論 文

Centralized Protocol Model for Videoconference Service over Wide-Area-Networks*

Yang Hee Choi*   *Regular Member*

광역망에서의 영상회의를 위한 중앙집중식 프로토콜 모델

正會員 崔 陽 熙*

要 約

본 논문은 광역 통신망에서의 비디오회의를 위한 중앙집중식 프로토콜 모델을 제시한다. 이 모델은 사용자, 서버, 믹서의 세 요소로 구성된다. 서버는 호관리기능을 전부 취급하고, 믹서는 실시간 트래픽 혼합을 행한다. 제어와 데이타는 본모델에서 각각 분리된 커넥션을 따라 운반된다. 호관리 프로토콜, 미디어 트랜스포트 프로토콜, 멀티미디어 동기 프로토콜 등 새로운 프로토콜들이 정의되었다. 본논문은 또한 믹서의 기능설계도 포함하고 있다.

ABSTRACT

In this paper, a centralized protocol model for videoconference service over a wide-area-network is presented. The model is comprised of three distinct components : clients, server, and mixer. The server handles all call management functions, and the mixer performs realtime traffic mixing. The control and data are separated, flowing over separate connections in the model. A set of new protocols are defined : call management protocol, media transport protocol, and multimedia synchronization protocol. This paper also presents the functional design of the mixer.

*서울大學校 컴퓨터工學科
Dept. of Computer Eng., Seoul National Univ.
論文番號 : 93-184

I. Introduction

Rapid progress in computer and telecommunications technologies brought multimedia-based applications to the hands of the domestic as well as business users. The service domains are varying dramatically from simple telephone-like point-to-point service to sophisticated group (multiple users) oriented ones : computer supported cooperative working (CSCW), teleeducation, group editing of multimedia documents, and multimedia teleconferences are the primary applications under active development at present.

The introduction of high-speed wide-area-networks, ISDN(Integrated Services Digital Network) and Broadband ISDN for example, are happening in many countries, making 1-100 Mbps dedicated digital pipes available to the public at a reasonable cost. The sophisticated services which were only possible within a local area network such as multimedia database servers, real-time transaction processing, real-time video-conferencing etc. are now being extended to wide-area-networks. The most promising service in the age of high-speed networks is the videoconference service. In this multipoint service, every conference participant is equipped with a videoconference terminal through which one can send and receive images, texts, voices and moving pictures (one's own figure, or some video clip from a local storage) to and from multiple participants at remote sites in parallel and in real-time.

Videoconference over the wide-area-networks bears many technical issues to solve : call/conference management, routing and flow control, media synchronization, dynamic quality of service control and guarantee, low-cost but effective terminal equipment etc. Those topics have been treated in many research testbeds[1,2,3,4,5] so far. However, none of the previous works provided an integrated communication and protocol architecture suitable for wide-area-networks. In this paper, we propose an overall architecture for videoconference systems, protocols, and performances which are independent of data compression techniques but easily applicable and scalable to general high-speed wide-area-networks. In the proposed architecture (Fig. 1), the terminals for the conference participants (client) are simple ones so as to be distributed easily at low-cost. The complex call establishment and route computation are all carried out by the centralized conference server. While the control informantion flows between the clients and the server, the video in formation from the clients are fed to a mixer that performs the media mixing (create a single video stream out of the incoming multiple video streams to the mixer, for example), and the resultant stream is broadcasted to the clients via a multicast route from the mixer to the clients. The proposed architecture is different from the previously mentioned ones[1,2,3,4,5] in many respects ; separation of control path and data path, use of multicast path in the reverse direction, flexible communication and

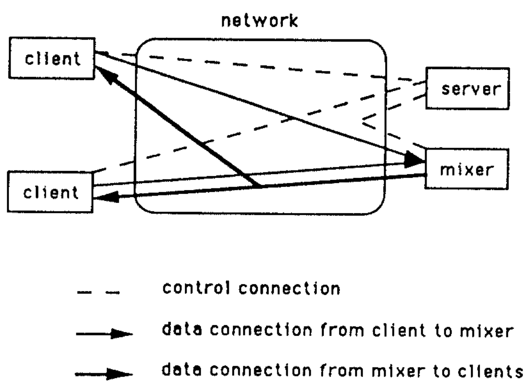

Fig 1. Communication Architecture of the Videoconference Service

protocol architecture that adapt and scale well with the evloving terminal and network techno logies.

