INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTÈMES ALÉATOIRES

PUBLICATION INTERNE

N^{o} 978

MULTI-VIDEO-SCHEME VIDEOCONFERENCING APPLICATIONS

BERNARD COUSIN AND ZHOU JUNHUI

ISSN 1166-8687

IRISA

CAMPUS UNIVERSITAIRE DE BEAULIEU - 35042 RENNES CEDEX - FRANCE

IRISA
INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTÈMES ALÉATOIRES
Campus de Beaulieu – 35042 Rennes Cedex – France
Tél. : (33) 99 84 71 00 – Fax : (33) 99 84 71 71

Multi-video-scheme Videoconferencing Applications

Bernard Cousin* and ZHOU Junhui

Programme 1 — Architectures parallèles, bases de données, réseaux
et systèmes distribués
Projet Adp

Publication interne n ° 978 — December 1995 — 16 pages

Abstract: The videoconferencing applications transmit video and audio data through data networks. The current networks such as the Internet are heterogeneous: their subnets may possess different bandwidths. When a videoconference covers subnets of different bandwidths, multiple video coding schemes of different bandwidth requirements can be used at the same time to make good use of the bandwidth resources. Sometimes difference of host processing power also leads to using multiple video schemes. In this paper, we discuss multiple video transport sessions of multi-video-scheme (*i.e* multiple video scheme) videoconferencing applications. Video data in different schemes must be transmitted over different multicast trees. Thus the multi-video-scheme application creates multiple video transport sessions. We describe the architecture for the multi-video-scheme videoconferencing application as an open system. It consists of mono-video-scheme subsystems with some gateways. Gateways translate and forward video data between the mono-video-scheme subsystems, and forward audio data between these subsystems. We discuss translation for different video schemes in two way: direct conversion, and "first to decode then to encode". Finally we present a multi-video-scheme videoconferencing application IVS-H261/JPEG ** on the Internet. IVS-H261/JPEG consists of two mono-video-scheme subsystems IVS-H261 and IVS-JPEG with a gateway IVS-SERV.

Key-words: videoconferencing, video scheme, congestion control, Internet.

(Résumé : tsvp)

*e-mail: {bcousin, jzhou}@irisa.fr
**IVS : INRIA Videoconferencing System.

CNRS INRIA
Centre National de la Recherche Scientifique
(URA 227) Université de Rennes 1 – Insa de Rennes

Institut National de Recherche en Informatique
et en Automatique – unité de recherche de Rennes

Applications de vidéoconférence à multi-formats vidéos

Résumé : Les applications de téléconférence transmettent des images et des sons numériques en utilisant les réseaux informatiques. Les réseaux informatiques actuels (par exemple Internet) sont hétérogènes : les sous-réseaux qui les composent peuvent présenter des caractéristiques (notamment de débit) différentes. Quand les participants à une même téléconférence sont répartis sur des sous-réseaux offrant des caractéristiques différentes, il peut être nécessaire d'utiliser simultanément différentes techniques de codage de la vidéo afin de respecter les limitations de débit. De même, des différences de puissance de traitement entre les stations des participants à une même téléconférence peuvent amener à utiliser simultanément plusieurs techniques de codage. Dans ce papier, nous étudions les applications de téléconférence multi-formats, c'est-à-dire utilisant des formats vidéos multiples. Ces applications ouvrent simultanément plusieurs sessions de transport pour transmettre la vidéo. En effet, les différents flux vidéos (correspondant aux différents formats) sont diffusés sur différents arbre de diffusion afin d'interconnecter au mieux les différents sous-réseaux. Nous présentons alors l'architecture que nous proposons pour de telles applications dans un environnement hétérogène. Elle est basée sur des sous-systèmes mono-formats interconnectés grâce à des passerelles spécifiques. Ces passerelles, permettent l'échange des données et des commandes entre les sous-systèmes mono-formats en transformant leur formats et en interprétant leur sémantique lorsque cela s'avère nécessaire. Deux méthodes de traduction des formats vidéos sont proposées : une traduction directe, et une traduction indirecte utilisant la mémoire vidéo. Finalement nous présentons notre application de téléconférence multi-format IVS-H261/JPEG.

