Chapter 15
Performance Analysis of a Decentralized Content Delivery System with FEC Recovery

Kenji Kirihara, Hiroyuki Masuyama, Shoji Kasahara, and Yutaka Takahashi

Abstract This chapter considers the performance of a decentralized content delivery system where video data are simultaneously delivered without duplication by multiple streaming video servers, resulting in a low sending rate per video server. Focusing on a multiple-server video streaming service reinforced by forward error correction (FEC), we model the system as a set of independent GI+M/M/1/K queues, and derive the block-level loss probability. Numerical results show that the decentralized content delivery system with FEC recovery is significantly effective to guarantee video quality even when the background traffic intensity is high.

15.1 Introduction

With the recent advancement of network technologies enabling ultra-high-speed data transmission, video streaming service over the Internet has attracted considerable attention. The Internet, however, is a best-effort network, and thus the quality of service (QoS) for video streaming is not strictly guaranteed due to packet loss and/or delay.

In order to enhance the resilience to packet loss, a number of approaches have been proposed and studied. Among them automatic repeat request (ARQ) and forward error correction (FEC) are commonly deployed for loss recovery. ARQ is an acknowledgment-based error recovery, in which lost data packets are retransmitted reactively by the sender host. ARQ is an efficient resilience mechanism for packet loss if the round-trip time between the sender and receiver hosts is significantly small. However, ARQ is not suitable for a network with a large round-trip time, resulting in a larger end-to-end delay caused by multiplicative transmissions.

K. Kirihara, H. Masuyama, S. Kasahara and Y. Takahashi
Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan
e-mail: kirihara@sys.i.kyoto-u.ac.jp; {masuyama, shoji, takahashi}@i.kyoto-u.ac.jp

W. Yue et al. (eds.), *Advances in Queueing Theory and Network Applications,*

On the other hand, FEC is a one-way recovery technique based on open-loop error control. FEC generates redundant data from original data, and both original and redundant data are transmitted to a destination. If the amount of lost data is less than or equal to a prespecified threshold, the lost data can be reconstructed. In this chapter, we consider a packet-level FEC scheme [1]. Because FEC needs no retransmission, it is suitable for real-time applications with stringent delay constraint such as video streaming. However, FEC does not work well against packet burst loss because the amount of redundant data has to be predetermined with the estimate of the packet loss probability.

An alternative approach to guarantee QoS against packet loss is multiple-sender video streaming [2]. This is a decentralized content delivery scheme in which video data are divided into segments to be simultaneously delivered by multiple streaming sender hosts. Each sending rate per server is significantly smaller than that of a single-sender case, achieving a small overall packet loss probability at the destination. In [2], Nguyen and Zakhor proposed a distributed video streaming protocol consisting of a rate allocation algorithm and a packet partition algorithm. Its performance was investigated by simulation and experiments with a real network. FEC recovery performance has also been studied in the literature [3]–[5], however, little work has been devoted to analyze the compound effect of the decentralized content distribution mechanism and FEC.

With the recent advancement of photonic networking technology such as wavelength division multiplexing (WDM), the bottleneck of data transmission has shifted from backbone networks to access ones (the last-mile bandwidth bottleneck [6]). This implies that edge routers of backbone networks are likely to be the bottleneck of data transmission for real-time applications. In real-time applications such as VoIP and Internet TV, packets are sent to the network at a constant bit rate. Therefore, it is important to consider the case where interarrival times of packets to a bottleneck edge router are almost the same.

In this chapter, we analyze the performance of this decentralized content delivery system by a queueing theoretical approach. We consider a multiple-sender video streaming service, and focus on disjoint parts of multiple routes to the destination. Assuming that there exists a bottleneck router along the disjoint part of each route, we model each bottleneck router as a single-server finite queueing system with both general renewal and Poisson inputs. We derive the block-level loss probability, and investigate the resulting video quality of multiple-sender video streaming with and without FEC. Note that the assumption of the general renewal input for main traffic enables us to describe various arrival processes including constant interarrival times.

The chapter is organized as follows. Section 15.2 describes the analysis model in detail, and derives the block-loss probability. Numerical results are presented in Sect. 15.3, and we conclude this chapter in Sect. 15.4.

15.2 Model and Analysis

We consider a multiple-sender distributed video streaming service. Let S denote the number of video servers. A video dataum is divided into S parts, each of which are simultaneously delivered along with different routes. We assume that a video data frame consists of D packets. N redundant data packets are generated from the D original data packets, and a set of $M(= D+N)$ packets is called a block. If the number of lost packets among the M packets is less than or equal to N, the original data packets can be reproduced completely regardless of the lost packets. On the other hand, if the number of lost packets among the M packets is greater than N, the original data packets cannot be recovered. We call this event a block loss.