This paper is organized as follows. In the next section, the communication requirements of the videoconference are identified. The protocols and the protocol architecture (server, client, and mixer) will be presented in the following sections. The laboratory experiment setup and future re search areas will be discussed in the final section.

Ⅱ. Requirements for Real-Time Videoc onference Service

The real-time videoconference service can be fully satisfied only when the appropriate functions are supported in the network, system, and conference application. The network and sys tem requirements are defined to support the videoconference service whose main features in our case are listed below.

- The number of conference participants can vary from two to several tens. This limit comes from the conference management viewpoint. If there are too many participants, it is better to organize the conference either in hierarchical fashion or in many parallel sub-conferences so that each participant only need to view a limited number (one or two) of windows on the screen.
- The conference is initiated by one of the participants whose primary role is the conference management(including floor control).
- Each participant may have systems of different capabilities, or may employ a set of media func tions different to each other.
- The number of participants, requesting com munication functions, and the quality of services can vary dynamically during a conference call.
- The conference service should be implement able on top of existing heterogeneous networking technologies(LAN, ISDN, Internent, and B-ISDN for example).

The network equirements that satisfy the above functions are[9]:

- Multicast network connections : Traffic from one participant should be able to be forwarded to multiple destinations. This is easily feasible in LAN environment, but difficult in WAN environ ment.
- Controlled Quality of Services for the connec tions : Bandwidth, delay and jitter are the main components to control in real time services.
- Fast Call Set-ups : The connections and participants should be added or deleted from the existing conference in the shortest delay.
- The ability to associate connections belonging to the same conference : This is not supported in today's network architectures.

The system's requirements for real-time video conference services are :

- Real time multimedia (audio, video) data com pression and decompression
- Friendly user interface for the service users
- The ability to cooperate with the network components to maintain the QoS targets.

It is also noted that the system requirements are dependent on the functions provided by the network, i.e. if the network provided only point to point connectivity between systems, the system should be capable of handling many connections in parallel, and in real-time.

Ⅲ. Protocol Architecture

The present day telecommunication networks do not easily support the network functions ident ified in the previous section. They are rather eas ily implemented in LAN environment though. There can be two alternative approaches for the design of the videoconference service in the con text of wide-area-networks : centralized and distributed. In the centralized scheme, each participant's system talk to the centralized conference server located within the network for the control and management of the conference call. The call initiator also calls the server to request the establishment of a conference call. Then the server does all the necessary control and computations required to set up the connections in the network. The multimedia traffic from each participant is fed to the centralized mixer that mixes the incoming traffic to produce a combined traffic broadcast to all the conference participants. The server and mixer are logically separate ones, and they are physically separated too in our scheme. This separation of data from control is vary common in telecommunication networks, but rare in computer networks(see Fig. 2).

In the distributed scheme, there are no server or mixer. Each participant can fully control the conferences, and does the job of multimedia mixing. The network only has to provide the connectivity between the participants (point-to-point or multipoint). This scheme has been popular in the LAN context, but not so attractive economically and technically in the WAN context. Therefore, we'll focus on the centralized approach from now on. The merits of the centralized approaches are identified at first.

- As the conference management(call, routing, QoS control, billing, etc.) is carried out at the server, the client's application software is largely simplified. Moreover the clients need not know the network characteristics (capabilities, status, cost etc.)
- The centralized server easily handles group managment(group creation, deletion, membership control etc.), name service (address resolution of shared resources, group addressing), and conference monitoring (number of active participants, active connections, used resources, and the conference owner for each on-going conference).
- The server, knowing the network's current usage and congestion status, can compute the optimal connection paths for the conference, increasing the network's effective capacity and improving the service performances at the same time. However the optimal path computation is not necessary when the underlying sub-network is capable of doing the computation and path set-up with its own network database server (as in the intelligent networking).
- By separating the control from the data, the dynamic call and resource control is easily achieved. The addition/deletion of participant/connection is a matter of a few control messages in the control connection.
- By accessing the server, one can obtain the information on existing conferences. A new mode of videoconferencing is possible in which one browses therough the currently available conferences (participants, duration etc.), and enters into one that interests him.
- The centralized mixer allows efficient resource sharing (mixing module needn't to be replicated at every client), optimal use of network bandwidth by multiplexing, sequenced packet delivery by global event ordering, and fast switching between traffic streams.
- When the number of conference participants are large, or when they are dispersed geographically, it is possible to employ multiple mixers in order to reduce the total bandwidth in the network.

