局所的適応制御型
リアルタイム・マルチキャストシステム

伊藤 嘉浩[†]　酒澤 茂之[†]　石倉 雅巳[†]　浅見 徹[†]

TCP/IP ネットワークの普及に伴い，様々なアプリケーションやサービスや展開されてきている．その中でも TV 会議システムのようなリアルタイム・マルチキャスト通信への要望が大きいが，現在の TCP/IP ネットワークでは，RSVP などによる QoS 保証機構が十分に整備されていないため，best-effort 型のサービスしか提供されない．従って，ネットワークの状態に合わせて送信者がデータの送信速度を負帰還制御するようなアプリケーションが必要であるが，マルチキャスト環境においては，受信者の環境が必ずしも均一ではないため，このような輻輳制御を行うことは困難であった．この解として，筆者らは，拠点毎に局所的な輻輳制御を行うことで，マルチキャストセッション内で適応制御を行う局所的制御型 RTP Mixer を提案した． 本報告では， 局所的適応制御型 Mixer を含む適応制御型リアルタイム・マルチキャストシステムについて述べる．

Rate-adaptive Real-time Multicast TV conference system with Locally Adaptive Packet Flow Control

Yoshihiro ITO[+]，Shigeyuki SAKAZAWA[+]，Masami ISHIKURA[+]，Tohru ASAMI[+]

As TCP/IP networks develop, various type of applications or services are appearing. Especially, many people want to use real time multicast applications over TCP/IP networks like a TV conference system. Most of the current TCP/IP networks, however, still do not support QoS guarantees using RSVP, so that they provide only a best-effort service. Therefore, such real time applications must control data transmitting rate by the network or receiver's condition. However, it is difficult to control data rate over a multicast session, since every receiver on a multicast network does not necessarily have the same environment. To solve this problem, the authors proposed a locally adaptive control intermediate system. This paper describes a rate-adaptive real-time multicast system with locally adaptive packet flow control.

1. Abstract

Currently, as the Internet develops, various of application services are appearing. Especially, a WWW service provides not only text but also multimedia information, so it is natural for users to want to transmit and receive multimedia data, voice or movies, in real-time.

Unlike applications such as ftp, telnet or WWW using usual file transfer services which demand only one "Quality of Service (QoS)" reliable data transfer service, applications that transmit a large amount of real-time data (voice or movie) need other types of QoS. In some case it will be guaranteed bandwidth, in another case it will be maximum delay time. There are two

[†]（株）ＫＤＤ研究所

[+] KDD R&D Laboratories Inc.

methods to provide QoS for application demands, i. e. first, the network provides enough QoS on demand by an application, and second, an application adaptively controls data transmitting rate like flow control to obtain the minimum QoS or not affect other traffic. The former is called an "application-initiated system" and the latter is called a "network-initiated system"[1].

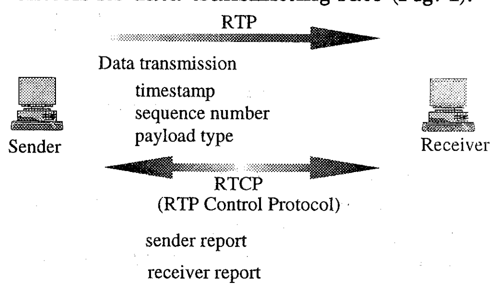
In general, the current Internet can provide only one QoS, best-effort data transfer service, but recently, IETF is standardizing RSVP (Resource reSerVation setup Protocol) [2] as a new protocol to guarantee QoS. RSVP is a multicast-oriented protocol, so it is suitable for a real-time multicast application over TCP/IP networks such as a TV conference system. However, RSVP is no more than a signaling protocol to carry resource request messages, so every node over the path between end nodes must have some mechanism to actually reserve resources. For example, each node needs traffic control modules to reserve bandwidth, but now, because of the difficulties concerned with policy control and accounting, RSVP is expected to function as a protocol to support next generation services and to be used within only limited networks (for example, private networks).

Therefore, over the current TCP/IP networks like the Internet which provide only best effort service, an application should be a network-initiated system. But under multicast networks, it is very difficult for a source to control rate-adaptively by feedback from receivers, since receivers are not always in the same network environment.

In this paper, we report a rate-adaptive real-time multicast system with a locally adaptive control RTP[3] mixer/translator. Section 2 and Section 3 describe real-time data transmission over TCP/IP, and the problem of real-time multicast transmission over TCP/IP and related matters. In Section 4, we propose a new architecture to control rate-adaptively in a real-time multicast environment. Section 5 and Section 6 give a simulation-based evaluation of this architecture and conclusions.

