Adaptive Rate Control with Dynamic FEC for Real-Time DV Streaming

Kazuhisa Matsuzono
Graduate School of Media and Governance, Keio University
252-8520 Kanagawa, Japan
kazuhisa@sfc.wide.ad.jp

Kazunori Sugiura
Research Institute for Digital Media and Content
Keio University, 108-072 Tokyo, Japan
uhyo@sfc.wide.ad.jp

Hitoshi Asaeda
Graduate School of Media and Governance, Keio University
252-8520 Kanagawa, Japan
asaeda@sfc.wide.ad.jp

***Abstract*—For higher streaming quality, a data sender adjusts the data transmission rate according to the network condition between the sender and receiver. The sender and the receiver exchange information about the network condition to decide the appropriate transmission rate. However, monitoring each data flow in real time is difficult, and controlling the sender to adjust the best quality for each receiver in a heterogeneous environment is a real challenge. In this paper, we study packet loss patterns and FEC recovery rate upon data transmission in a congested network, and define an adaptive rate control mechanism that dynamically adjusts the transmission rate and the FEC encoding rate. We then show our adaptive DV Transmission System (DVTS) that supports appropriate rate control for provisioning multimedia streaming with the best possible quality, and evaluate the system on top of our testbed network.**

I. Introduction

The Digital Video (DV) format gives consumers the ability to edit video with almost professional quality without adding the special hardware requirements to consumer electronics or PCs. Many consumer products or equipments support the DV format, helping it fulfill its promise. Due to dissemination of high speed DSL and FTTH, high quality DV streaming over IP will become common on the Internet.

On the other hand, requirements for receiving high quality stream that consumes a large amount of bandwidth and competes against other data flows are not always fulfilled to everyone. Real-time multimedia stream in the Internet is usually transmitted over RTP [1] carried on top of IP and UDP, and RTCP [1] sent with RTP supports "rate control" as one of the fundamental functions to adjust the transmission rate based on the capacity of the available bandwidth or to adapt to network congestion. To make things work properly, precise knowledge of the available bandwidth between a sender and a receiver is necessary, but it is difficult to expect appropriate data transmission rate as the network condition for each user is very changeable in the heterogeneous environment.

To provide stable streaming quality, in a different technological approach, a sender adds redundant data to its stream and a receiver detects and corrects errors occurring during transmission without the need to ask the sender for additional data. A "Forward Error Correction (FEC)" [2], [3] is the well-known algorithm that has been notably used in various applications. Although the FEC encoding rate can be coordinated by a sender and a receiver, it is difficult to decide the most appropriate encoding rate in real time.

Accordingly, we study a mechanism that *simultaneously* uses both rate control and FEC technologies for providing smooth and high quality end-to-end streaming. The mechanism dynamically adjusts transmission rate and FEC rate by estimating available bandwidth between the sender and receiver. One of the difficulties in this mechanism is that when FEC rate is increased to recover packet loss, the total amount of transmitted data is also increased in the network path and additional packet loss may occur as a result. Since the effective rate control may work against the optimal FEC encoding rate, knowing a good balance to decide the timing and the redundancy level for increasing or decreasing the transmission rate and FEC rate is vital.

In this paper, we propose an *adaptive* rate control mechanism for smooth DV streaming. We study packet loss patterns and FEC recovery rate by simulation, and define an effective algorithm that incorporates transmission rate control with adjustment of the FEC encoding rate. The bandwidth estimation is calculated from the inputs given by the condition of consecutive packet loss and FEC recovery rate. We then show an actual implementation of adaptive DV Transmission System over IP (DVTS) [4], [5] that gives good performance for provisioning multimedia streaming with the adaptive rate control mechanism.

The remainder of this paper is organized as follows: section II examines the requirements for effective streaming. Section III describes the approach how an FEC mechanism can be adapted with a rate control mechanism, and introduces the adaptive DVTS implementation with the corresponding algorithm. Section IV evaluates the adaptive DVTS on top of our testbed network, and we conclude in section V.

II. Quality Adaptation for Streaming over IP

A. Problem Statements

Audio and video transmission encoded for real-time streaming generally tries to keep pace with the user's communication condition. For maintaining streaming quality, an application

This work was supported by National Institute of Information and Communications Technology (NICT), Japan

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.

adopts a resource reservation or congestion control mechanism to avoid reductions in streaming quality caused by packet loss, jitter, or delay. To escape from interruptions and stalling, an application compensates packet losses by various approaches whose suitability may be dependent on encoding techniques and communication environments.