The most interesting point, however, is the subnetwork-independence of the service architecture. The switching and transmission elements of the subnetwork in the centralized scheme does not need to be changed or upgraded for service introduction. The server and mixer does the necessary emulations (mulicasting, synchronization etc.). Thus with the rapidly evolving wide area network technologies, the service architecture is superior to the distributed one in terms of network management and service integration.

The overall centralized protocol model is depicted in Fig.2. The protocol architecture is composed of two separated planes : control and data. For the control plane, the server, the mixer, and the participating clients execute the conference control applications at the application layer(in the OSI sense) respectively. The data plane is for multimedia traffic that flows following the client mixer client path. The command/responses are exchanged between the two planes at the clients and the mixer. The commands mainly come from the server, and the responses also generally go back to the server for further processing.

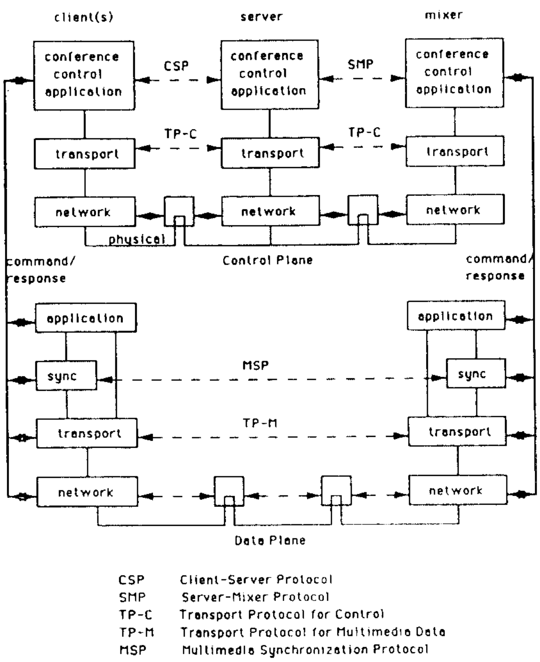

Fig 2. Centralized Protocol Model for Videoconference Service

In the control plane, a number of protocols are required. The CSP (Client-Server Protocol) is used to carry the conference call control information at the highest level. The SMP (Server-Mixer Protocol) allows the passing of connection information and mixing rules from the server to the mixer. TPC(Transport Protocol for Control) is a transport layer protocol providing reliable and efficient data transfer service to the application layer above. Simplified versions of OSI Transport Protocol Class 4 may be used for this purpose. The network layer is dependent on the underlying subnetwork, and is not included in our protocol model which is independent of the subnetwork's evolution.

In the data plane, MSP(Multimedia Synchronization Protocol) is defined. The videoconference traffic is normally composed of several media data (voice, video, data, graphics etc.). The non-ideal subnetwork may introduce variable bandwidth, delay and jitter to the channels used in the conference. The MSP resolves the synchronization problem by means of a simple delay adjustment algorithm (it is rather an algorithm than a protocol in a strict sense, however for the generality, we use the term protocol here.). The multimedia traffic is error controlled and flow controlled by the new TP-M (Transport Protocol for Multimedia Data) which is a simplified version of the proposed HSTP (High Speed Transport Protocol). TP-M handles the end-to-end transport-level QoS management as well as the normal data transport.

The centralized protocol model will allow a flexible yet efficient upgrade of the videoconference service because the functions are resident only on the end systems, not in the network components (switches, transmission systems etc.).

Ⅳ. Call Management Protocol

The conference management is carried out at the server, clients, and mixer. The conference call control is performed at three differenct levels : conference, system, and connection. A conference is set up among a number of end-systems, and each system manages a number of connections that are terminated at it. Call control at the conference level requires the complete knowledge of the controlled conference, and this is very easily done when the control is done at a central site(server in the centralized protocol model). The control at the system level is executed at each end-system participating in the conference. The main function at the system-level control is the grouping(association) of multiple connections for a conference and performing operations on them as a whole. The control at the connection level is performed at each system for each connection separately.

The control messages flow over the control channels that are setup in a point-to-point star fashion(server-client, server-mixer) at the start of a new conference or with the addition of a new participant.

If we consider managing a call equivalent to the management of informations associated to a call as seen by the call controlling entity, it is very important to define precisely the informations that are controlled at each level : conference, system, and connection. The informations are represented as contexts at the corresponding level, so we have Conference-Context, System-Context, and Connection-Context. The contexts are defined as follows.

