme has the best delay performance and the SW-ARQ scheme has the worst delay performance.

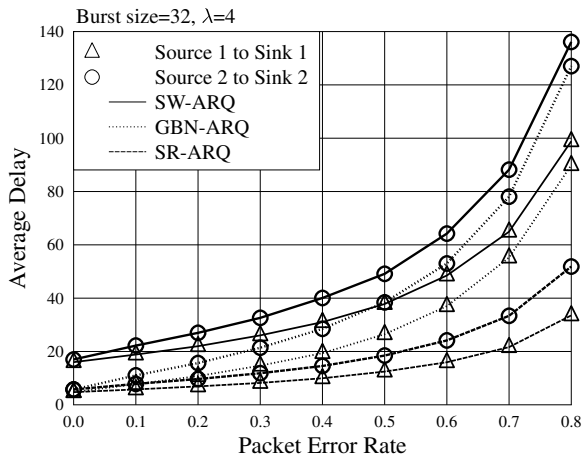


Fig. 4. Average delay versus packet error rate performance of the Butterfly network with different ARQ schemes.

Finally, in Fig. 4, we illustrate the average delay versus PER performance, when the packet arrival rate is  $\lambda = 4$ . In this case, for the SW-ARQ, since the transmitters can only send one packet per RTT, therefore, at  $p_e = 0$ , the total time of  $(16 - 1) \div 1 = 15$  RTTs is required to send one burst. By contrast, for the GBN-ARQ and SR-ARQ schemes, the transmitters are capable of sending 4 packets per RTT. Therefore, the time required to send one burst is  $(16 - 1) \div 4 = 3.75$  RTTs. Additionally, for the SW-ARQ scheme, the delay at  $p_e = 0$  of the path from source  $X_1$  to Sink 1 is  $15 + 1 = 16$  RTTs, while the delay of the path from source  $X_2$  to Sink 2 is  $15 + 2 = 17$  RTTs. At  $p_e = 0$ , for the GBN-ARQ and SR-ARQ schemes, the delay of the path from source  $X_1$  to Sink 1 is  $3.75 + 1 = 4.75$  RTTs, while the delay of the path from source  $X_2$  to Sink 2 is  $3.75 + 2 = 5.75$  RTTs. Consequently, as

seen in Fig. 4, there is a gap at  $p_e = 0$  between the delay of the SW-ARQ scheme and that of the other two ARQ schemes. The other observations in Fig. 4 are similar as that in Fig. 3.

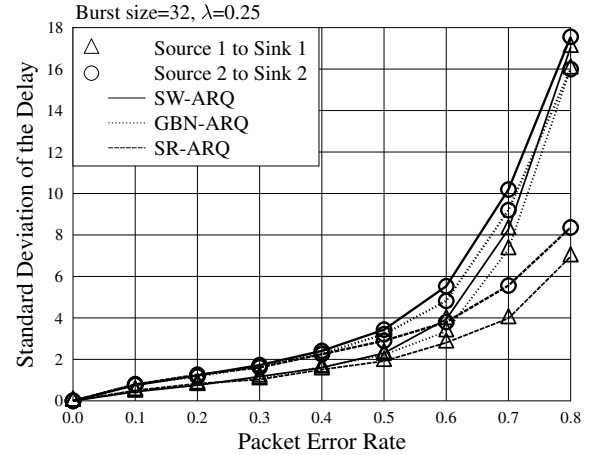


Fig. 5. Standard deviation of burst delay for the Butterfly network with different types of ARQ schemes.

In Figs. 5 - 7 the SD of delay is addressed. Fig. 5 shows the SD of delay versus PER performance of both the path from source  $X_1$  to Sink 1 and the path from source  $X_2$  to Sink 2, when  $\lambda = 0.25$ . When  $p_e = 0$ , the SD of delay is equal to zero for both the paths in the context of all the three ARQ schemes. As seen in Fig. 5, all the three ARQ schemes yield a similar SD of delay, when the PER is relatively low, such as, when  $p_e \leq 0.4$ . When  $p_e > 0.4$ , the SD of delay increases significantly, as  $p_e$  increases. In this region, for any of the two paths, the SW-ARQ scheme generates the highest SD of delay, while the SR-ARQ scheme yields the lowest SD of delay. Furthermore, it can be seen in Fig. 5 that, for any of the two paths and for any a given ARQ scheme, a higher PER results in a higher SD of delay within all the considered range of the PER. Finally, the SD of the path from source  $X_2$  to Sink 2 is higher than that of the path from source  $X_1$  to Sink 1 for each of the ARQ schemes, if the SD values are different.