Mots-clé : vidéoconférence, format video, contrôle de congestion, Internet.

1 Introduction

Recent years have been seeing rapid development of videoconferencing technology. It has become a very important aspect in the area of computer communications. The videoconferencing application acquires, transmits and presents real time continuous media data including video and audio data. It allows people at different locations to take part in a videoconference through data networks. This will bring great facility to us.

Videoconferences can be divided into interactive conferences and non-interactive conferences. The former allow any participants to send data, and the latter only allow the speaker of the conference to do so. Interactive conferences have more strict quality of service (QoS) requirements than non-interactive conferences. For example, interactive conferences need to limit transport delay to prevent the people from feeling uncomfortable.

Because real time video have huge data rate (uncompressed video data rate in CIF format is 36.25 Mbits/s [16]), it can only be transmitted in the networks after being compressed. There are numerous video compression techniques. Among them we can find intraframe compression coding schemes adopted by coding standards such as JPEG [5], or interframe compression coding schemes adopted by coding standards such as H261 [6] and MPEG [7]. Interframe compression achieves greater compress ratio than intraframe compression because interframe compression takes advantage of correlation of consecutive frames while intraframe compression only treats individual frame. But obviously, interframe compression needs much more processing power than intraframe compression.

Usually, videoconferencing applications use only one video coding scheme. But the current networks such as the Internet are heterogeneous: their subnets may possess different bandwidths. On the other hand, different coding schemes generally generate different bandwidth requirements under the same conditions. And greater bandwidth a scheme needs, higher video quality it achieves. When a videoconferencing application covers subnets of different bandwidths, we can use multiple video schemes at the same time to make good use of the bandwidth resources and to achieve the best videoconference service.

Sometimes difference of host processing power leads us to using multiple video schemes. For example, some hosts are equipped with specific video hardware which can process video of high quality. But such video hardware is not always available for all participants. Those who have not the video hardware may use another video scheme implemented in software. In this case, multiple video schemes are used to adapt to different host environments.

Therefore, to use multiple video coding schemes in a videoconferencing application is possible and sometimes required.

Multiple video schemes can be considered to be generalization of layered coding [14]. Layered coding generates two types of streams: base stream and high quality stream. Base stream is corresponding to low quality video, while base stream + high quality stream corresponding to high quality video. Their different bandwidths and processing power requirements make them be suitable to different networks and hosts. If we substitute low quality video and high quality video by video in different schemes, layered coding will become multiple video schemes.

In next section, we will discuss multiple video transport sessions and the system architecture of multi-video-scheme videoconferencing applications. Since multiple video schemes are used in the same videoconferencing application, and generally different video schemes don't have interoperability, translation between them is required. Section 3 will describes translation for different video schemes. In Section 4, we present a multi-video-scheme videoconferencing system IVS-H261/JPEG. Two mono-video-scheme subsystems, the gateway, congestion control mechanism in IVS-H261/JPEG are described. Section 5 concludes the paper.

2 Multi-video-scheme architecture for videoconferencing applications

2.1 Multiple video transport sessions

We first discuss the multiple video transport sessions of multi-video-scheme videoconferencing applications.

The videoconferencing application transmits data in multicast communication mode [8]. All the participants spread over a multicast tree spanning the network. Any participant can send data to all the other participants and receive data from any other participant through the multicast tree. A multicast tree T can be identified by its group address $\{group\_address\}$. On the Internet, multicast group address is IP address of class D.

During the videoconference, a video transport session is created between the participants over the multicast tree. Fig.1 shows video data streams in a video transport session where some of the participants send video data (for the non-interactive conference, only one participant (*i.e* the speaker) sends video data).