Video streaming service is supported by S servers. We divide M packets per frame into S groups: group l $(l = 1,\ldots,S)$ with $M^{(l)}$ packets. Note that $\sum_{l=1}^{S} M^{(l)} = M$. Server l manages the $M^{(l)}$ packets in group l and sends those packets to the destination. Note that we have S streaming routes for a video service. Suppose that there exists a bottleneck router along each route and that packet loss occurs independently at each bottleneck router.

We model each bottleneck router as a single-server queueing system with a finite buffer that is fed by two independent input processes. In the following, the packet flow sent from a streaming server is called the main traffic, and the other packet flow the background traffic. The interarrival times of packets in the main traffic sent from server l are independently identically distributed (i.i.d.) with a general distribution $G^{(l)}(x)$. The packet arrivals in the background traffic form a Poisson process with rate $\lambda^{(l)}$. The capacity of the system with server l is $K^{(l)}$. Note that the bottleneck router forwards not only the packets from the video server but also the packets belonging to the background traffic. Therefore, it is natural to assume that the packet size is not the same. Then, we assume that the service time of a packet is exponentially distributed with rate $\mu^{(l)}$. From the above assumptions, we have a GI+M/M/1/K-type queueing model for each bottleneck router.

We derive the block-loss probability that a block is not eventually retrieved at the destination. Let $p^{(l)}(k \mid M^{(l)})$ $(l = 1,\ldots,S)$ denote the probability that k $(k = 0,1,\ldots,M^{(l)})$ packets out of $M^{(l)}$ packets sent from server l are lost. We can compute $p^{(l)}(k \mid M^{(l)})$ from the analytical result in [5]. (The derivation of $p^{(l)}(k \mid M^{(l)})$ is summarized in the appendix.) Noting that original data packets for a video frame can be recovered if the number of lost packets is less than or equal to N, the block-loss probability P_{Loss} is given by

$$P_{Loss} = 1 - \sum_{n=0}^{N} \sum_{k_1+\cdots+k_S=n} p^{(1)}(k_1 \mid M^{(1)}) p^{(2)}(k_2 \mid M^{(2)}) \cdots p^{(S)}(k_S \mid M^{(S)}). \quad (15.1)$$

15.3 Numerical Results

We assume that the transmission rate of a video streaming service for the single-server case is 10 Mbps, and that the output transmission speed of bottleneck routers is 100 Mbps. It is supposed that the video frame rate is 30 [frame/s], and that a frame has the same number of packets as that of a block. The packet size is 1250 bytes. Thus a block has $D = 34$ original data packets, and the packet service rate of packets at bottleneck routers is $\mu = 1 \times 10^4$ [packet/s].

For the multiple-sender case, we assume that the number of video servers is two and that $M^{(1)} = M^{(2)} = (34+N)/2$. The number of FEC redundant packets N is set to 0, 2, and 4. The packet interarrival time of main traffic from each video server is constant. The system capacities $K^{(l)}$ are assumed to be the same and set to $K^{(l)} = K = 10$ and 100. In what follows, we assume that the flow rates of background traffic are equal at both of the bottleneck routers. Note that when N FEC redundant packets are added to D original data packets, the resulting packet transmission rate becomes $(D+N)/D$ times larger than the original one.

The block-loss probability for the multiple-sender case is calculated by (15.1). We also calculate the block-loss probability for the single-sender case using the result in [5]. We define the FEC redundancy as N/D.

15.3.1 Impact of Background Traffic

In this subsection, we investigate how the bandwidth of background traffic affects the block-loss probability.

Figures 15.1 and 15.2 show the block-loss probability against the bandwidth of background traffic in the cases $K = 10$ and 100, respectively. We observe from

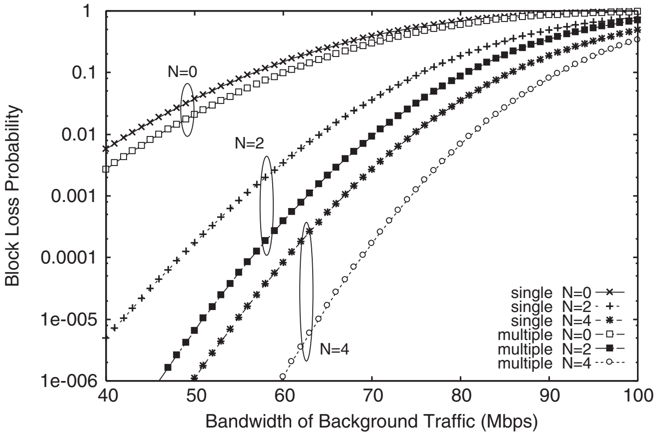

Fig. 15.1 Block-loss probability versus bandwidth of background traffic ($S = 2$, $K = 10$, $D = 34$).