Fig. 6 illustrates the SD of delay versus PER performance for both the path from source  $X_1$  to Sink 1 and the path from source  $X_2$  to Sink 2, when assuming  $\lambda = 1$ . When  $p_e < 0.55$ , the SW-ARQ scheme generates the highest SD of delay among the three ARQ schemes considered, while the SD values are the same for both the paths of the Butterfly network. For the SW-ARQ scheme, in the region of  $p_e = 0.62 \sim 0.75$ , the SD of the path from source  $X_1$  to Sink 1 is higher than that of the path from source  $X_2$  to Sink 2. In the rest of the range of  $p_e = 0.55 \sim 0.8$ , the SD of the path from source  $X_2$  to Sink 2 is higher than that of the path from source  $X_1$  to Sink 1. Furthermore, as shown in Fig. 6, the SD values attained by the GBN-ARQ scheme are similar to those attained by the SR-ARQ scheme when  $p_e \leq 0.1$ . They then starts increasing significantly from  $p_e = 0.1$  and converge to those attained by the SW-ARQ scheme at  $p_e = 0.5$ . For the GBN-ARQ scheme, in the region of  $p_e = 0 \sim 0.5$ , the SD of the path from source  $X_2$  to Sink 2 is higher than that of

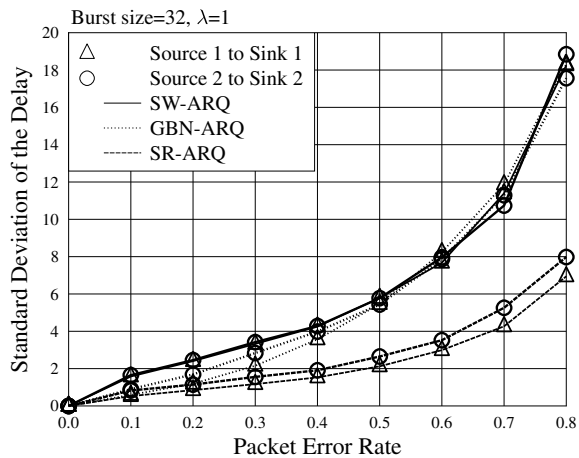


Fig. 6. Standard deviation of burst delay for the Butterfly network with different types of ARQ schemes.

the path from source  $X_1$  to Sink 1, while when  $p_e > 0.5$ , the SD of the path from source  $X_1$  to Sink 1 exceeds that of the path from source  $X_2$  to Sink 2. Finally, as seen in Fig. 6, the SR-ARQ scheme attains the lowest SD of delay among the ARQ schemes considered within the whole range of PER considered. For the SR-ARQ scheme, the SD of the path from source  $X_2$  to Sink 2 is always higher than that of the path from source  $X_1$  to Sink 1 except at  $p_e = 0$ .

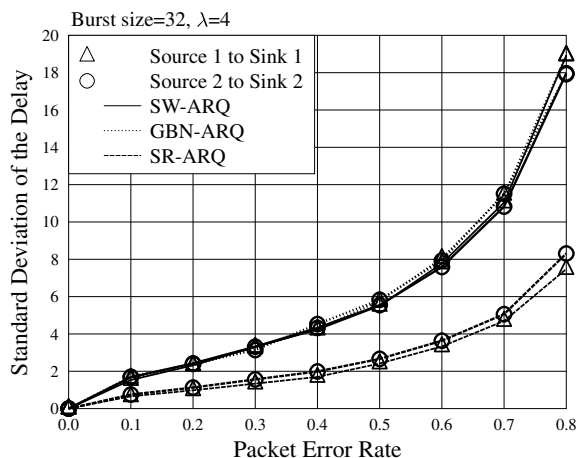


Fig. 7. Standard deviation of burst delay for the Butterfly network with different types of ARQ schemes.