Obviously, the mono-video-scheme videoconferencing application creates one single video transport session. All the participants are in the same session. Multiple coding

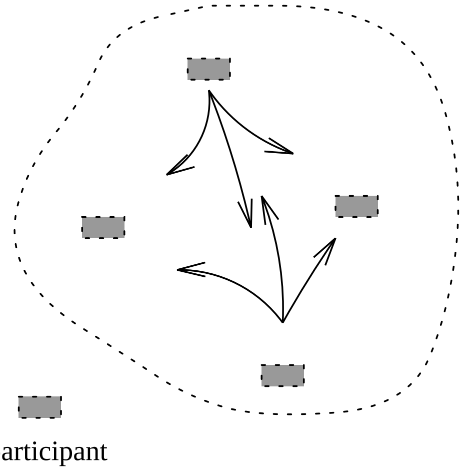

Figure 1: Video transport session in a videoconferencing application.

scheme applications may also create single transport session. For example, a system adopts multiple coding schemes, but only one coding scheme is used at any one time, and different coding schemes can only be used at different times. Such applications are of single transport session type.

Since the use of multiple video schemes is to enable videoconferencing applications to be suitable to different network and host environments, and besides different video schemes don't have interoperability, transmissions of video data in different schemes are definitely simultaneously performed over different multicast trees T_1, T_2, T_3, ... , and T_N. N is the number of video schemes used. Therefore the application will create N video transport sessions in the network. Therefore multi-video-scheme applications are multiple video transport session applications.

In case of absence of video translator to translate and forward video data, the N sessions created by the multi-video-scheme application are separate. The participants in one session cannot send/receive video data to/from another. They seem to be independent video transport sessions. In order to connect the separate sessions, translators are required. In fact, to connect the separate sessions is to translate and forward video data between the sessions. The number of the translators to connect every two sessions between the N sessions will be

$$C_N^2 = \frac{N \cdot (N-1)}{2}$$

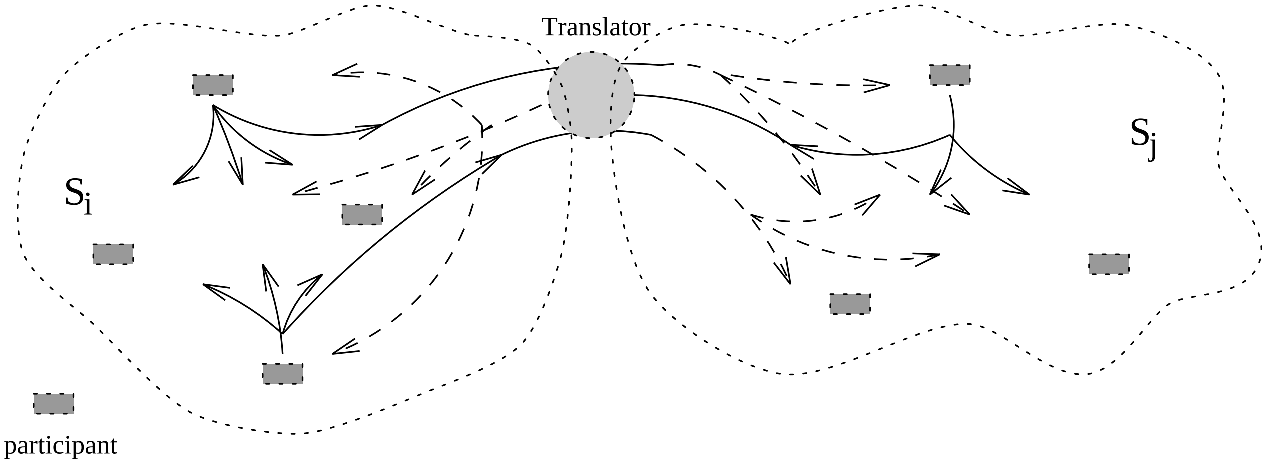

Figure 2: Two video transport sessions with a video translator.