Network resource management is the technology being used to keep streaming quality. One of its major protocols is "Resource Reservation Protocol (RSVP)" [6], which reserves network bandwidth between a sender and a receiver for data transmission. "Class-Based Queueing (CBQ)" [7] is a resource sharing mechanism that shares the bandwidth on a link in packet networks. Both require resource management mechanisms at the network equipment level, where requiring a gateway to accommodate an essential component is hard to assume and lacks the flexibility of communication. *It is difficult to adopt these techniques for every common streaming architecture used over the wide-spread Internet*.

A congestion control mechanism would be indispensable to avoid packet loss and make end-to-end DV streaming with higher quality workable. One of the possible approaches is "TCP-friendly rate control" [8], [9]; it behaves fairly with respect to coexistent TCP flows, in order to provide a promising mechanism for avoiding severe fluctuations in the transmission rate. While it ensures fairness with competing TCP flows, the throughput of non-TCP flows does not exceed the throughput of a conformant TCP connection under the same conditions, but *this condition is not reasonable for DV streaming that consumes bandwidth* (e.g. 30Mbps).

As an open-loop approach, Forward Error Correction (FEC) is effective especially for streaming applications because it adds redundant information to packets in order to allow a receiver to correct missing packets without retransmission requests. The redundancy level is defined as FEC encoding rate, which is decided by a Bit Error Rate (BER) at the receiving side and the previously used encoding rate. FEC rate control is used to change the redundancy of data; a higher value increases the possibility of recovering the stream but increases the amount of traffic. Thus, *simply increasing FEC rate without intellectual decision does not improve the streaming quality, because additional network congestion may be occurred*.

B. Design Aspects

Although designing the ideal transmission rate control mechanism is necessary for streaming applications, there are various difficulties to find the best transmission rate to maximize streaming quality on dynamic networks. Even though the best transmission rate is examined, smooth rate control that minimizes lost packets and jitter is always preferred with respect to abrupt quality changes in a heterogeneous communication environment. The requirement is sensitive especially for transmitting DV stream over IP, since DV stream consumes a large amount of bandwidth like 30 Mbps generally.

As part of a common streaming application, RTCP is used to estimate the available bandwidth between a sender and a receiver and notify the network condition to the sender. After the data transmission rate is negotiated whether it is increased or decreased, the application triggers rate control to adjust data transmission rate being suitable for the available network bandwidth.

[10] studied Internet packet dynamics. Under the packet loss conditions the author investigated, the bottleneck comes from the slowest forwarding element (i.e. router or link) in the end-to-end chain that comprises the network path. On the other hand, it is complex to distinguish and expect packet loss patterns only by observing the current data packet loss. According to these thoughts, although "packet loss rate" becomes an important factor for expecting network condition, getting and using other characteristics would be also encouraged to adapt usable bandwidth to streaming applications.

In [11], the authors considered the problem of distributing real-time data and measured audio packets and packet loss patterns. They analyzed that the number of consecutively lost packets is small especially when the network load is low or moderate. Based on their results, only about 10% of the total lost packets are in consecutive losses of more than three packets in the stream. Therefore, we define "consecutive packet loss" as loss of more than three packets consecutively, and assume "the number of consecutively lost packets" is a useful indicator for the condition of the network and triggering transmission rate control.

Additional analysis in [11] relates to packet loss recovery. An FEC mechanism is effective for a streaming application especially when lost packets are dispersed through the stream of packets, or when network condition is unstable or changed at frequent intervals. This analysis inspires us to monitor the "FEC recovery rate" as one network condition value, because the packet loss is recovered only when it is lower than the FEC encoding rate. For instance, if FEC does not completely recover the lost packets during streaming for a certain period, it implies that the network congestion may not be converged and more rate control would be needed.

C. Related Work

There is a proposal [12] that studied adjusting FEC with quality scaling for streaming MPEG data. It proposed an analytic model called a quality adjusted FEC (QAFEC) that captures the quality distortion of streaming MPEG data in the presence of the scaling level and frame loss. However, this proposal can only detect the best combination of FEC and scaling level under a certain capacity such as maximum transmission rate adjusted by a TCP-friendly method, and cannot be applied for a high quality DV streaming that generally requires more network bandwidth.