Conference-Context=(Conf-id, {Client[i]}, {System-Context[i]})
System-Context=(System-Context-id, System-QoS-Context, {Connection[i]}, {Connection-Context[i]})
Connection-Context=(Connection-id, Connection-QoS-Context)
System-QoS-Context=(Sync-Parameters, …)
Connection-QoS-Context=(Peak-Data-Rate, Avg-Data-Rate, Delay-Bound, Loss-Bound, Jitter-Bound, Cost-Bound, …)

The Conference-Context is held and managed by the central server in our model. The System-Context is managed by each client (conference participant), and by the mixer (which is also a kind of end-system). The Connection-Context is also maintained at each system for the management of individual connections.

The Coinference-Context contains the conference identifier, the addresses of clients, and the System-Contexts for the clients. In the System-Context, its identifier, quality-of-services concerning the group of connections terminated at the system(such as Inter-connection synchronization requirements), and the identifiers of the connections for the system are present. The Connection-Contexts are also identified in the System-Context. The Connection-Context is composed of its identifier, and the quality-of-service parameters for the connection. In our case, QoS included the traffic description carried over the connection, such as data rate, burstness etc. It also includes performance parameters like delay, jitter, loss bound etc. The relationship among

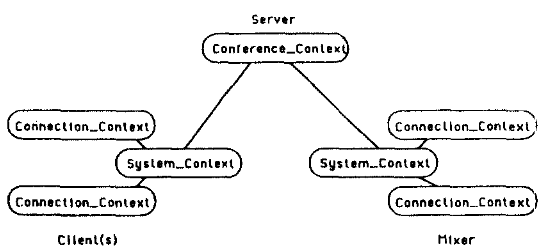

Fig 3. Relationship among Contexts

contexts is depicted in Fig. 3.

As the mixer in our case is treated as an end-system, it has its own System-Context, and Connection-Contexts.

The call management protocols (CSP, SMP in the model) are for the creation, modification, and deletion of the contexts and their parameters. The operations on the contexts and their parameters can be classified into four distinct ones : Establish, Release, Modify, and Status.

The Establish command is to request the creation of the concerned context in the system. The Release command is for the deletion of the referred context. The Modify command is used when the related context needs to be modified. For example, the addition of a new client, deletion of a client from the conference, addition/deletion of connections, change of mixing rules, change of QoS parameters etc. STatus signals are generated from all participating entities to inform others of its current status : performance or connectivity. The information in Status is often the source of subsequent Modify command whose aim is to correct/adjust the abnormal conditions reported in the Status signal.

Fig. 4 shows a typical scenario of a video conference call. A client requests establishment of a conference via the Establish(Conference-Context) command to the server. On receiving this request, the server first identifies the destination of the clients, and then sends Establish (System-Context) to them and the mixer. On receiving positive answers, the server now sends Establish(Connection-Context) to the clients and the mixer. If the connections are set up without problems, the conference can now start. During a conference, each system sends a Status signal to the server now and the. And one can send Modify(…Context) to the server, when it decides to change the context. The server examines the received Modify command, and then sends the Modify command to the related systems. The Release(…Context) is treated in the same way.

The server is assumed to maintain a global topology database and the actual link/system usages. With that, it can compute the optimal network path for the connections. How the paths are set up depends on the underlying network's capabilities. What we need for the centralized videoconference protocol model to be effective is point-to-point unidirectional network paths from the clients to the mixer, and multicast unidirectional path from the mixer to the clients. The server calculates the optimal network paths and passes the result to the clients/mixer in the Establish(Connection-Context) command[16]. This is similar to the source routing method in LAN. However, when the network does not allow source routing, the control of network paths can