The translator must join in the two multicast trees corresponding to the two sessions to be connected by it. Suppose a translator G_{ij} is to connect two sessions S_i and S_j of which the corresponding multicast trees are T_i and T_j, and the corresponding video schemes V_i and V_j, respectively. G_{ij} will be a new node in both T_i and T_j, as well as a commun node in the two trees.

The translator G_{ij} receives video data from both S_i and S_j. It translates the data from S_i into the scheme V_j then forwards to S_j, and simultaneously translates the data from S_j into the scheme V_i then forwards to S_i. Thus the participants in one session are now able to send/receive video data to/from the other. We will discuss translation from one video scheme to another in next section. Fig.2 shows two video transport sessions with a video translator. In each session, between the participants, the data streams are as in Fig.1. The translator translates and forwards the data stream for each data sender from one session to the other.

Through the video translators, all the sessions are connected with each other. Any participant can send video data to all the participants, and can receive video data from any other participant in the conference.

Non-interactive conference is a simplified case but is probably the typical application case for multi-video-scheme videoconferencing. In non-interactive conference, only one session, where the speaker of the conference is, needs to be connected with

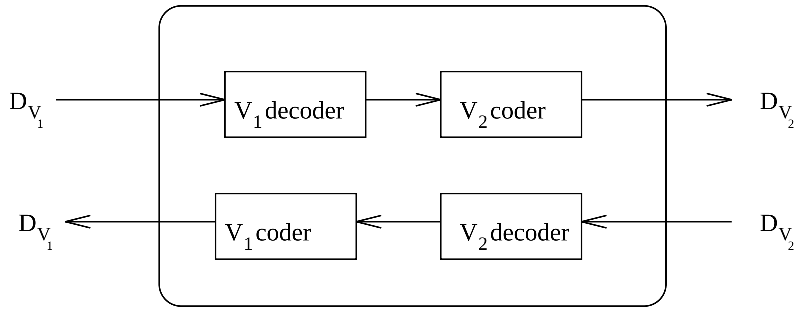

Figure 3: Video translator between the video schemes V_1 and V_2.

every other session. Thus only $N-1$ translators are required for non-interactive N-video scheme conference.

2.2 System architecture for multi-video-scheme videoconferencing applications

Multi-video-scheme videoconferencing applications can have two types of system architectures. One is one single system with multiple video scheme codec. The other is multiple subsystems each of which contains one single video scheme codec. The former may make the programme code of the application become too large due to many schemes adopted and be difficult to maintain. The latter has not such drawback. More important, the latter gives an open multi-video-scheme videoconferencing application system structure. We can use some existing systems to implement the multi-video-scheme videoconferencing application. This is just the approach to implement our multi-video-scheme videoconferencing application IVS-H261/JPEG. Therefore later on, we will consider that the multi-video-scheme videoconferencing application consists of multiple usual videoconferencing systems in different video schemes with some gateways. The gateways implement the video translators and can be considered to translate and forward video data between these mono-video-scheme subsystems.

A videoconferencing system transmits audio data over the same multicast tree as video data. Thus the multi-video-scheme videoconferencing application also creates multiple audio transport sessions. The gateways need to forward audio data between the audio transport sessions. We stipulate that the subsystems in a multi-video-scheme application have audio interoperability. Audio translation is not required.

The subsystems in a multi-video-scheme application should have the same architecture. Otherwise the complexity of the system increases unnecessarily, and it may be difficult for the subsystems to implement commmunications. The gateways between the subsystems should also have the same architecture as the subsystems. Thus all the mono-video-scheme subsystems are similar except for their different video codecs, and all the gateways are similar except for their different video translators. This implies the optimized way to implement a multi-video-scheme videoconferencing application is to develop a mono-video-scheme system or choose an existing mono-video-scheme system then to substitute the video codec in it by other video codecs to achieve other mono-video-scheme systems, and in the same way to develop gateways for two systems then to substitute the video translator in it by other video translators to achieve other gateways.