2. Real-time data transmission over TCP/IP networks

IETF (Internet Engineering Task Force) is standardizing RTP (Realtime Transport Protocol) for real-time data communication. Unlike RSVP or other reservation protocols, RTP does not guarantee any QoS. RTP carries data that has real-time properties, and a timestamp, sequence number, etc. Also, RTP control protocol (RTCP) monitors the QoS and conveys information about the participants in a session between senders and receivers. With RTP/RTCP, therefore, applications can perform a network-initiated QoS control. For example, an application can estimate its network status from receivers' reports and when it detects congestion, it controls its data transmitting rate (Fig. 1).



**RTP itself cannot guarantee any QoS.*

Fig. 1 Control with RTP/RTCP

The authors developed a dynamically rate-adaptive TV conference system with RTP/RTCP and showed that feedback rate control with RTP/RTCP like this can work well in point-to-point communications[4].

3. The problem of rate-adaptive real-time multicast communication

It is difficult to extend the feedback control method with RTP/RTCP proposed in 4) to real-time multicast communication. The reason is that a multicast environment is generally heterogeneous, that is, the network environment of each receiver is not always the same. Some users may access the network via a ten kbps telephone line with modems, and others may use hundreds of kbps bandwidth via fast LANs. On the other hand, in a network which has many users, users may hardly use any bandwidth because of the network congestion.

Considering network-initiated control with RTP/RTCP in a multicast environment, if a sender receives a RTCP receiver report from one receiver and estimates the network congestion of that receiver, the sender, however, may estimate no congestion from receiver reports from other receivers. In this case, the sender will be unable to decide whether he should decrease the data transmitting rate. If the sender decides to decrease the rate on the demand of the receiver who lives in a congested network, other receivers who live in a "silent" network have to receive low-quality data although they have enough resources to receive higher quality data. Conversely, if he decides to continue to send data at same rate at the sacrifice of one receiver, the congestion in his network becomes worse.

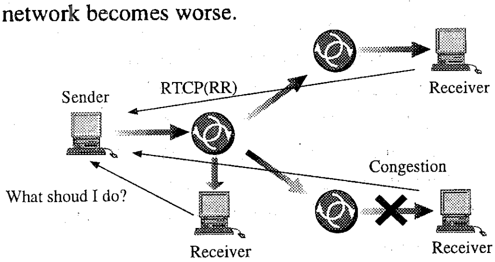


Fig. 2 The problem under a simple rate-adaptive multicast

3.1 HMC

To transmit multicast data efficiently in a heterogeneous network, which has many users in various type of environments, Heterogeneous MultiCast (HMC) was proposed[5]. The HMC uses layered encoding and transmits each layered data with an individual multicast session. A receiver selects necessary sessions according to its environment so that every receiver can acquire data appropriately. The HMC adapts to a heterogeneous receivers environment statically, so the HMC works very well under networks that guarantee QoS like a bandwidth guarantee for every receiver with ATM or RSVP, that is, networks where applications can perform application-initiate control. Therefore, the HMC cannot adapt to networks which provide only best-effort service like the Internet, that is , applications must perform network-initiated control.

3.2 RLM

To control rate-adaptively under a real-time multicast environment, Receiver-driven Layered Multicast (RLM) was proposed[6]. Like the HMC, the RLM uses layered encoding and transmits each layered data with an individual multicast session. In the RLM, receivers join and leave the multicast sessions according to their network environment or status, so that the amount of traffic can be controlled. When no receivers who join a session which sends a layer exist further than some router, the path beyond that router will be deleted by a multicast routing protocol, disused traffic will not flow and congestion will be avoided. In the RLM, a sender or an intermediate system does not detect and control congestion but receivers do. RLM proposes shared learning when the number of receivers increases, however, as the number of receiver increases, the behavior is thought to be unstable. Also, as a common point with HMC, both senders and receivers must manage a large number of (RTP/RTCP) sessions in proportion to the number of layers. And more, it is difficult for many multicast sessions to be synchronized.