Fig. 7 shows the SD of delay versus PER performance, when assuming  $\lambda = 4$ . Within the whole range of PER considered, the SW-ARQ and the GBN-ARQ schemes yield very similar SD, with the SD of the path from source  $X_1$  to Sink 1 very close to that of the path from source  $X_2$  to Sink 2. Finally, as seen in Fig. 7, the SR-ARQ scheme attains the lowest SD among the three ARQ schemes within the whole range of PER considered. For the SR-ARQ scheme, the SD of the path from source  $X_2$  to Sink 2 is always higher than that of the path from source  $X_1$  to Sink 1 except at  $p_e = 0$ .

## V. CONCLUSIONS

The average delay of packet bursts and the SD of the packet delay in the Butterfly networks have been investigated and compared, when the SW-ARQ, GBN-ARQ and the SR-ARQ schemes are invoked for data transmission. From our studies, we find that the delay performance of the Butterfly network is determined by the path containing the coding node. When comparing the three types of ARQ schemes, the SW-ARQ scheme is found to yield the highest delay and also the highest SD of delay at a given PER. Among the three ARQ schemes, given a PER, the SR-ARQ scheme is capable of attaining the lowest delay and also the lowest SD of delay. Furthermore, as the arrival rate of packets increases, the SD of delay of the GBN-ARQ scheme converges to that of the SW-ARQ scheme, whilst the SD of delay of the path from the source  $X_2$  to Sink 2 converges to that of the path from source  $X_1$  to Sink 1. Therefore, we can conclude that, for the Butterfly network, the SR-ARQ scheme may be one of the appropriate schemes for supporting data transmission. However, we should realize that the SR-ARQ scheme requires the highest implementation complexity, while the SW-ARQ scheme has the lowest complexity, among the three ARQ schemes considered. Additionally, for a given ARQ scheme operated at a given PER, we find that the burst delay and the delay's SD from source  $X_1$  to Sink 1 are usually higher than that from source  $X_2$  to Sink 2.

## REFERENCES

- [1] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.
- [2] D. S. Lun, M. Medard, and M. Effros, "On coding for reliable communication over packet networks," in *Proc. 42nd Annual Allerton Conference on Communication, Control, and Computing*, Sept./Oct. 2004.
- [3] S. Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.
- [4] D. S. Lun, M. Medard, R. Koetter, and M. Effros, "Further results on coding for reliable communication over packet networks," in *Proc. International Symposium on Information Theory '05*, 4–9 Sept. 2005, pp. 1848–1852.
- [5] S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed. Upper Saddle River, New Jersey: Prentice Hall, 2004.
- [6] J. Davidson, J. Peters, and B. Grace, *Voice over IP fundamentals*. Cisco Press, 2000.
- [7] C. Fragouli, D. Lun, M. Medard, and P. Pakzad, "On feedback for network coding," in *Proc. 41st Annual Conference on Information Sciences and Systems 2007*, 14–16 March 2007, pp. 248–252.
- [8] J. Kumar Sundararajan, D. Shah, and M. Medard, "ARQ for network coding," in *Proc. IEEE International Symposium on Information Theory 2008*, 6–11 July 2008, pp. 1651–1655.
- [9] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, "Hybrid ARQ-random network coding for wireless media streaming," in *Proc. Second International Conference on Communications and Electronics 2008*, 4–6 June 2008, pp. 115–120.
- [10] W. Stallings, *Data and Computer Communications*, 8th ed. New Jersey: Prentice Hall, 2006.
- [11] G. ITU-T and I. Switzerland, "711: Pulse code modulation (PCM) of voice frequencies," *ITU-T Recommendation G*, vol. 711.
- [12] W. Jiang and H. Schulzrinne, "Analysis of on-off patterns in VoIP and their effect on voice traffic aggregation," 2000.
- [13] P. T. Brandy, "Model for generating on-off speech patterns in two-way conversation," *The Journal of the Acoustical Society of America*, vol. 46, no. 1A, pp. 109–109, 1969.
- [14] D. P. Bertsekas and R. G. Gallager, *Data Networks*. Upper Saddle River, New Jersey: Prentice Hall, 1992.