2004년도 대한전자공학회 하계종합학술대회 논문집 27권 1호

QoS Based Enhanced Collaboration System Using JMF in MDO

김종성

LG 전자 CDMA(연), 한국향개발실
●● : 02-850-3670

Jong-Sung Kim

Development Dept 3. CDMA Handsets , LG Electronic Inc
Kumchong-gu Seoul 153-023 Korea
E-mail: newpower76@lge.com

Abstract

This paper presents the design and implementation of a QoS based enhanced collaboration system in MDO. This is an efficient distributed communication tool between designers. It supports text communication, audio/video communication, file transfer and XML data sending/receiving. Specially, this system supports a dynamic QoS self-adaptation by using the improved direct adjustment algorithm (DAA+). The original direct adjustment algorithm adjusts the transmission rate according to the congestion level of the network, based on the end to end real time transport protocol (RTP), and controls the transmission rate by using the information of loss ratio in real time transport control protocol (RTCP). But the direct adjustment algorithm does not consider when the RTCP packets are lost. We suggest an improved direct adjustment algorithm to solve this problem. We apply our improved direct adjustment algorithm to our of QoS (Quality of Service) [1] based collaboration system and show the improved performance of transmission rate and loss ratio.

I. INTRODUCTION

It is important that the collaboration system guarantees any level of QoS to provide good service to a collaboration group. In this paper, we solve this problem by using the improved direct adjustment algorithm (DAA+). The improved direct adjustment algorithm (DAA+) is using the RTCP timer. RTCP timer which is proposed by us is time interval from receiving the RTCP packet to receiving the next RTCP packet. We can control the transmission rate by using RTCP timer when the RTCP packet is lost. Implementation of the QoS based collaboration system includes audio, video and file transfer, RTP monitoring, controlling transmission rate, whiteboard, XML data sending/receiving in distributed environments.

The rest of the paper is organized as follows. Section 2. describes the direct adjustment algorithm. Section 3 presents the architecture of our QoS based collaboration system and implementation. In section 4, we propose the improved direct adjustment algorithm (DAA+). Section 5 evaluates the performance. The last section concludes the paper.

II. RELATED WORKS

2.1 The Direct Adjustment Algorithm

During a collaborative session, each receiver reports in its control packets the percentage of lost data noticed since sending the last control packet. At the sender site, the RTCP[3] packets are processed and depending on the loss values reported within the RTCP packet, the sender can increase, decrease or keep its current sending rate.

III. ARCHITECTURE AND IMPLEMENTATION

The design goal of a QoS based collaboration system provides two major aspects. First, the system supports many types of activities to each collaborator: audio/video communication, text communication, files sending, white board, XML data sending/receiving, etc. Second, the system has a dynamic QoS self-adaptation ability based on the current network performance.

Figure 1 is the architecture of proposed QoS based collaboration system. CollsysServer carries out the role of Server, and Collsys carries out the role of Client.

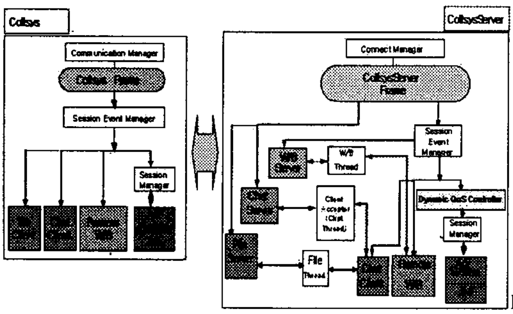

Figure 1: The architecture of proposed QoS based collaboration system

As shown in Figure 2, each session receiver issues its RTCP report to the network periodically. The RTCP report contains the receiver's id and current receiving status including the packet loss ratio. The RTP monitor collects these reports and sends it to the dynamic QoS self-adaptation controller. The dynamic self-adaptation QoS controller makes a proper QoS adaptive decision by using the improved direct adjustment algorithm and sends it to the sender if necessary. The sender dynamically adjusts its transmission rate accordingly and sends an adapted stream to the network.