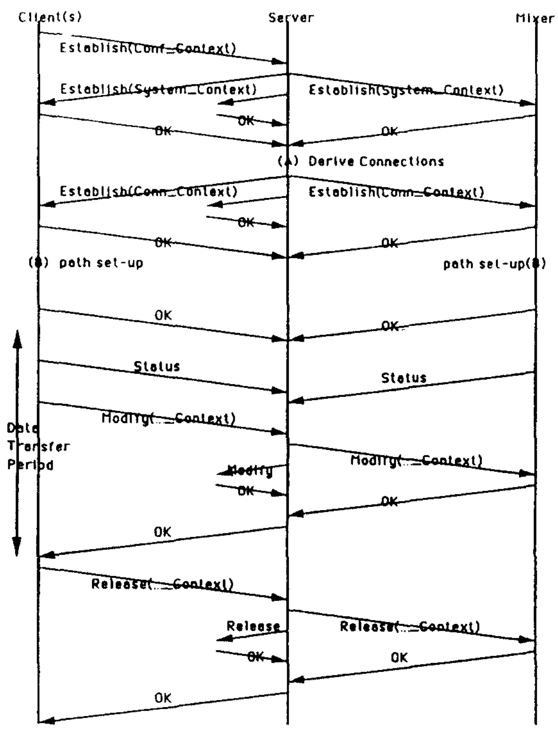

Fig 4. Call Management Protocol : Example Scenario

not be done using the Establish command. If the server can send/receive control commands directly from/to the subnetwork components (switches, gateways etc.) as in the case of intelligent networks, the optimal paths can be set up under the serever's direct control.

V. Media Transport Protocols

The videoconference traffic flows are carried over the client-to-mixer point-to-point connections, and processed at the mixer according to the mixing rules defined in the conference and system contexts. Then the mixed traffic flows back to the clients from the mixer over the mixer-to-client connections. In the forward direction (client-to-mixer), each client has several connections (audio connection, video connection, and data connection for example) defined under the same System-Context. These connections are terminated at the mixer, and the mixer performs media mixing on the traffic of the same type from the clients. Therefore, basic transport protocol functions are performed between the client and mixer. The transport functions are heavily dependent on the media type and mixing rules at the mixer.

In the audio media case, assuming that traffic is generated periodically at the sourcess (clients), synchronization is required at the receivers (mixer or clients). Simple forward flow control (i.e. rate control at the source only) will suffice for the traffic regulation, and no error control is required. Audio signals are normally combined (overlapped) at the mixer, and the bandwidth for the mixer-to-client connection is the same as that for each incoming audio connection to the mixer.

In the video media case, the overlapping of signals is not desirable, and the mixing rules may be a) passing a single video stream, b) passing multiple video streams, or c) passing all video streams from the clients. It is obvious that the mixer-to-client video connection bandwidth is proportional to the number of video streams we want to distribute. In present day systems, each client views the client who is holding the floor. In this case the mixer passes only the video traffic for the designated client. The designation can be done in many different ways ; the loudest speaker, token passing, voting etc.

The transport functions for the video media are synchronization and forward flow control. No error control is required as the video traffic is real-time.

For the data connections, the mixer should not modify the contrents, and should pass all traffic. Error control is required, but no synchronization function in necessary. Flow control is new window-based due to the asynchronous nature of the data traffic.

From the discussions above, we need three different types of media transport protocols for the videoconference service : one for audio, one for video and another for data. The simplified High-Speed Transport Protocol (HSTPP)[14,15] is sufficient for the first two types, and the simplified Transport Protocol(TP) Class 4 of ISO [10] is also suitable for the data connections.

As the videoconference service deals with a large amount of real-time traffic from many sources, it is important to distribute the control functions at appropriate points in the network in order to guarantee the specified QoS parameters in the contexts. Fig.5 shows that the clients and the mixer collaborated for QoS control employing various techniques ; input rate control for audio/video traffic at the traffic originators, synchronization at the traffic receivers, window flow control for data connection at the receivers, and QoS/traffic status monitoring/reporting at every point[8,17].

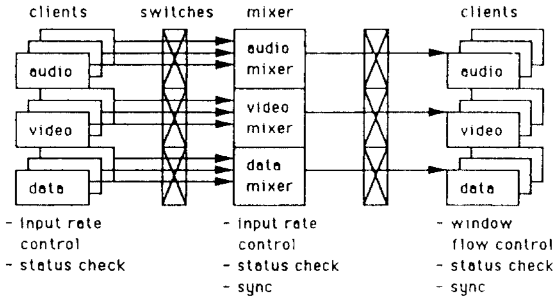

Fig 5. Distribution of QoS Control Functions

We now turn to the mixer[7,11]. Fig.6 shows the internal structure of a mixer, particularly for the audio/video connections. The traffic flowing from the clients are buffered and synchronized (see next section for details) after the mixing module. The output of the synchronization module is again rate controlled before sending (to the clients). For the synchronization, the related control parameters are specified in the System Context, and the measured performances are signaled back to the server for further processing if required The mixing rules are also specified in the System-Context, and the mixing module also generates status signals. The rate control is easily done with a simple leaky bucket scheme[18] whose parameters are given in the mixer's System-Context too.