[13] introduced a new Rate Control Scheme (RCS) for real-time traffic in networks with high bandwidth-delay products. It could improve throughput using low priority dummy packets to probe available network resources. The authors evaluated RCS how the proposed algorithm achieves high performance in terms of throughput and fairness with TCP connections by simulation. The proposed rate control scheme would be able

to apply in a closed network, but not in the Internet, because this technique requires all the routers in the connection path support the same priority policy to treat the dummy packets. Furthermore, the experimental results derived by simulation should be evaluated its feasibility by its actual implementation on a real network.

III. Adaptive Rate Control

A. Overview

We propose an *adaptive* rate control mechanism that seamlessly cooperates with "transmission rate control" and "dynamic FEC" technologies for providing the best possible DV streaming quality. The mechanism investigates three characteristics, "packet loss rate", "the number of consecutively lost packets", and "FEC recovery rate", during streaming, to characterize the network condition between a sender and a receiver. It then decides the appropriate "flow type", which is defined as a combination of "data transmission rate" and "FEC encoding rate", and dynamically adjusts these rates based on the network condition. As the concrete implementation, we implemented "adaptive DVTS". It supports that a receiver gives the information of "packet loss rate", "the number of consecutively lost packets", and "FEC recovery rate" to a sender by extended RTCP RR, and a sender adjusts the transmission rate and the FEC encoding rate of the stream. Regarding an FEC codec, we used the Reed-Solomon FEC codec developed by L. Rizzo.

B. Simulation

In order to know how the flow type can be effectively adjusted during streaming, we analyzed the correlation between packet loss patterns and FEC recovery rate by using NS-2 [14]. The parameters and procedures used in the simulation are defined as follows;

1) A sender transmits T Mbps UDP packets, and a receiver receives the packets and measures; 1) the packet loss rate ($P_{los}\%$), 2) the number of consecutively lost packets (N), and 3) the number of non-recovered UDP packets (L) within 5 seconds. Here, a combination of packet loss rate and the number of consecutively lost packets is defined as a "packet loss pattern" (P_{los}, N).
2) In the presence of competitive traffic in the simulation network, the receiver measures the different packet loss pattern and the number of non-recovered UDP packets within 5 seconds. This measurement is repeated for each 5 seconds by changing competitive traffic patterns.
3) In the same network condition in which L was given, the sender transmits T Mbps UDP packets with $F_{enc}\%$ FEC encoding rate packets, and the receiver measures the number of non-recovered UDP packets (L') within 5 seconds. This measurement is also repeated.
4) "FEC recovery rate" ($F_{rec}\%$) defined with the following form is calculated for each flow type (T, F_{enc})

$$F_{rec} = \begin{cases} \frac{L-L'}{L} \cdot 100, & (L > L', L \neq 0) \\ 0, & (L \leq L', L \neq 0) \end{cases} \quad (1)$$

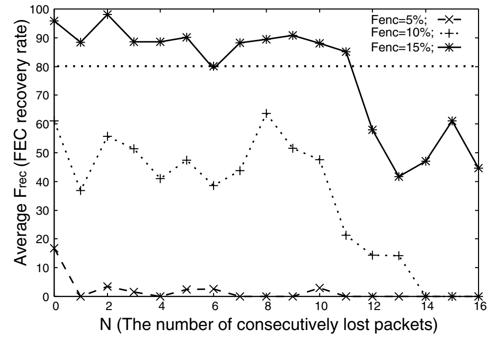

Fig. 1. The number of consecutively lost packet and FEC recovery rate.

TABLE I
Hardware in our experiment

	sender and receiver	*dummynet*
CPU	Intel Pentium M 1GHz	Intel Xeon 3.60GHz
Memory	512 MB	3 GB
OS	Linux Kernel 2.6.17	FreeBSD 5.4 Release
NIC	RealTech 100Base-TX	Intel 1000Base-T

As one of the experimental results (where $T = 30$), the correlation between the packet loss pattern (where $P_{los} = 1$ to 3%) and the average F_{rec} with different F_{enc} is shown in Fig. 1. The result indicates that the packet loss pattern is a good metric to change the flow type. When we assume that F_{rec} should be always higher than 80% for the best possible streaming, a sender should add 15% F_{enc} with the same transmission rate when N is below 11 in P_{los} =[1, 3]. On the other hand, when N is more than 11, the FEC effectiveness deteriorates severely, and hence the sender should either add more than 15% F_{enc} with the same transmission rate or reduce the data transmission rate. If the UDP packets with higher FEC encoding rate than 15% provides 80% or higher F_{rec}, the sender adds more FEC redundancy (e.g. 20%). If the UDP packets does not provide 80% or higher F_{rec}, the sender reduces the transmission rate to fulfill the 80% or higher F_{rec} quality.