Consider system complexity, bandwidth characteristics of the current networks, and other factors. Two-video-scheme videoconferencing applications may be a good trade-off at the present. A two-video-scheme application consists of two mono-video-scheme subsystems and a gateway. One subsystem generates high quality video of large bandwidth requirement, and on the contrary, the other generates low quality video of small bandwidth requirement. The former is suitable to the subnets that can meet the large bandwidth requirement, and the latter is suitable to the other subnets.

3 Translation for two video schemes

Translation for two different video schemes V_1 and V_2 can be achieved through conversion. If we denote video data in V_1 and V_2 as D_{V_1} and D_{V_2}, respectively, then we have

$$D_{V_1} = \Psi_{V_2-V_1}\{D_{V_2}\} \ ;$$

$$D_{V_2} = \Psi_{V_1-V_2}\{D_{V_1}\} \ .$$

$\Psi_{V_2-V_1}$ and $\Psi_{V_1-V_2}$ represent the conversion from V_2 to V_1 and that from V_1 to V_2, respectively. Direct conversion is an effective way to translate video data. However in fact, $\Psi_{V_2-V_1}$ and $\Psi_{V_1-V_2}$ don't always exist for some schemes. To study on conversion between specific video schemes is significant. But that is beyond the scope of this paper.

Another way to translate between two video schemes is first to decode then to encode, that is, first to decode video data in one scheme to get original data, then to encode in the other scheme. Conversion may not be implemented for some schemes. But this way can always be implemented. Such translating processes can be described as two directional pairs:

$$\Theta_{V_1-V_2} : < V_1 \; decoding,\; V_2 \; encoding > ;$$

$$\Theta_{V_2-V_1} : < V_2 \; decoding,\; V_1 \; encoding > \; .$$

The pair $\Theta_{V_1-V_2}$ translates video data from V_1 into V_2, and the pair $\Theta_{V_2-V_1}$ translates from V_2 into V_1. Of course $\Theta_{V_1-V_2}$ and $\Theta_{V_2-V_1}$ could also be considered to be of a type of special conversion.

Obviously, $\Theta_{V_1-V_2}$ and $\Theta_{V_2-V_1}$ may need certain extra time. Greater the processing power of the translator host is, less the time overhead is. Thus the translator host should be well chosen. Otherwise translation in this way may take much time and influence the quality of service, especially for interactive conferences. The video translator to implement $\Theta_{V_1-V_2}$ and $\Theta_{V_2-V_1}$ is shown in Fig.3.

4 Multi-video-scheme videoconferencing application IVS-H261/JPEG

IVS-H261/JPEG is a multi-video-scheme videoconferencing application in the Internet. It consists of two mono-video-scheme videoconferencing system IVS-H261 [17] and IVS-JPEG [18] with a gateway IVS-SERV. IVS-H261 and IVS-JPEG had been developed successively, then we have developed IVS-SERV to enable them to become a multi-video-scheme videoconferencing application.

4.1 Mono-video-scheme systems IVS-H261 and IVS-JPEG

Both IVS-H261 and IVS-JPEG use the Real-Time Transport Protocol (RTP) [19] to transmit video and audio data. RTP is proposed by the Internet Audio-Video Transport Working Group to implement real time video and audio data transport in packet networks such as the Internet. RTP is comprised of two sub-protocols : RTP Data Transfer Protocol and RTP Control Protocol. The former provides data delivery service including data type identification, data packets sequence numbering, and data samples timestamping. The latter transports periodically participant information and monitors the quality of service (QoS). On the Internet, RTP is typically performed on top of the User Datagram Protocol (UDP). Video transport session, audio transport session and RTCP session can differ from one another through different UDP ports. Therefore a session S_i can be defined by the IP group address

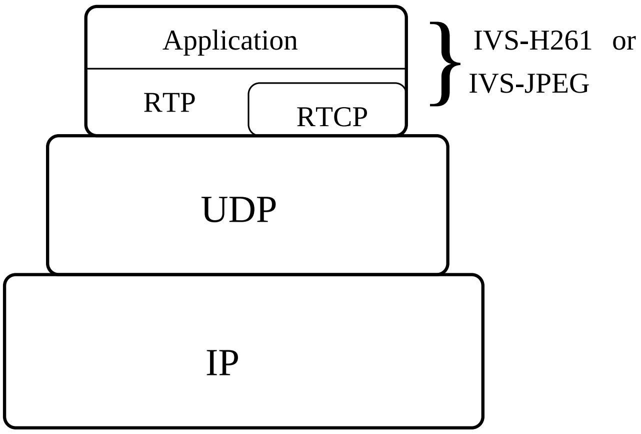

Figure 4: Architecture of IVS-H261 and IVS-JPEG.