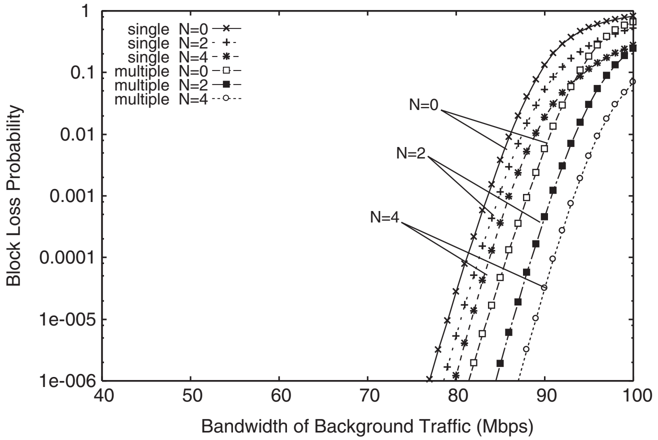

Fig. 15.2 Block-loss probability versus bandwidth of background traffic ($S = 2, K = 100, D = 34$).

Figs. 15.1 and 15.2 that the block-loss probability increases monotonically when the bandwidth of the background traffic is large, as expected. Owing to the multiple-sender effect, the block-loss probability is further improved for the same number of redundant packets as the single-sender case. We also observe the decrease in the block-loss probability when the number of FEC redundant packets increases, and that the block-loss probability is effectively reduced by FEC redundant packets in the system with a small capacity. When the system capacity is small, a packet-loss event frequently occurs, making the packet-loss process random. Because FEC works well against random packet-loss processes, the block-loss probability is greatly improved by FEC when the system capacity is small.

Next we investigate how the bandwidth of background traffic affects the minimum FEC redundancy. Here, the minimum FEC redundancy is such that the block-loss probability is smaller than a prespecified value $P_{Loss}^{(\alpha)}$. Figures 15.3 and 15.4 illustrate the minimum FEC redundancy against the bandwidth of background traffic in cases of K=10 and K=100, respectively. For each value of K, we calculated the minimum FEC redundancy for the cases $P_{Loss}^{(\alpha)} = 10^{-2}, 10^{-3}$ and 10^{-4}.

It is observed from Fig. 15.3 that the FEC redundancy in the multiple-sender case is smaller than that in the single-sender case when $P_{Loss}^{(\alpha)}$ is fixed. This is due to a small packet-loss probability in the multiple-sender case. In Fig. 15.4, the minimum FEC redundancy remains zero at 80 Mbps background traffic in all the cases, because the packet-loss events hardly occur in a system with large capacity. When the bandwidth of background traffic is greater than 80 Mbps, the minimum FEC redundancy increases rapidly. This tendency of the minimum FEC redundancy is the same as in Fig. 15.3. Comparing Fig. 15.3 with Fig. 15.4, FEC is effective in a wide range of background traffic when the system capacity is small.

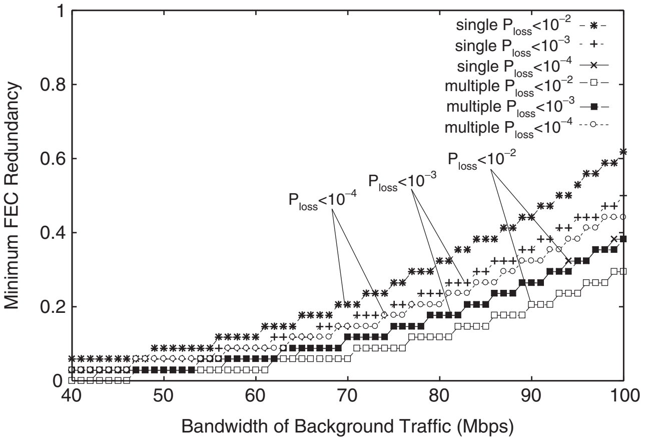

Fig. 15.3 Minimum FEC redundancy versus bandwidth of background traffic ($S = 2$, $K = 10$).

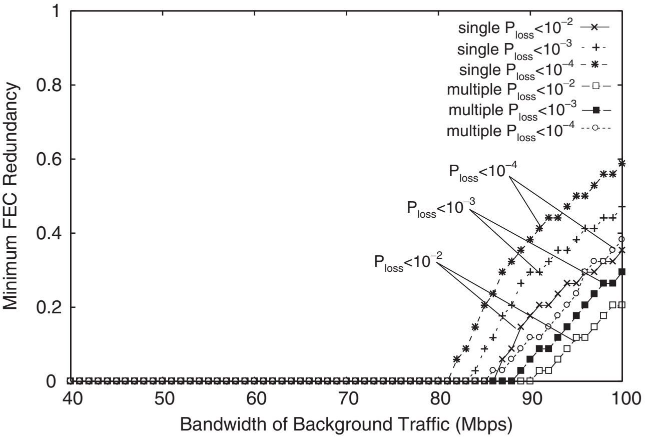

Fig. 15.4 Minimum FEC redundancy versus bandwidth of background traffic ($S = 2$, $K = 100$).

15.3.2 Impact of Service Rate at Bottleneck Router

In this subsection, we investigate how the output transmission speed of the bottleneck router affects the block-loss probability and the minimum FEC redundancy.