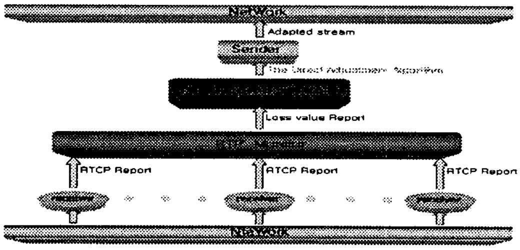

Figure 2: Dynamic QoS self-adaptation flow chart

IV. AN IMPROVED DIRECT ADJUSTMENT ALGORITHM (DAA+)

4.1 Problems of the DAA

The control of the transmission rate in DAA is fundamentally using information of the RTCP packets such as loss, jitter and delay etc. But the sender does not know the state of current network when the loss of RTCP packet happens. In this case, the sender controls sending rate by using the previous information of the RTCP packets. This situation can cause following problems:

- First, the sender does not know the state of current network because the sender does not know the RTCP packet loss. The sender controls the transmission rate by using the information of pervious RTCP packet, which can cause the congestion of current network.
- Second, the sender can't exactly know the current state of network because the sender can't know the information in lost RTCP packets such as loss and jitter. For instance, if the RTP loss ratio before losing RTCP packets is relatively larger than that after losing RTCP packets, the sender will not exactly control the transmission rate. Although congestion of the current network is going down, the sender will not know the fact. Therefore the sender can't exactly control the transmission rate.

4.2 Proposal of the DAA+

We solve these problems by proposing the direct adjustment algorithm (DAA+) which is using $RTCP_{timer}$. The proposed $RTCP_{timer}$ is time from receiving the RTCP packet to receiving the next RTCP packet. We can determine the $RTCP_{timer}$ by using Equation 1. And k is the constant value and it is dependent state of current network.

$$RTCP_{timer} = RTCP_{interval} + k \qquad (1)$$

$RTCP_{interval}$ is the RTCP transmission time. As shown Equation 2, L(t) represents the number of other users that have been heard from at time t. The state is initialized to L(0)=1 when the user joins the group. And C is the average interval between arrivals of RTCP packets (from any user) at end-system. We can determine the $RTCP_{interval}$ by using Equation 2.

$$T_d = \max(T_{\min}, CL(t)) \qquad (2)$$

$T_{\min}$ is 2.5 seconds for the initial packet from the user, and 5 seconds for all other packets. To avoid synchronization, the actual interval is then computed as a random number uniformly distribution between 0.5 and 1.5 times T_d.

If the sender does not receive the RTCP packet within $RTCP_{timer}$, the sender will decrease the transmission rate by using Equation 3. Figure 3 shows the state diagram of the improved direct adjustment algorithm (DAA+).

$$rate = rate \times (1 - \delta) \qquad (3)$$

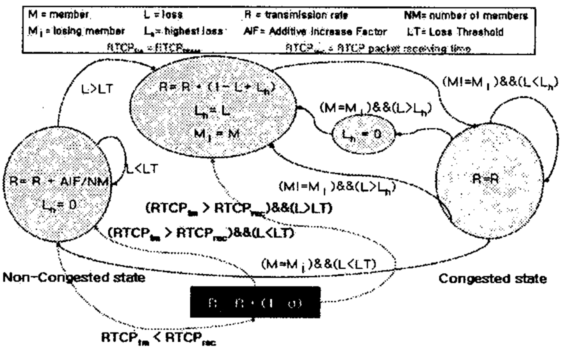

Figure 3: State diagram of the improved direct adjustment algorithm (DAA+)

V. SIMULATION

5.1 Modeling for simulation

We design the value of the link capacity is 1Mbps and initial RTP transmission rate is 5Mbps and $RTCP_{interval}$ is 5 seconds. Table 1 shows parameters of the simulation. We set the value of δ is a half of Loss Threshold (LT). In this simulation, to evaluate the transmission rate and RTP packet loss value of the DAA+, the RTP sender 1 sends RTP packets to RTP receiver. After 300 seconds, the RTP sender 2 sends RTP packets to RTP receiver.

Table 1: Parameters of the simulation

AIF/NM	25kbps	$RTCP_{interval}$	5 seconds
RTCP packet size	200bytes	k	1 second
RTCP loss ratio	20%	δ	6.5%
Loss Threshold (LT)	13%	Simulation time	20 minutes