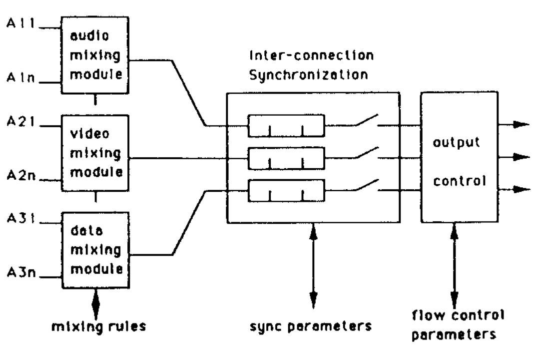

Fig 6. Mixer Organization

In the following informal statement, the mixing rule is defined.

(Mixing Rule) :

input : B1, B2, ··· Bn

output : case operation=pass single, then output is Bi,

case operation=pass-multiple, then output is {Bi},

case operation=pass-all, then output is {Bi}, i=1···n

where the value(s) of i is defined according to the Context.

The selection of i can be done locally at the mixer or by the message from the server over the control connection. If the loudest speaker holds the floor, the value of i comes from the audio mixer to the video mixer for the display of the speaker's face to all. Token control or voting are easily done by the server, but they are slower than the local decision scheme.

It is preferable that the mixing module be capable of ordering the incoming packets by their generation time. For this purpose, various algorithms have been proposed [7]. However, in the videoconference application, it is not considered critical if packets are misordered by several time slots, and we simply mix the packets by their arriving time to the mixing module.

Ⅵ. Media Synchronization Protocol

In the centralized protocol mode, the multimedia synchronization protocol is required between the clients and the mixer, because the traffic flow is altered (by mixing) by the mixer. This is not the case in the distributed protocol model, where the media synchronization is done on an end-to end basis between clients.

The media synchronization function is classified in three different modes : intra-connection, inter-connection, and inter-client. The intra-connection synchronization is performed at the receiver, to recover the original timing relations (at the source) between successive pakets of a connection. The intra-connection synchronization is needed for the real-time·continuous media data (voice and video). The inter-connection synchronziation is to recover the original timing relations between packets of different connections. This is performed between source and destination clients for the parallel connections linking them together. In the video conference service, inter-connection synchronization is performed between audio, video and sometimes data connections. The inter-client synchronization is to recover the global timing relations of packets from multiple sources. For example, data packets are delivered in order, to the receiver's application by their global departure time at the sources.

When there's no global clock, and the network delay between the source and destination is variable, it is impossible to determine the global order of received packets. Hence, inter-client synchronization is often omitted in the videoconference wher global synchronization is not absolutely needed.

The intra-connection synchronization scheme we adopted for the videoconference is depicted in Fig. 7. It is assumed that the variable length packets are generated at every T seconds for the continuous media. This is true fo the data compression standards (JPEG, MPEG, and H261). The problem here is how to align the arrived packets with variable jitter and delay so that they are delivered to the application at every T seconds exactly. In the proposed scheme, this is achieved by introducing an artificial (intentional) delay between the incoming packets (at point A in the figure) and the application (B in the figure) so that the sum of network delay and the artificial delay is constant for all packets. The artificial delay should be long enough to compensate the maximum allowable jitter, and short enough to avoid service degradation.

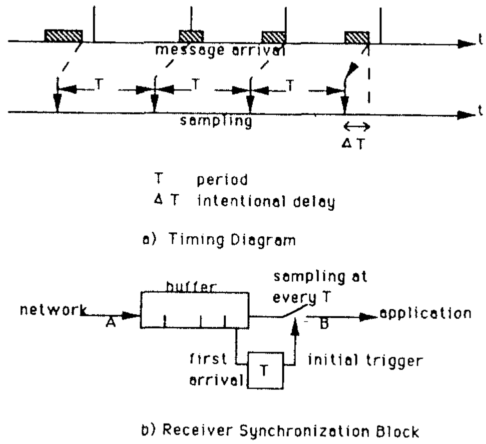

Fig 7. Intra-connections Synchronization based on Intentional Initial Delay

The intentional delay is not computed for every packet. Instead, only the first arrived packet in the receiving buffer is delayed by ΔT, a value which may cover most jitters. The subsequently arrived packets are sampled every T seconds after the first one. In order to distinguish lost or late packets, sampling is performed on the packets with correct sequence numbers only. It is noted that the buffer length is one or two in normal cases where the jitter is not excessive. The value T and ΔT are passed to the receiver via the control connection in our case.