C. Experimental Analysis

In this section, we create an algorithm that provides reference of the flow types and patterns of changing the transmission state by using DVTS implementation [15], and propose the *adaptive* DVTS that supports the defined algorithm. Table I shows the hardware spec and OS of a DVTS sender, a DVTS receiver, and a *dummynet* [16] network emulator. The sender and the receiver directly connect to the *dummynet* that proportionally narrows the available bandwidth from 50 Mbps to 25 Mbps within 125 seconds for each test.

1) Effectiveness of frequent transmission rate changes: Although adjusting the transmission rate in streaming is necessary for quality adaptation, according to the property of DV transmission, frequently changing the data transmission rate by discarding frame data is not highly effective or rather decreases DV streaming quality [17]. In our experiment, we also observed that the DV streaming quality was not effectively improved by changing the transmission rate frequently. Therefore, we decided that the adaptive DVTS does not change the transmission rate frequently or susceptibly, but simply changes the rate from 100% to 50% or vice versa.

2) Correlation between FEC encoding rate and non-recovery rate in full rate DV stream: In the next experiment, a DV sender sent full rate DV stream (30 Mbps) to a receiver and statically added FEC redundancy in the streaming data. The FEC encoding rate was changed from 0% to 10%, 20% and 30% step-by-step. Fig.2 shows the measurement results. In the regular DV transmission (i.e. with no FEC redundancy), the packet loss was observed from 50 seconds (at 40 Mbps bandwidth). In the DV transmission with 10% FEC redundancy, the packets were lost in 65 sec. (37 Mbps b/w). With 20% FEC redundancy, non-recovered data was observed in 75 sec. (35 Mbps b/w). With 30% FEC redundancy, non-recovered data was observed in 85 sec. (33 Mbps b/w). In 90 sec. from the transmission (32 Mbps b/w), the non-recovery rate was turned over by the transmission with higher FEC redundancy, because the total amount of traffic was gradually increased and hence additional packet loss was triggered. In addition, the non-recovered data with more than 40% FEC was observed in nearly the same timing in 30% FEC redundancy. Therefore, we decided that maximum FEC rate in the full rate should be 30%.

3) Correlation between FEC encoding rate and non-recovery rate in half rate DV stream: Next, the sender changed the DV transmission rate to half (15 Mbps) and the FEC encoding rate from 50% up to 100%. Fig.3 shows the results that when the sender sent half rate DV stream with no FEC redundancy, the packet loss was observed in nearly the same timing in the full rate transmission (i.e. in 50 seconds)[1]. However, in the half rate transmission with 50% FEC redundancy, non-recovery data was appeared in 88 sec. (32 Mbps b/w), and with 60% FEC redundancy, non-recovery data was observed in 95 sec. (31 Mbps b/w). With 100% FEC redundancy, DV packets would be able to be recovered if the available bandwidth is more than 26 Mbps. According to these results, the transmission rate should be reduced from full to half rate with more than 50% FEC rate to give higher effort when network condition is not good to keep the full rate transmission. Moreover, because adding more than 110% FEC redundancy was not highly effective than 100% FEC redundancy, we decided that the maximum FEC encoding rate in the half rate could be 100%. Note that if non-recovery rate is not converged even with 100% FEC redundancy with half rate transmission, DVTS continuously keeps the same condition.

D. Adaptive DVTS Implementation

In this section, we define an algorithm for the adaptive DVTS that dynamically adjusts transmission rate and FEC encoding rate in light of our NS-2 simulation and the experimental analysis aforementioned.

[1]This happened due to the current Linux driver that bursty transmits DV frames in a particular timing.