4 Rate-adaptive real-time multicast system with Locally Adaptive RTP Mixer/Translator

4.1 Locally adaptive control RTP mixer/translator

RTP defines a mixer and a translator as intermediate nodes. Some kinds of mixer or translator, for example, can decode received data and re-encode changing the coding rate, so that they can send data at an appropriate transmitting rate per each receiver. By this definition, however, an intermediate node itself does not change the transmitting rate depending on its network status, so it must have knowledge about the environment of all receivers. Further, TCP/IP networks cannot guarantee constant network status, so a mixer/translator can not control local fluctuations of TCP/IP network traffic.

When an application is transmitting real-time data with UDP which has no flow control mechanism, it has to control congestion in order to prevent a network from becoming worse or increasing packet loss. In the case of using an ordinary mixer/translator, only the data source dynamically controls congestion, so the data transmitting rate to all multicast receivers decreases because of congestion in a partial network. Here, by extension of this ordinary mixer/translator, we propose a Local Adaptive Control RTP Mixer/translator (LACRM) which controls data transmitting rate according to the local congestion of networks when it relays data from a data source. Using LACRM, an application can transmit data at an appropriate transmitting rate in response to each receiver's status. A LACRM is placed in each local network and acquires RTCP receiver report (RR) packets from receivers. A LACRM sends its own receiver report to a sender or a parent LACRM (Fig. 3).

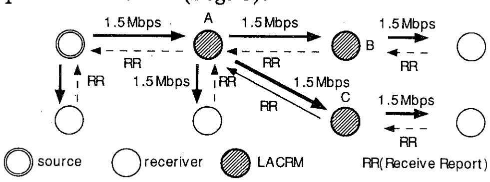


Fig. 3 Abstract of LACRM (before congestion)

When a LACRM detects congestion in a local network based on the packet loss rate in RTCP RR messages, it decreases the data transmitting rate to the congested network. In Fig. 3, for example, congestion occurs in the network between LACRM A and C, so A decreases the data transmitting rate to C (Fig. 4). However, LACRM A does not report the local congestion information as a receiver report, so networks beyond B can communicate at the same rate as before.

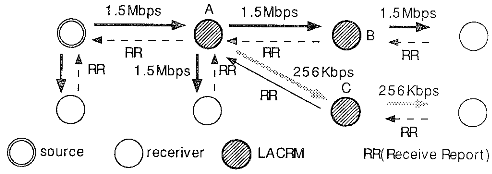


Fig. 4 Abstract of LACRM (after congestion)

4.2 Multicast routing

We can select two types of behavior of LACRM when a LACRM relays data as follows. 1) router mode : A LACRM sends data to the session with the same multicast IP address and port number as received data. 2) gateway mode : A LACRM sends data to the session with a different multicast IP address or port number from the received data. In router mode, an application uses only two multicast sessions as RTP and RTCP for one data, but needs to interact with its routing process in order to send data only to its subordinate networks within the same multicast network selectively. On the other hand, in gateway mode, the number of sessions that an application needs to communicate with RTP/RTCP is (the number of its subordinate networks) times 2. This means that the processing load at a LACRM increases depending on the number of sessions but a LACRM can work independently of the multicast routing by using unique multicast sessions for each subordinates network. Therefore, a LACRM can coexist with current routers so that it is easy to shift.

4.3 Packet flow control

In order to control the data transmitting rate at intermediate nodes, it is necessary to decode once and re-encode at an appropriate rate for each destination network. However in general, this process causes an intermediate node to take a very heavy load, so that the delay time between a sender and a receiver increases and this delay time causes a problem in real-time systems. One solution to this problem is as follows. First, the information to transmit is classified according to its priority with layered coding. Next, this classified information is encapsulated into packets and the priority is included within the packet header. Finally, when the network is congested, the priority within a header is checked and the packets whose priority are low are discarded. Hence, only the processing header can control the data transmitting rate efficiently.

In our implementation as one example, we used MPEG-1 based layered encoding[7]. Each type of frame of MPEG-1 is divided into two subtypes, low resolution information and enhancement information, so there exists 6 types of frames I_L, I_H, P_L, P_H, B_L and B_H. The subscript L means low resolution information and H means enhancement information. Table 1 shows the relationship of each frame type.

	Previous	Next
I_L	N/A	N/A
I_H	I_L	N/A
P_L	P_L or I_L	N/A
P_H	P_L	N/A
B_L	P_L or I_L	P_L or I_L
B_H	B_L and P_H or I_H	P_H or I_H

Table 1 Relationship between frames

I_L does not need information from other frames, but to decode the B_H frame needs the previous B_L frame, previous I_H or P_H frame, next B_L frame, and next I_H or P_H frame. Therefore, we can control data transmitting rate at 6-levels[i]. If we consider that spatial information is more important than temporal information, the priority becomes $I_L > I_H > P_L > P_H > B_L > B_H$, otherwise, $I_L > P_L > B_L > I_H > P_H > B_H$, and if the identifier of these frame types is included within a packet header, an application and a LACRM can control the rate only with packet header processing.