Figures 15.5 and 15.6 show the block-loss probability against the transmission speed in the cases $K = 10$ and 100, respectively. Note that when the transmission speed is η bps, the corresponding service rate of a packet at the bottleneck router μ is equal to $\eta \times 10^4$ [packet/s]. The bandwidth of background traffic is set to 50 Mbps.

We observe from Fig. 15.5 that the block-loss probability decreases monotonically and gradually when the transmission speed increases. It is also observed that the decentralized content delivery system is greatly effective for the block-loss

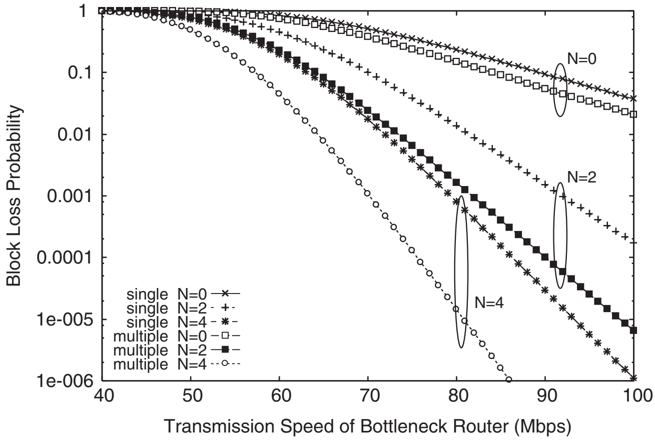

Fig. 15.5 Block-loss probability versus transmission speed ($S = 2$, $K = 10$, $D = 34$).

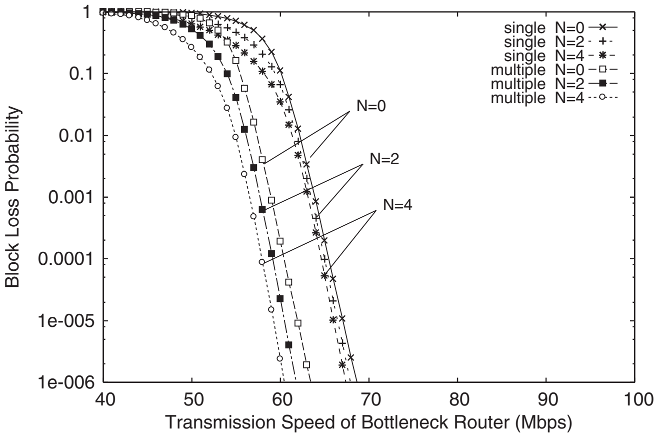

Fig. 15.6 Block-loss probability versus transmission speed ($S = 2$, $K = 100$, $D = 34$).

probability as the FEC redundancy increases. In addition, the block-loss probability for the multiple-sender case is significantly smaller than that for the single-sender case.

It is observed from Fig. 15.6 that the block-loss probability for $K = 100$ exhibits the same tendency as that in Fig. 15.5. Note that the block-loss probability is greatly improved by a high transmission speed, rather than FEC and the decentralized content distribution mechanism.

Figures 15.7 and 15.8 illustrate the minimum FEC redundancy against the transmission speed of the bottleneck router in the cases $K = 10$ and 100, respectively. The minimum FEC redundancy was calculated for $P_{Loss}^{(\alpha)} = 10^{-2}$, 10^{-3}, and 10^{-4}.

In Fig. 15.7, the minimum FEC redundancy in the multiple-sender case is smaller than that in the single-sender case when $P_{Loss}^{(\alpha)}$ is fixed. In addition, the minimum

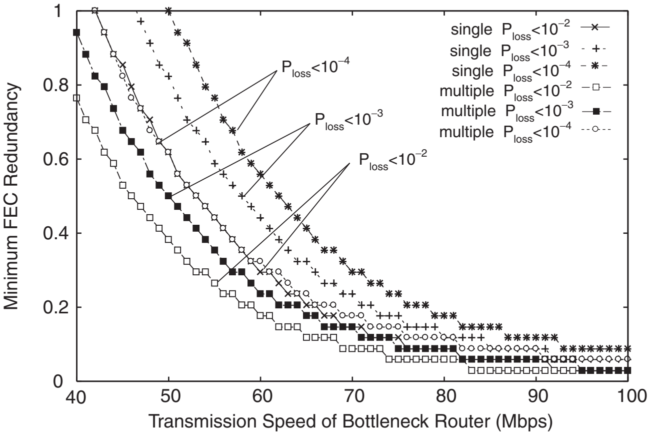

Fig. 15.7 Minimum FEC redundancy versus transmission speed ($S = 2$, $K = 10$).