The problem of a wrong ΔT can be rectified by monitoring the packet loss rate, and adjusting the sampling time in Fig. 7 accordingly. If ΔT is too short, packet loss rate by the empty buffer will be high, and inserting additional delay at B solves the problem. If ΔT is too long, the packet loss by buffer overflow will be high, and this is solved by shortening one sampling cycle at B.

For the inter-connection synchronization, a similar scheme as shown in Fig. 8 is proposed for the videoconference service. The packets arriving at the synchronization module with jitter (A1, A2,…An) are first buffered and then sampled at regular intervals producing synchronized packet streams at B1, B2, … Bn. Each connection performs intra-connection synchronization independently with different sampling period T1, T2,… Tn. However, whenever packets between connections need to be synchronized, they are sent at the source at the same instant with the marker bit on in the header. The first packet on each connection is alwasy marked, and the intentional delay is incurred only to one particular connection (reference connection). The other connections put the same amount of delay as their intentional delay. In other words, the first sampling at the reference connection acts as the global trigger signal for sampling at all connections. This permits the initial synchronization between connections, and each conncetion samples packets with its own timing Ti afterwards until another packet with marker is received. When a packet with a marker is at the head of the conncetion i's buffer, and Enable(i) signal from the buffer is put to True. If Enable(r) signal at the reference connection's buffer is also True, The sampling point Bi is resynchronized to the first sampling pulse of the reference connection after Enable(i). AND. Enable(r)=True. Therefore the inter-connection synchronization occurs only between designated connections at designated instants. This is very flexible in contrast to the previously reported ones[12] that requires periodic synchronization. As in the case of intra-connection synchronization, a sequence number is required in order to copy wth lost or late packets.

The synchronization parameters Ti, ΔT, QoS are informed to the synchronization module through the control connection. The performance of synchronization can be measured by packet loss rate at the buffer, and the corresponding connection's bandwidth should be enlarged if the loss is excessive. This adjustment can be done by the Modify command.

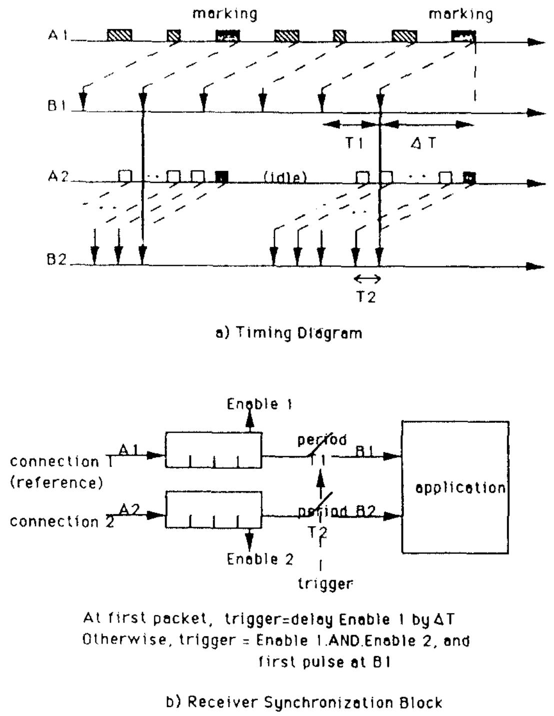

Fig 8. Inter-connection Synchronization based on Intentional Initial delay with Marking

Ⅶ. Conclusion

In this paper, a centralized protocol model for the videoconference service in a wide-area-network is presented. The proposed architecture is composed of three distinct components : client, server, and mixer. The central server handles call management, resource allocation, and QoS control in the centralized manner. The control signals flow over the point-to-point control connections between the server and the other components. The multimedia traffic flows, however, on separate connections between clients via the mixer. The mixer's role is to merge the traffic from the sources, and then produce streams to be delivered back to the clients according to the mixing rule identified in the conference request command.

The protocols needed in the model are call control protocols (client-server, server-mixer), transport protocol for call control data, media transport protocols (voice, video, and data), and media synchronization protocol(intra and inter-connection). The conventional high-speed transaction processing system can be used for the server, but for the mixer, a new system architecture is defined.

In a practical implementation, the server may be a physically separate one from the mixer, or they can be implemented as one integrated central system. However, the logical distinction between the server and the mixer should be respected in any case. As the network size and the number of conference participants grow, it is preferred to have multiple mixers in the network in order to minimize the total cost and to have better performances. The separation of the server from the mixer is inevitable in a multi-mixer environment.