4.4 Congestion detection and recovery

In this system, a data source detects network congestion by monitoring packet losses at each receiver as with TCP. However, a data source cannot distinguish losses caused due to congestion from ones by temporal lowering of line quality. On the other hand, as a method to recover the data transmitting rate, we can envisage a slow-start rate recovery, but these are not always appropriate solutions. As a future topic, we will consider not only packet losses but also fluctuation of jitter as an indication of network congestion, since accumulating packets in queues of intermediate systems during congestion increases delay time.

4.5 Real-time socket interface

The authors considered RTP as a transport protocol for real-time communication and proposed an extended socket interface with SOCK-REALTIME[8]. This interface lowers the cost of developing a real-time application. The LACRM needs to be implemented for each application, and we developed our system with this interface for wide use. This section simply describes this real-time socket interface.

In general, RTP uses a service of an under layer protocol (UDP, etc.) to identify RTP/RTCP or to identify multi-sessions. RTP is implemented on UDP in this interface. Figure 5 shows the protocol stack of this interface.

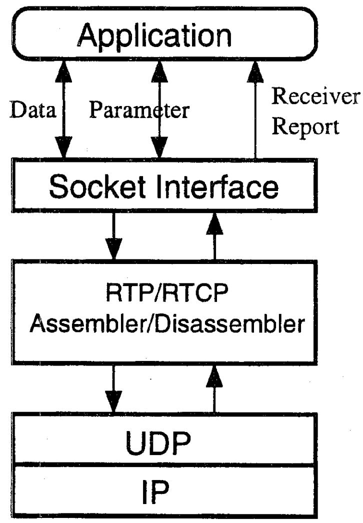


Fig. 5 Protocol stack of real-time socket interface.

An application transmits data, parameters and receiver reports etc. via this real-time socket interface. Adding to TCP/UDP parameters, an application must set RTP parameters, for example CNAME, when using this real-time socket interface. For example, with Windows Sockets 2 API[9]

---

i Adding voice data to this encoding, we can control at 7-levels, and an application can select the voice data as most important or least important.

or RAPI[2], an application can set QoS parameters based on RFC1363 but cannot set RTP specific parameters like CNAME or RTCP interval. In this interface, to provide a uniform interface, we extended SETSOCKOPT call, which is the standard function of the UNIX/socket interface. With this new SETSOCKOPT call, an application can set RTP parameters. The real-time socket interface, then, includes this information in RTP/RTCP packets, encapsulates UDP packets, and sends them to a network.

5. Evaluation of LACRM using simulation models

In this section, we evaluate LACRM using simulation models. We used Alta Group BONeS DESIGNER[10] as a simulation platform. Fig. 6 shows a real-time multicast environment without LACRM and Fig. 7 shows one with LACRM.

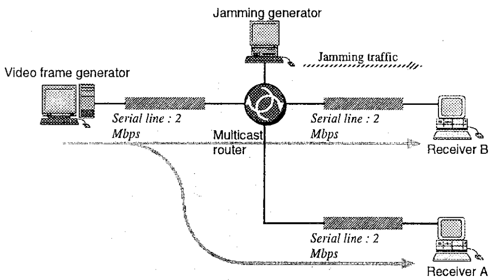


Fig. 6 Experimental environment without a LACRM.

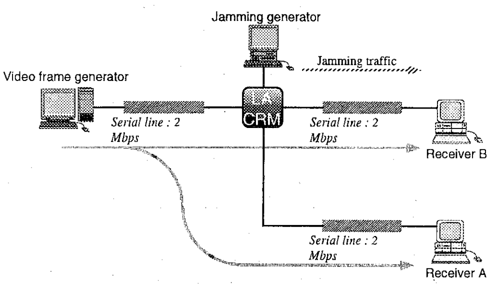


Fig. 7 Experimental environment with a LACRM.

In Fig. 6 and 7, the video frame generator generates layer encoded pseudo-frames, whose type is I_L, I_H, P_L, P_H, B_L or B_H. The video frames are generated periodically (30 frames per second) in a fixed order and the length of each type of frame has a normal distribution with each mean and variance based on real movie data[7]. Table 2 shows means and variances of these types of frame length[7].