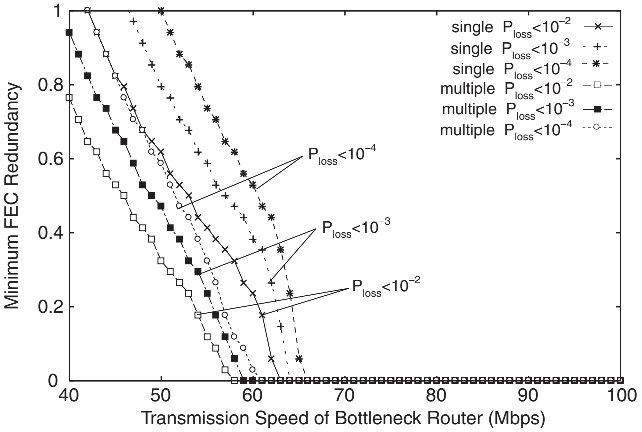

Fig. 15.8 Minimum FEC redundancy versus transmission speed ($S = 2$, $K = 100$).

FEC redundancy gradually decreases when the transmission speed increases. We observe from Fig. 15.8 that the minimum FEC redundancy reaches zero when the transmission speed is larger than around 60 Mbps, and that the other tendency is the same as in Fig. 15.7. These results imply that when the output transmission speed is small, the decentralized content delivery system with FEC recovery is significantly effective for the block-loss probability.

15.3.3 Impact of System Capacity

In this subsection, we investigate the impact of the system capacity on the minimum FEC redundancy. The QoS requirement considered here is $P^{(\alpha)}_{Loss} = 10^{-2}$, 10^{-3}, and 10^{-4}. With each $P^{(\alpha)}_{Loss}$, we calculated the minimum FEC redundancy in cases of

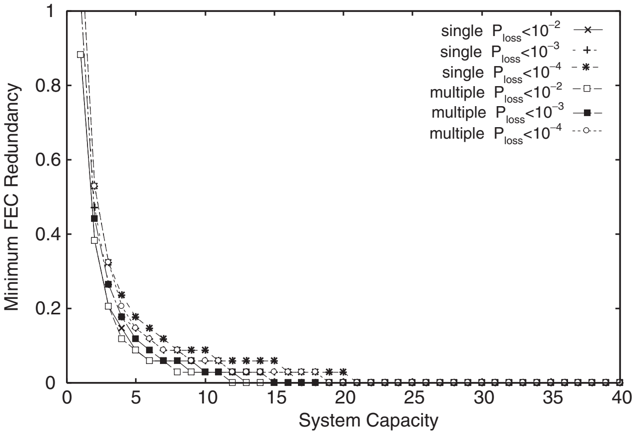

Fig. 15.9 Minimum FEC redundancy versus system capacity ($S = 2$, 50 Mbps background traffic).

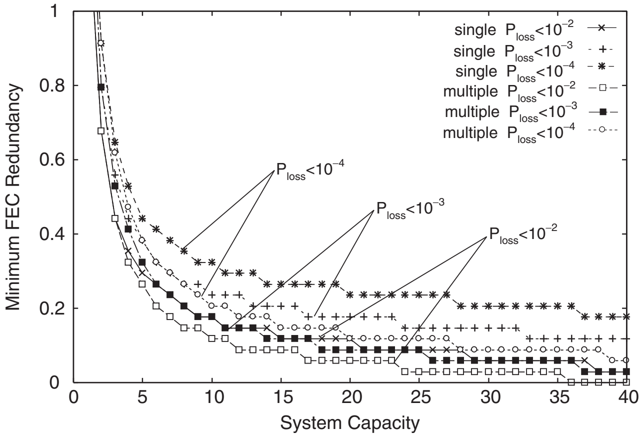

Fig. 15.10 Minimum FEC redundancy versus system capacity ($S = 2$, 80 Mbps background traffic).

50 Mbps and 80 Mbps of the bandwidth of background traffic. It is supposed that the output transmission speed of the bottleneck router is 100 Mbps.

Figure 15.9 shows the minimum FEC redundancy against the system capacity when the bandwidth of background traffic is 50 Mbps. We observe from Fig. 15.9 that the minimum FEC redundancy decreases rapidly when the system capacity is about 5, and that for each P_{Loss} the minimum FEC redundancy reaches zero when the system capacity is greater than 20. Note that the variation of the minimum FEC redundancy is small. This implies that the block-loss probability is greatly improved by the system capacity rather than the decentralized content distribution mechanism with FEC recovery.

Figure 15.10 shows the minimum FEC redundancy when the bandwidth of background traffic is 80 Mbps. In Fig. 15.10, the minimum FEC redundancy decreases monotonically as the system capacity is large. This tendency is the same as in Fig. 15.9. Comparing Fig. 15.9 with Fig. 15.10, there is a difference between the minimum FEC redundancy in the multiple-sender case and in the single-sender case. A remarkable point of Fig. 15.10 is that the minimum FEC redundancy is likely to remain constant when the system capacity increases. This implies that the decentralized content delivery system with FEC recovery is more effective than enriching system capacity when the system is overloaded.