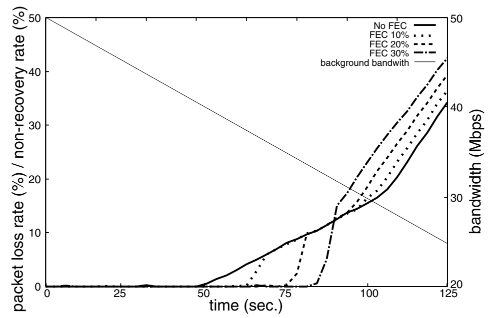

Fig. 2. Packet loss and non-recovery rate in full rate DVTS.

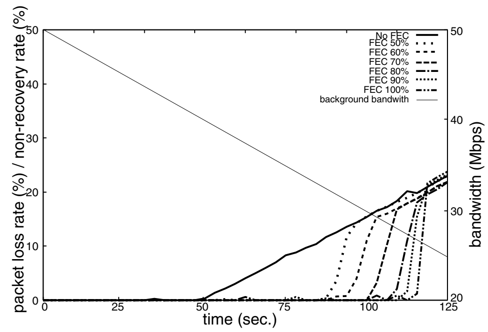

Fig. 3. Packet loss and non-recovery rate in half rate DVTS.

Table II and III provide the information how the flow type is changed during DV streaming. The algorithm checks the network state once every150 frame data (i.e. 5 seconds). For instance, on the condition with the full rate streaming with no FEC, if 1% packet loss is observed by the receiver and the number of consecutive loss is 5, then the application adds 20% FEC redundancy due to the network state that is defined “F5” in Table II. And if the network state in the next 150 frame transmission is not changed (i.e. F5 with FEC 20% in the table), then the FEC rate is decreased to 10%. This is because the sender recognizes available bandwidth has not decreased drastically and the streaming quality can be maintained even using lower FEC. If the network state in the next 150 frame transmission is changed to “F7”, the FEC rate is increased to 30%. If the network state is changed to “F15”, the sender recognizes network congestion and changes the transmission rate to half and the FEC rate to 50%. The sender then uses the algorithm defined in Table III for the next transmission. Our proposed mechanism feasibly increases the transmission rate when an application recognizes that the network becomes stable or the available bandwidth is increased (i.e. “H1” on the condition with 100% FEC). The mechanism changes the transmission rate back to the full rate without the extreme quality degradation, because FEC seamlessly recovers the data even if packet loss is again happened during expecting the available bandwidth. This mechanism has been the operational aspiration in the traditional rate control algorithm or quality adaptation mechanism.

The algorithm shown in Table II and III is also adapted the condition in which FEC does not completely recover packet losses. When the FEC recovery rate is continuously lower

TABLE II

NETWORK STATES AND FLOW TYPES IN FULL RATE TRANSMISSION (30 MBPS).

state	P_{los}	N	0%	10%	20%	30%
F1	0	0	0%	0%	0%	0%
F2	$0 < P_{los} < 1$	$0 \leq N < 3$	10%	10%	10%	0%
F3		$3 \leq N < 10$	20%	10%	10%	10%
F4		$10 \leq N$	30%	20%	20%	10%
F5	$1 \leq P_{los} < 3$	$0 \leq N < 10$	20%	20%	10%	10%
F6		$10 \leq N < 25$	20%	20%	20%	20%
F7		$25 \leq N$	30%	30%	30%	30%
F8	$3 \leq P_{los} < 8$	$0 \leq N < 30$	20%	10%	10%	10%
F9		$30 \leq N < 70$	30%	20%	20%	20%
F10		$70 \leq N$	Half + 50%	30%	30%	30%
F11	$8 \leq P_{los} < 13$	$0 \leq N < 100$	30%	20%	20%	20%
F12		$100 \leq N < 170$	Half + 50%	30%	20%	20%
F13		$170 \leq N$	Half + 60%	Half + 50%	30%	30%
F14	$13 \leq P_{los} < 18$	$0 \leq N < 180$	Half + 60%	Half + 50%	30%	30%
F15		$180 \leq N < 250$	Half + 70%	Half + 60%	Half + 50%	Half + 50%
F16		$250 \leq N$	Half + 80%	Half + 70%	Half + 60%	Half + 60%
F17	$18 \leq P_{los} < 23$	$0 \leq N \leq 300$	Half + 80%	Half + 70%	Half + 60%	Half + 60%
F18		$300 \leq N < 380$	Half + 90%	Half + 80%	Half + 80%	Half + 70%
F19		$380 \leq N$	Half + 90%	Half + 90%	Half + 90%	Half + 80%
F20	$23 \leq P_{los}$	$N =$ any	Half + 90%	Half + 90%	Half + 90%	Half + 90%