	Mean [bits]	Variance
I_L	56095	34683223
I_H	76247	254603849
P_L	28807	28573353
P_H	41618	111231894
B_L	20728	56043053
B_H	12109	23945851

Table 2 Mean and variance of each type of frame

The video frame receiver receives and assembles these pseudo-frames. Each receiver monitors the sequence number of packets, calculates packet losses, and reports packet losses to a sender with RTCP RR packets. A receiver sends RTCP RR packets when it receives RTP data packets. Except for I_L frames, if a frame that is used for other participating frames is lost, the frames generated using that frame are discarded. In each environment, there are two video frame receivers via a 2Mbps serial line. To make the analysis more simple, in this model we considered that the transmission delay times over every serial line are all zero. The network which has receiver B has a Jamming generator which generates other traffic. As the general traffic, the jamming traffic consists of packets based on a normal distribution packet length and Poisson distribution packet interval. Considering the default MTU length over unknown networks, the mean length is 500 [bytes], and the variance of length is 100000. To make the mean bandwidth used by the jamming traffic 2 [Mbps], the mean packet interval is 0.002 [sec]. In this environment, we measured used bandwidth for each receiver and the number of frames that each receiver could assemble. Here, we considered three cases. First, a video frame generator sends a multicast packet to receiver A and B, and does not control data transmitting rate. Second, a video frame generator controls data transmitting rate depending on the status of either receiver without LACRM. Third, a video frame generator controls data transmitting rate depending on the status of a LACRM, and a LACRM controls rates according to each receiver. The first case uses a Fig. 6 configuration, and the second case also uses a Fig. 6 configuration. The third case uses a Fig. 7 configuration. In each case, a video frame generator sends data for a hundred seconds. A jamming generator generates jamming traffic from 30 [sec] to 50 [sec]. These results are shown in Fig. 8, 9 and 10.

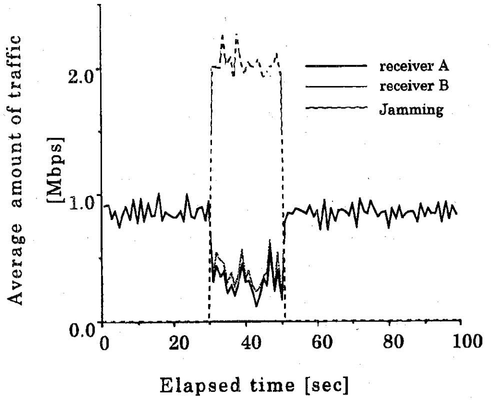


Fig. 8 Average amount of traffic (usual multicast)

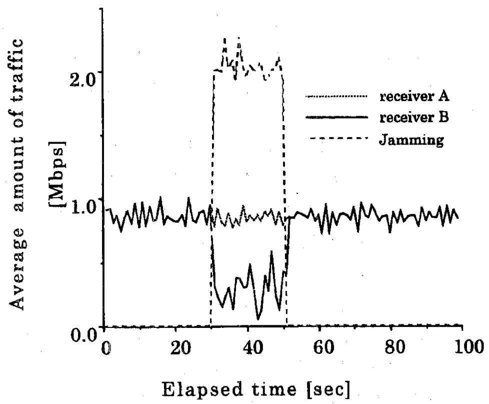


Fig. 9 Average amount of traffic (with LACRM)

Fig. 8 shows the results using the second environment, and Fig. 9 shows the results using the third environment. In each figure, the horizontal axis means the elapsed time [sec]. In Fig. 8 and 9, the vertical axis means average amount of traffic per second [Mbps]. From these results it is seen that, with a LACRM, each receiver could acquire data at appropriate rates.

Fig. 10 shows the average number of frames that were received by a receiver and also able to be decoded in the receiver per second. Fig. 11 shows the number of total unconstructable frames that were discarded in the receiver during congested periods. Before and after congested periods, no frames were discarded. This figure includes results of the first, second and third cases.