15.3.4 Impact of Number of Video Servers

Finally, we investigate how the number of video servers improves the video QoS. We set $N = 4$ and hence the number of packets in a block M is 38. We consider four cases of $S = 1$, 2, 3, and 4. Table 15.1 shows the parameter values of $M^{(l)}$s for each S.

Figure 15.11 represents the block-loss probability against the bandwidth of background traffic for $K = 10$ and 100. We assume that all the background traffic intensities at bottleneck routers are the same. This scenario can be regarded as the worst case for multiple-sender transmission.

Table 15.1 Number of packets within each group.

S	Parameters	Values
1	M	38
2	$(M^{(1)}, M^{(2)})$	$(19, 19)$
3	$(M^{(1)}, M^{(2)}, M^{(3)})$	$(13, 13, 12)$
4	$(M^{(1)}, M^{(2)}, M^{(3)}, M^{(4)})$	$(10, 10, 9, 9)$

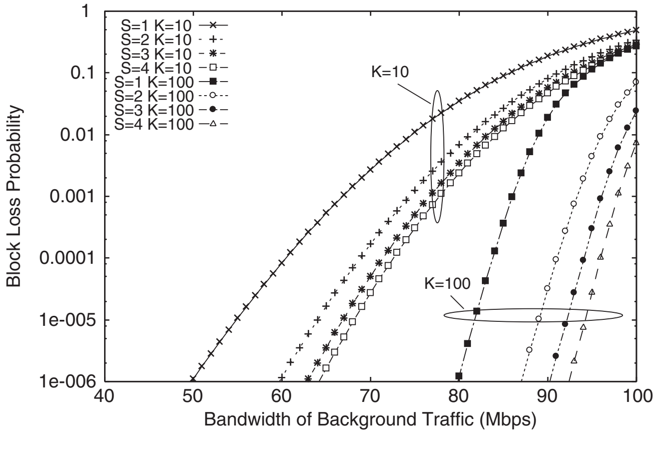

Fig. 15.11 Block-loss probability versus bandwidth of background traffic ($D = 34$, $N = 4$).

It is observed from the figure that for both Ks, the block-loss probability is improved with the increase in the number of video servers, as expected. When $K = 10$, the block-loss probability for $S = 2$ is significantly smaller than that for the single-server case. However, the block-loss probabilities for $S = 3$ and 4 are not greatly improved. Note that $M = 38$ corresponds to 11.4 Mbps of the video sending rate for the single-server case. Roughly speaking, the video sending rate per server is 5.7 Mbps for $S = 2$, 3.8 Mbps for $S = 3$, and 2.9 Mbps for $S = 4$. That is, the resulting video sending rates per server are relatively small in comparison with the background traffic intensity. When $K = 100$, on the other hand, the block-loss probability is significantly small and greatly improved with the increase in the number of video servers. This result suggests that the decentralized content delivery system supported by multiple servers can guarantee video QoS effectively when the network is heavily congested.

Figure 15.12 shows the minimum FEC redundancy against the system capacity per router when the bandwidth of background traffic is 80 Mbps. The QoS requirement considered here is $P_{loss}^{(\alpha)} = 10^{-4}$. In Fig. 15.12, the minimum FEC redundancy for $S = 1$ is the largest, and the minimum FEC redundancy decreases with the increase in S. A remarkable point here is that the minimum FEC redundancies for $S = 2$, 3, and 4 are almost the same, regardless of the system size. Note that the bandwidth of background traffic is 80 Mbps. Because the link capacity is set to 100 Mbps, the overall traffic intensity at each bottleneck router is more than 0.8; that is, the system is heavily utilized. Under such a heavy loaded condition, the increase in the system capacity is more effective for video streaming than increasing the number of video servers. This result also implies that if the buffer size of bottleneck routers is large, video QoS can be guaranteed with two video servers.

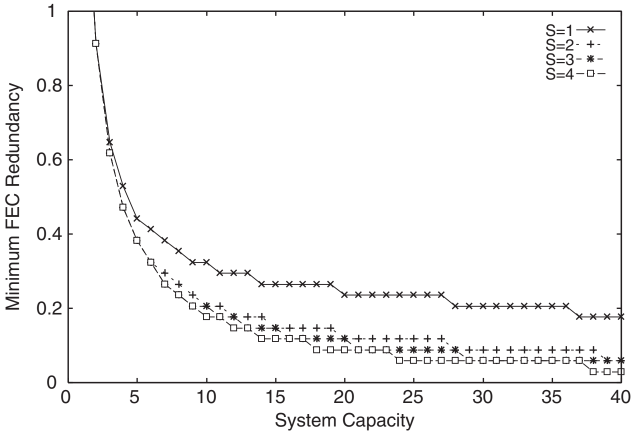

Fig. 15.12 Minimum FEC redundancy versus system capacity (80 Mbps background traffic, $P_{loss} < 10^{-4}$).