TABLE III

NETWORK STATES AND FLOW TYPES IN HALF RATE TRANSMISSION (15 MBPS).

state	P_{los}	N	50%	60%	70%	80%	90%	100%
H1	0	0	100%	100%	100%	100%	100%	Full + 0%
H2	$0 < P_{los} < 1$	$0 \leq N < 3$	100%	100%	100%	100%	100%	Full + 10%
H3		$3 \leq N < 10$	100%	100%	100%	100%	100%	Full + 20%
H4		$10 \leq N$	100%	100%	100%	100%	100%	Full + 30%
H5	$1 \leq P_{los} < 3$	$0 \leq N < 10$	100%	100%	100%	100%	100%	Full + 20%
H6		$10 \leq N < 25$	100%	100%	100%	100%	100%	Full + 20%
H7		$25 \leq N$	90%	100%	100%	100%	100%	Full + 30%
H8	$3 \leq P_{los} < 8$	$0 \leq N < 30$	90%	90%	90%	90%	100%	Full + 20%
H9		$30 \leq N < 70$	90%	90%	90%	90%	100%	Full + 30%
H10		$70 \leq N$	90%	90%	90%	80%	70%	60%
H11	$8 \leq P_{los} < 13$	$0 \leq N < 100$	90%	90%	90%	90%	100%	Full + 30%
H12		$100 \leq N < 170$	90%	90%	90%	70%	70%	60%
H13		$170 \leq N$	90%	90%	90%	60%	60%	60%
H14	$13 \leq P_{los} < 18$	$0 \leq N < 180$	60%	60%	60%	50%	50%	50%
H15		$180 \leq N < 250$	80%	80%	70%	70%	70%	70%
H16		$250 \leq N$	90%	90%	90%	80%	80%	80%
H17	$18 \leq P_{los} < 23$	$0 \leq N < 300$	90%	90%	90%	80%	80%	80%
H18		$300 \leq N < 380$	90%	90%	90%	90%	80%	80%
H19		$380 \leq N$	100%	100%	100%	90%	90%	90%
H20	$23 \leq P_{los} < 28$	$0 \leq N < 420$	100%	100%	90%	90%	90%	90%
H21		$420 \leq N < 600$	100%	100%	100%	90%	90%	90%
H22		$600 \leq N$	100%	100%	100%	100%	100%	100%
H23	$28 \leq P_{los} < 33$	$0 \leq N < 750$	100%	100%	100%	100%	90%	90%
H24		$750 \leq N$	100%	100%	100%	100%	100%	100%
H25	$33 \leq P_{los}$	$0 \leq N < 1300$	100%	100%	100%	100%	100%	90%
H26		$1300 \leq N$	100%	100%	100%	100%	100%	100%

than 80% with the same flow type within 450 frame data transmission, the algorithm increases 10% redundancy from the current FEC encoding rate, except the case that the FEC rate is already maximum in the transmission rate (i.e. 30% FEC with full rate transmission or 100% FEC with half rate transmission). If it happens on the condition with the 30% FEC with full rate transmission, the algorithm changes the flow type to 50% FEC with half rate transmission.

IV. EVALUATION

To evaluate the effectiveness of the adaptive rate control mechanism, we set up *dummynet* and Cisco Catalyst switch boxes in the testbed network (Fig.4) with various configurations.

Fig.5, 6, and 7 show the measurement results obtained in our testbed network. In each test, S_1 transmits DV stream and R_1 receives it, and both hosts work with the adaptive DVTS. To add complexity, S_2 transmits another DV stream to R_2 in the middle of the evaluation to emulate network collision or congestion over shared link.

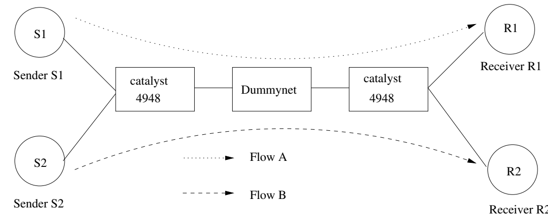

Fig. 4. Testbed network for evaluating the implementation.