The proposed centralized protocol model is going to be experimented in the SMART(Seoul Multimedia Advanced Research Testbed)[13], at the Seoul National University. As the model's main contribution is in the separation of data and control, the experimental testbed will be built in a LAN environment, with a WAN simulator(see Fig. 9). The testbed is to demonstrate the feasibility of the protocol model, and to assess the newly defined protocols' peroformances.

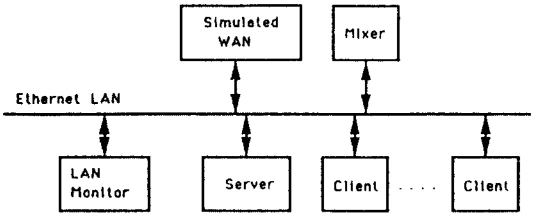

Fig 9. Architecture of Videoconference Testbed

The centralized protocol model is not absolutely necessary when the network bandwidth is abundant, and the client's data handling capacity is enough to handle full mesh connections to other clients. This approach is popular in the LAN context, but will be very costly in the WAN context. In the fully distributed protocol, the server's functions will be handled by the call initiating client. And the mixing is done by each client for the received packets from all other clients. The media transport protocols, and the multimedia synchronization protocol presented in this paper can be easily adapted to the fully distributed model.

참 고 문 헌

1. W.J. Clark, "The European MIAS System for ISDN Multimedia Conferencing," Multimedia '92.
2. S. Masaki et al., "A Promising Groupware System for Broadband ISDN : PMTC," Multimedia '92.
3. S.R. Ahuja, J.R. Ensor, "Call and Connection Management : Making Desktop Conferencing System a Real Service," Multimedia '92.
4. T. Ohmori et al., "Multiparty and Multimedia Desktop Conferencing System : MERMAID," Multimedia '92.

5. M. Chen, T.Barzilai, H.M. Vin, "Software Architecture of DiCE : A Distribued Collaboration Environment," Multimedia '92.
6. Yanghee Choi, "Research Network Testbed : SMART(Seoul Multimedia Advanced REsearch Testbed)," ICCS/ISITA '92, Singapore, 1992.
7. P.V. Rangan, H.M. Vin, S. Ramanathan, "Communication Architectures and Algorithms for Media Mixing in Multimedia Conferences," IEEE/ACM Trans. on Networking, vol. 1, no. 1, Feb. 1993.
8. D. Ferrari, "Real Time Communication in an Internetwork", Journal of High Speed Network, vol.1, no.1, 1992.
9. CCITT I.374, Network Capabilities to Support Mulitimedia Services, 1992.
10. ISO 8073, Transport Service Protocol, 1983.
11. CCITT X.6, Multicast Services and Audiographic Conferencing, April 1992.
12. T.D.C Little, A. Ghafoor, "multimedia Synchronization Protocols for Broadband integrated Services," IEEE JSAC, vol.9, no.9, Dec. 1991.
13. S.C. Park, Yanghee Choi, "Multimedia Communication Architecture in SMART (Seoul Multimedia Advanced Research Testbed," Multimedia Communication '93, Banff, Canada, 1993.
14. XTP Protocol Definition R3.6, Protocol Engines Inc. Jan. 1992.
15. ISO/IEC JTC1/SC6/WG4 N811, Draft Text fo High Speed Transort Protocol Specification, Dec. 1992.
16. S.E. Deering, D.R. Cheriton, "Multicast Routing in Datagram Internetworks and Extended LANs," ACM Trans. on Computer Systems, vol.8, no.2, pp.85 110, May 1990.
17. H. Zhang, S.Keshav, "Comparison of Rate based Server Disciplines," Proc. ACM SIGCOMMM '92, 1991.
18. M. Sidi, et al, "Congestion Control through Input Rate Regulation," Proc. Globecom '89, Dallas, 1989.

崔 陽 熙(Yanghee Choi) 종신회원

1955년 7월 27일생

1975년 2월 : 서울대학교 공과대학 전자공학과(학사)

1977년 2월 : 한국과학원 전기 및 전자공학과(석사)

1984년 6월 : 프랑스 국립 전기통신대학 전산과(공학박사)

1977년~1979년 : 한국전기통신연구소 연구원

1981년~1984년 : 프랑스 국립 전기통신연구소 연구원

1988년~1989년 : IBM Thomas J. Watson Research Center 방문연구원

1984년~1991년 : 한국전자통신연구소 책임연구원

1991년~현재 : 서울대학교 컴퓨터공학과 조교수, 서울대학교 중앙교육연구전산원 부원장