15.4 Conclusions

This chapter analyzed the performance of the decentralized content delivery system with FEC recovery. We focused on a multiple-sender video streaming service and modeled it as a set of GI+M/M/1/K queues, deriving the block-level loss probability. Numerical results showed that decentralized content delivery in cooperation with FEC recovery is significantly effective for preserving video quality even when the background traffic intensity is high. In particular, when the system capacity is small and the network is overloaded, multiple-sender video streaming succeeds in guaranteeing video QoS with less FEC redundancy than the single-server case. A remarkable point is that two video servers are enough to guarantee video QoS even when the network is heavily utilized. In this overloaded condition, enriching router buffers is more effective than increasing the number of video servers. In general, increasing the number of video servers causes a large control overhead of video-content management. The fact that a few video servers are enough to guarantee video QoS is significant from the viewpoint of video-content management.

Appendix: Derivation of Probability $\mathbf{p^{(l)}(k \mid M^{(l)})}$

This appendix summarizes the derivation of the probability $p^{(l)}(k \mid M^{(l)})$ ($l = 1,\dots,S$). For details, see [5], where $\mathbf{p}_M(k)\mathbf{e}$ corresponds to $p^{(l)}(k \mid M^{(l)})$. For simplicity, we omit superscript "$^{(l)}$" in this appendix. Thus, for example, although we write λ, μ, and M for $\lambda^{(l)}$, $\mu^{(l)}$, and $M^{(l)}$, respectively, λs, μs, and Ms herein are different from original λs, μs, and Ms in the preceding sections.

We first consider the stationary queue length distribution immediately before an arrival from main traffic in the GI+M/M/1/K queue, which models a bottleneck router. Let T_m ($m = 0, \pm 1, \pm 2, \dots$) denote the arrival epoch of the mth packet from main traffic. We then assume that the system reaches steady state at time T_0. Let L_m^- ($m = 0, \pm 1, \pm 2, \dots$) denote the number of packets in the system immediately before time T_m. Note that during each interval between arrivals (T_m, T_{m+1}), the behavior of the queueing process is stochastically the same as that of the M/M/1/K queue with arrival rate λ and service rate μ. Thus $\{L_m^-; m = 0, \pm 1, \pm 2, \dots\}$ is a Markov chain whose transition probability matrix $\mathit{\Pi}$ is given by

$$\mathit{\Pi} = \mathit{\Lambda} \int_0^\infty \exp(\mathbf{Q}x) dG(x), \tag{15.2}$$

where $G(x)$ denotes the distribution of interarrival times of packets from main traffic, and where $\mathit{\Lambda}$ and $\mathbf{Q}$ denote $(K+1) \times (K+1)$ matrices that are given by

$$\Lambda = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ddots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix},$$

$$\mathbf{Q} = \begin{pmatrix} -\lambda & \lambda & 0 & \dots & 0 & 0 \\ \mu & -(\lambda+\mu) & \lambda & \ddots & \vdots & \vdots \\ 0 & \mu & -(\lambda+\mu) & \ddots & 0 & 0 \\ 0 & 0 & \mu & \ddots & \lambda & 0 \\ \vdots & \vdots & \ddots & \ddots & -(\lambda+\mu) & \lambda \\ 0 & 0 & 0 & \ddots & \mu & -\mu \end{pmatrix}.$$

We define π^- as a $1 \times (K+1)$ vector whose jth ($j = 0,1,\dots,K$) element π_j^- represents $\Pr[L_1^- = j]$. We then have

$$\pi^- \Pi = \pi^-, \qquad \pi^- \mathbf{e} = 1,$$

where $\mathbf{e}$ denotes a column vector of ones with appropriate dimension.

Next in order to derive $p(k \mid M)$, we consider an arbitrary group that consists of M packets sent from a server. We assume that the M packets of the group arrive at the system at times T_1 through T_M. We then call the packet arriving at time T_m ($m = 1,2,\dots,M$) packet m. Let L_m denote the number of packets in the system at time T_m. Let N_m ($m = 1,2,\dots,M$) denote the number of lost packets among packets 1 through m at time T_m. We define $\mathbf{p}_m(k)$ ($m = 1,2,\dots,M$, $k = 0,1,\dots,M$) as a $1 \times K$ vector whose jth ($j = 1,2,\dots,K$) element $p_{m,j}(k)$ is given by

$$p_{m,j}(k) = \Pr[N_m = k, L_m = j].$$

Because $N_m \leq m$ for all $m = 1,2,\dots,M$,

$$\mathbf{p}_m(k) = \mathbf{0}, \quad \text{for all } k = m+1, m+2, \dots, M. \tag{15.3}$$