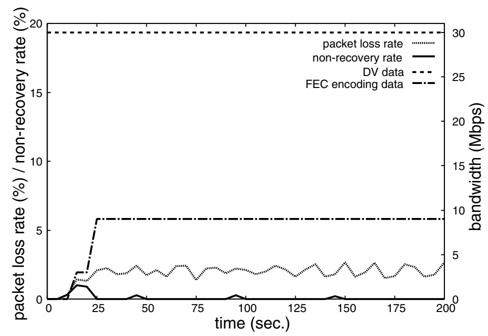

Fig. 5. Competition with normal DVTS flow over 100 Mbps link

Fig.5 shows the behavior of the adaptive DVTS when the shared link bandwidth was 100 Mbps. The DV stream consumed 30 Mbps bandwidth and the corresponding FEC redundancy used 9 Mbps (30% FEC). While the packet loss was observed in the transmission, non-recovery data was negligible (0.07%) after the flow type was converged. It took 17.4 sec to finally converge.

In Fig.6, we saw the rate control was repeated in the quality adaptation over 95 Mbps shared link, because our mechanism tries to keep high network utilization when possible. This was a worst case in the adaptive rate control mechanism, because quality degradation was observed (1% non-recovery data) in the full rate transmission.

Fig.7 shows the behavior of the adaptive DVTS over 90 Mbps shared link. The flow converged at the half rate transmission after the packet loss was observed. Then, to determine whether the transmission rate could be increased, our mechanism frequently changed the only FEC rate. In this phase, packet loss happened, but the quality degradation was avoided by FEC recovery. Although FEC caused 27.6 sec. delay of playing DV data in the receiver-side [15], the non-recovery rate was suppressed to 0.09%. It took 27.6 sec to converge at the half rate transmission.

To emulate network congestion and recovery, S_2 sent the full rate DV stream to R_2 over 90 Mbps shared link, then stopped. Fig.8 shows the behavior of our mechanism. In this measurement, when S_2's DVTS flow competed in the link (in 10 sec.), the adaptive DVTS increased the FEC rate and then decreased the transmission rate (35 sec.), because it observed consecutive packet loss. After S_2 stopped the flow, the packet loss was decreased (55 sec.). The adaptive DVTS adjusted the FEC rate to expect the available bandwidth (55 to 70

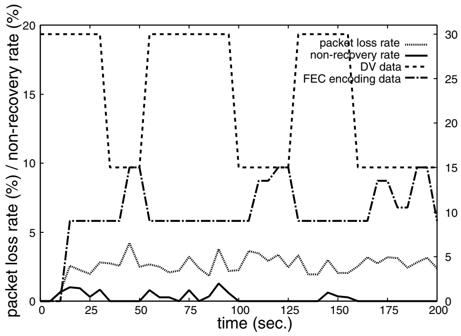

Fig. 6. Competition with normal DVTS flow over 95 Mbps link

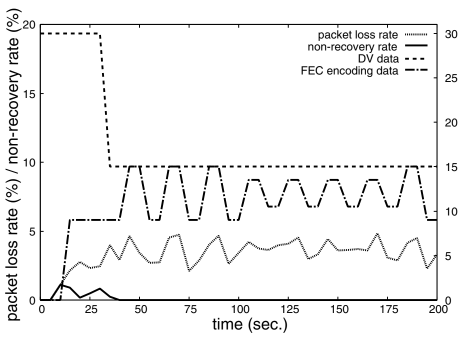

Fig. 7. Competition with normal DVTS flow over 90 Mbps link

sec.), and then reduced the FEC rate and returned the full rate transmission. It took 17.8 sec to return to the full rate transmission.

V. Conclusion and Future Work

In this paper, we propose an *adaptive* rate control mechanism in DVTS that provides the best possible quality for DV streaming. It gives a good balance to incorporate transmission rate control and dynamic FEC technologies.

The adaptive DVTS controls the transmission rate and the FEC encoding rate based on the network condition between a sender and a receiver. It investigates packet loss rate, the number of consecutively lost packets, and FEC recovery rate during streaming. After the mechanism examines the condition of the network, it decides the flow type, which is defined as the combination of the data transmission rate and FEC encoding rate, and transmits the appropriate DV stream to the receiver.