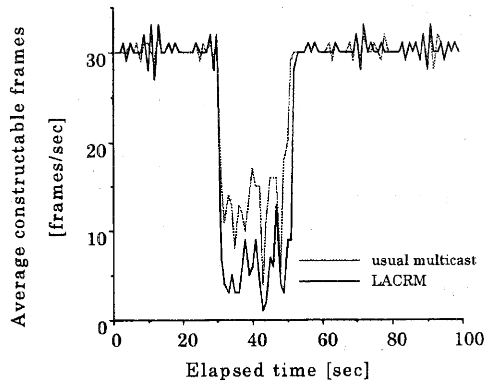


Fig. 10 Average constructable frames at receiver B

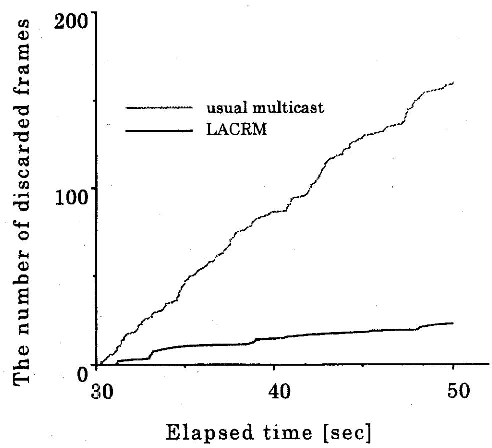


Fig. 11 Number of discarded frames at receiver B

From Fig. 10, the average frame rate under usual multicast is larger than the one with a LACRM during congestion. At a glance, a LACRM seems to be unnecessary. This was caused by inappropriate congestion control. But regarding the number of frames discarded, the number for a usual multicast exceeds extremely the one with a LACRM. This means the following. Without a LACRM, a large amount of data were transmitted even when the network was congested. This makes the network worse and may disrupt the network.

In this model, we used RTP data packets receives as RTCP RR transmit triggers. In general, RTCP RR packets are transmitted periodically and the interval is defined by the available bandwidth and the number of receivers. Therefore as the number of receivers increases, the interval increases. This large interval dulls the control, however the system using LACRMs is controlled locally, so the number of receivers which are assigned one LACRM will not become very large.

From these results, it was found that a LACRM could control transmitting rate locally, and this makes total performance under a real-time multicast session more efficient.

6. Conclusion

In this paper, we proposed a rate-adaptive real-time multicast system with LACRM and showed the effectiveness of a LACRM using simulation. We have already implemented a video frame generator and receiver on Windows95/NT, and now we are implementing a LACRM on FreeBSD and Windows95/NT.

In future work, we should consider some other things. First, we need more consideration of congestion detection, avoidance and recovery. Second we should consider how to initially join sessions, that is, which LACRM a receiver should access when he wants to join a certain session. This needs a session control protocol. Third is implementation using other layered or hierarchical encoding systems. In this implementation, we used a real-time socket interface that we developed, and this interface make the cost of implementing a LACRM for other encoding low. Fourth is the placement of a LACRM. The number of LACRMs and the appropriate placement to communicate efficiently are depends on the scale and the environment of the networks. We will evaluate these using our implementation.

References

1) I. Busse, B. Deffner, H. Schulzrinne, "Dynamic QoS control of multimedia applications based on RTP", Computer Communications, p49-58, 19, 1996.

2) R. Braden, Ed., L. Zhang, S. Berson, S. Herzog and S. Jamin, "Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification", RFC 2205, IETF, Sep. 1997.

3) H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, IETF, Jan. 1996.

4) H. Yokota, Y. Ito, M, Ishikura, S. Iisaku and T. Asami, "Implementation of Teleconference Control System using Real-time Transport Protocol," 51st National conventions of IPSJ, 5F-3, 1-199, Sep. 1995.

5) N. Shacham, "Multipoint communication by hierarchically encoded data," in Proc. IEEE Infocom'92, Florence, Italy, May 1992 vol. 3, pp. 2107-2114.

6) S. MCCanne, V. Jacobson, M. Vetterli, "Receiver-driven layered multicast," Proc. ACM Sigcomm '96, Stanford University, CA, pp.117-130, Sep. 1996.

7) S. Sakazawa, Y. Takishima and M. Wada, "A Coding and Transmission Control Scheme for Video Multicast," The Journal of the Institute of Image Information and Television Engineers, Vol. 52, No. 6, pp. 863-870, Jun. 1998.

8)Y. Ito, O. Maeshima, M. Ishikura and T. Asami, "A proposal of socket interface for real time communication," 54th National conventions of IPSJ, 4N-7, 3-427, Mar. 1997.

9) "Windows Sockets 2 Application Programming Interface", ftp://ftp.microsoft.com/bussys/winsock/winsock2/WSAPI22.DOC, May, 1996.

10)http://www.cadence.com/alta/products/newdatasheets/designer.html