By using $\mathbf{p}_m(k)$, the probability $p(k \mid M)$ is given by

$$p(k \mid M) = \mathbf{p}_M(k)\mathbf{e}. \tag{15.4}$$

In what follows, we discuss the $\mathbf{p}_m(k)$s ($m = 1,2,\dots,M$, $k = 0,1,\dots,M$). Note that if $L_1^- < K$, packet 1 can join the queue and hence $L_1 = L_1^- + 1$. Note also that if $L_1^- = K$, packet 1 is lost and $L_1 = K$. Thus $p_{1,j}(0)$ and $p_{1,j}(1)$ ($j = 1,2,\dots,K$) are given by

$$p_{1,j}(0) = \Pr[L_1^- < K, L_1^- = j-1] = \pi_{j-1}^-, \qquad j = 1,2,\dots,K, \tag{15.5}$$

$$p_{1,j}(1) = \begin{cases} 0, & j = 1,2,\dots,K-1 \\ \pi_K^-, & j = K, \end{cases} \tag{15.6}$$

respectively, or equivalently,

$$\mathbf{p}_1(0) = (\pi_0^-, \pi_1^-, \dots, \pi_{K-1}^-), \qquad \mathbf{p}_1(1) = (0,0,\dots,0,\pi_K^-).$$

We now define $\mathbf{A}(\nu)$ $(\nu = 0,1)$ as a $K \times K$ matrix whose (i,j)th element $A_{i,j}(\nu)$ $(i,j = 1,2,\dots,K)$ is given by

$$A_{i,j}(\nu) = \Pr[\Theta_m = \nu, L_m = j \mid L_{m-1} = i],$$

where $\Theta_m = 1$ if packet m is lost, and otherwise $\Theta_m = 0$. It is easy to see that for $m = 2,3,\dots,M$,

$$\mathbf{p}_m(0) = \mathbf{p}_{m-1}(0)\mathbf{A}(0), \tag{15.7}$$

$$\mathbf{p}_m(k) = \mathbf{p}_{m-1}(k-1)\mathbf{A}(1) + \mathbf{p}_{m-1}(k)\mathbf{A}(0), \qquad k = 1,2,\dots,M. \tag{15.8}$$

Finally, we consider $\mathbf{A}(\nu)$ $(\nu = 0,1)$. If $L_m^- < K$, $\Theta_m = 0$ and $L_m = L_m^- + 1$. Thus we have

$$A_{i,j}(0) = \Pr[L_m^- < K, L_m^- = j-1 \mid L_{m-1} = i] = \Gamma_{i,j-1}, \qquad i,j = 1,2,\dots,K, \tag{15.9}$$

where $\Gamma_{i,j}$ $(i,j = 0,1,\dots,K)$ denotes the (i,j)th element of $\Gamma = \int_0^\infty \exp(\mathbf{Q}x)dG(x)$. Note here that $\Pi = \Lambda\Gamma$ (see (15.2)). Furthermore, because $\{\Theta_m = 1\}$ is equivalent to $\{L_m = L_m^- = K\}$,

$$A_{i,j}(1) = \begin{cases} 0, & j = 1,2,\dots,K-1, \\ \Gamma_{i,K}, & j = K, \end{cases} \qquad i = 1,2,\dots,K. \tag{15.10}$$

In matrix notation, (15.9) and (15.10) are written as follows:

$$\mathbf{A}(0) = \begin{pmatrix} \Gamma_{1,0} & \Gamma_{1,1} & \cdots & \Gamma_{1,K-1} \\ \Gamma_{2,0} & \Gamma_{2,1} & \cdots & \Gamma_{2,K-1} \\ \vdots & \vdots & \ddots & \vdots \\ \Gamma_{K,0} & \Gamma_{K,1} & \cdots & \Gamma_{K,K-1} \end{pmatrix}, \qquad \mathbf{A}(1) = \begin{pmatrix} 0 & \cdots & 0 & \Gamma_{1,K} \\ 0 & \cdots & 0 & \Gamma_{2,K} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & \Gamma_{K,K} \end{pmatrix}.$$

References

1. N. Shacham and P. Pckenney, Packet recovery in high-speed networks using coding and buffer management, in *Proc. IEEE INFOCOM*, vol. 1, pp. 124–131, 1990.
2. T. Nguyen and A. Zakhor, Multiple sender distributed video streaming, *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 315–326, 2004.

3. E. Altman and A. Jean-Marie, Loss probabilities for messages with redundant packets feeding a finite buffer, *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 5, pp. 778–787, 1998.
4. I. Cidon, A. Khamisy, and M. Sidi, Analysis of packet loss processes in high-speed networks, *IEEE Transactions on Information Theory*, vol. 39, no. 1, pp. 98–108, 1993.
5. S. Muraoka, H. Masuyama, S. Kasahara, and Y. Takahashi, Performance analysis of FEC recovery using finite-buffer queueing system with general renewal and Poisson inputs, *Managing Traffic Performance in Converged Networks*, LNCS4516, Springer, pp. 707–718, 2007.
6. P.E. Green, Fiber to the home: The next big broadband thing, *IEEE Communications Magazine*, vol. 42, no. 9, pp. 100–106, 2004.