We verified efficiency of the adaptive DVTS implementation on our testbed network. According to the result, our proposed mechanism utilizes network resource effectively. It accordingly maintains DV streaming quality by dynamic FEC, even though packet loss occurs. However, we saw the transmission rate did not converge in the worst case with the quality degradation.

In our future work, we will improve our mechanism and evaluate on a heterogeneous communication environment like PlanetLab testbed [18], in order to analyze the adaptive DVTS in case of a large scale deployment over the Internet. It is also

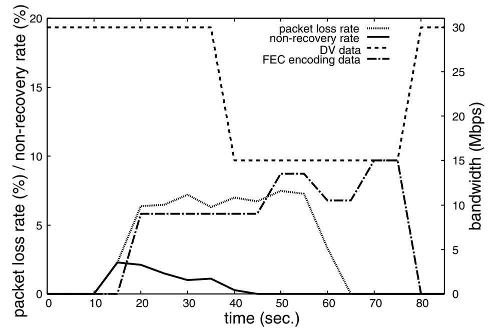

Fig. 8. Adaptive DVTS in the recovery of the network congestion

motivated to support one-to-many multicast communication in the future.

References

[1] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” RFC 3550, July 2003.

[2] J.C. Bolot and A. V. Garcia, “Control Mechanisms for Packet Audio in the Internet,” Proc. of IEEE INFOCOM, March 1996.

[3] C. Perkins and O. Hodson, “Options for Repair of Streaming Media,” RFC 2354, June 1998.

[4] A. Ogawa, K. Kobayashi, K. Sugiura, O. Nakamura, and J. Murai, “Design and Implementation of DV based video over RTP,” Proc. of Packet Video Workshop 2000, May 2000.

[5] K. Kobayashi, A. Ogawa, S. Casner, and C. Bormann. “RTP Payload Format for DV (IEC 61834) Video,” RFC 3189, January 2002.

[6] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, and S. Jamin, “Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification,” RFC 2205, September 1997.

[7] S. Floyd and V. Jacobson, “Link-sharing and Resource Management Models for Packet Networks,” IEEE/ACM Transactions on Networking, Vol.3, No.4, pp.365-386, August 1995.

[8] S. Floyd and K. Fall, “Promoting the Use of End-to-End Congestion Control in the Internet,” IEEE/ACM Transactions on Networking, Vol.7, No.4, pp.458-472, August 1999.

[9] M. Handley, S. Floyd, J. Padhye, and J. Widmer, “TCP Friendly Rate Control (TFRC): Protocol Specification,” RFC 3448, January 2003.

[10] V. Paxson, “End-to-End Internet Packet Dynamics,” Proc. of ACM SIGCOMM ’97, pp.139-152, 1997, Cannes, France.

[11] J.C. Bolot, H. Crépin, and A. Vega-Garcia, “Analysis of Audio Packet Loss over Packet-Switched Networks,” Proc. of ACM NOSSDAV ’95, April 1995, New Hampshire, USA.

[12] H. Wu, M. Claypool, and R. Kinicki, “Adjusting Forward Error Correction with Quality Scaling for Streaming MPEG,” Proc. of ACM NOSSDAV ’05, June 2005, Washington, USA.

[13] I. F. Akyildiz, Ö. B. Akan, and G. Morabito, “A rate control scheme for adaptive real-time applications in IP networks with lossy links and long round trip times,” IEEE/ACM Transactions on Networking, Vol.13, Issue 3, pp.554-567, June 2005.

[14] “The Network Simulator – ns-2,” http://www.isi.edu/nsnam/ns

[15] K. Matsuzono, H. Asaeda, K.Sugiura, O. Nakamura, and J. Murai, “Analysis of FEC function for Real-Time DV streaming,” Asian Internet Engineering Conference ’07, LNCS 4866, pp.114-122, Novemver 2007.

[16] L. Rizzo, dummynet, http://info.iet.unipi.it/~luigi/ip_dummynet/

[17] Xin WANG and Henning SCHULZRINNE, “Comparison of Adaptive Internet Multimedia Applications,” IEICE Tran. COMMUN., Vol.E82-B, NO.6, June 1999.

[18] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, “PlanetLab: An Overlay Testbed for Broad-Coverage Services,” ACM SIGCOMM Computer Communication Review, Vol.33, No.3, July 2003.