5.2 Performance analysis

To verify the performance of DAA+, we show the transmission rate and RTP packet loss ratio. Figure 4, 6 show the transmission rate and RTP packet loss ratio by using the original direct adjustment algorithm (DAA) and Figure 6, 8 show the transmission rate and RTP packet loss ratio by using the DAA+. Figure 4, 6 show the RTP sender 1 increases the transmission rate up to link capacity (1Mbps) by using the direct adjustment algorithm ($L < LT$). After 300 seconds, the RTP sender 1 decreases the transmission rate by using the direct adjustment algorithm ($L > LT$). And the transmission rate fluctuates because the loss ratio fluctuates. The RTP sender 2 controls the transmission rate like the RTP sender 1. After 443 seconds, both RTP sender 1 and RTP sender 2 have a same transmission rate, that is to say, RTP sender 1 and RTP sender 2 have a same bandwidth (each 0.5Mbps). As shown Figure 5 and 7, in case of initial transmission times, the transmission rate of the RTP sender 1 and 2 dramatically decrease in short time. This consequence is that the RTP sender 1 and 2 decrease the transmission rate by using the Equation 3 (when the RTCP packet loss happens). In comparison Figure 6 with Figure 7, we can know that the loss ratio of using the DAA+ is smaller than that of using the DAA. Table 2 shows the performance of loss ratio. From the Table 2, we see that using the DAA+ is the more efficient than using the DAA.

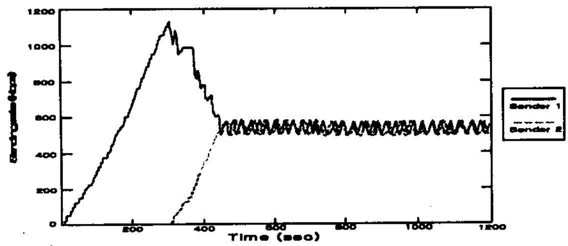

Figure 4: Transmission rate of using the DAA

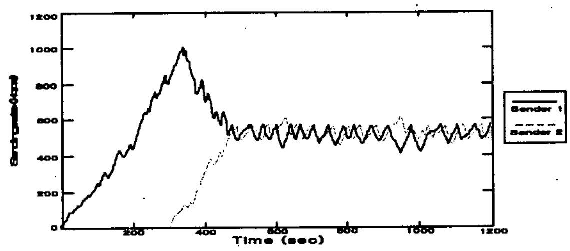

Figure 5: Transmission rate of using the DAA+

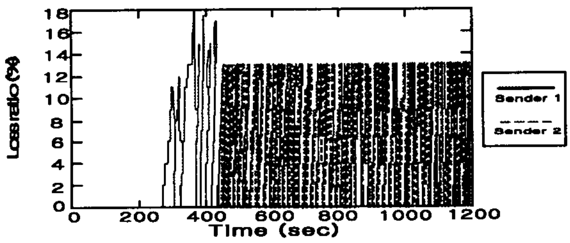

Figure 6: Loss ratio in DAA

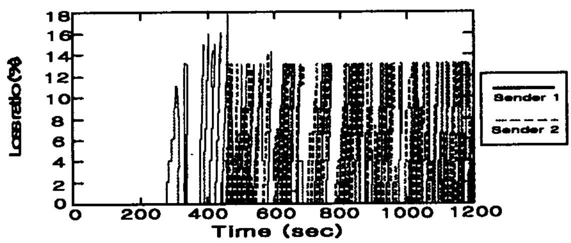

Figure 7: Loss ratio in DAA+

Table 2: Performance of loss ratio

	The sender 1	The sender 2
Loss capacity in DAA	25,553bytes	17,576 bytes
Loss capacity in DAA+	8,994bytes	5,273 bytes
Loss ratio	35%	30%

•. CONCLUSION

In this paper, we designed and implemented the QoS based enhanced collaboration system as one of modules in distributed environments. Specially, the proposed QoS based collaboration system supports the dynamic self-adaptation QoS control by using the proposed direct adjustment algorithm (DAA+) and XML data sending/receiving. DAA+ can be a better solution to solve an important problem about the RTCP packet loss. We show the good performance through simulations. Also this system is not only platform independence but also convenient for user and management. The QoS self-adaptation controller is not fully implemented. Future work is to implement perfectly the QoS self-adaptation controller and to measure its performance in an extended experimental circumstance.

References

[1] Nichoals John Yeadon, "Quality of Service Filtering for Multimedia Communications", Ph.D thesis, Lancaster University, UK, April 1996.

[2] Schulzrinne, H., Casner, S., Frederick, R. & Jacobson, 1996. RTP: a transport protocol for real-time applications, RFC 1889, Internet Engineering Task Force.

[3] Francisco Afonso, "Virtual Reality Transfer Protocol (VRTP): Implementing a application for the Real-time Transport Protocol (RTP) using the Java Media Framework (JMF)," Master's thesis, Naval Postgraduate School, USA, March 1999.