$@m$ plus a UDP port p_i as follows:

$$S_i : \{@m, p_i\} .$$

The architecture of both IVS-H261 and IVS-JPEG is shown in Fig.4. RTP is integrated into the system. The system can be logically divided into two layers: RTP layer and application layer. The latter includes video codec, audio codec, and user interface.

IVS-H261 and IVS-JPEG adopte the video scheme H261 and JPEG, respectively. In IVS-JPEG, the video codec is hardware codec in XVideo Parallax card. Its frame rate can arrive at 30frames/s, and meanwhile the video throughput is about 1,500kbits/s. IVS-JPEG is suitable to well equipped hosts and broadband subnets of the Internet. In IVS-H261, the video codec is implemented in software. The video throughput of IVS-H261 is relatively low (about 30kbits/s in CIF). This is because, on the one hand, H261 is an interframe compression scheme which can gains a great compression ratio, and on the other hand, the processing power of current workstations is limited (the video frame rate of IVS-H261 is only 5-10 Frames/s). Thus IVS-H261 is suitable to narrow-band subnets of the Internet.

4.2 Gateway for IVS-H261 and IVS-JPEG

The gateway IVS-SERV is supposed to have the same architecture as IVS-H261 and IVS-JPEG. It will use RTP on top of UDP. In a videoconference, IVS-H261 creates a group of sessions {video transport session S_v^H, audio transport session S_a^H, RTCP session S_c^H}, and IVS-JPEG creates another group $\{S_v^J,\ S_a^J,\ S_c^J\}$. We denote them respectively as

$$\{\{@m^H,\ p_v^H\}\ ,\ \{@m^H,\ p_a^H\}\ ,\ \{@m^H,\ p_c^H\}\}\ ;$$

$$\{\{@m^J,\ p_v^J\}\ ,\ \{@m^J,\ p_a^J\}\ ,\ \{@m^J,\ p_c^J\}\}\ .$$

$$m^H \neq m^J\ .$$

The gateway will join in both multicast trees $@m^H$ and $@m^J$, translate and forward video data between S_v^H and S_v^J, forward audio data between S_a^H and S_a^J, and also forward participant information delivered by RTCP between S_c^H and S_c^J.

To forward audio data and to forward participant information between two sessions are to simply forward packets between two sessions without other processing. However the gateway will receive the packets forwarded by it. For example, the gateway forwards a audio data packet from S_a^H to S_a^J, then the gateway will receive it again in S_a^J. At this time, the gateway should not forward it to S_a^H. Otherwise, catastrophic forwarding loops will be created: the packet is continously forwarded from this session to that session, then reversely. In order to recognize the packets forwarded by the gateway, the gateway can check the source address of each packet received. If the source address of a packet is identical to the host address of the gateway, it is certainly forwarded by the gateway, and will be ignored. The algorithm to forward packets between two session S' and S'' is as follows:

Begin: the gateway receives a packet in S'.

1. Check whether or not the source address of the packet is identical to the host address of the gateway, if yes, goto 3;
2. Forward the packet to S'';
3. End

The gateway uses the directional pairs

$$<\ JPEG\ decoding,\ H261\ encoding\ >\ ;$$

$$<\ H261\ decoding,\ JPEG\ encoding\ >$$

to implement translation between the video schemes H261 and JPEG. In the gateway, the H261 codec is also implemented in software as in IVS-H261; the JPEG codec is that in XVideo Parallax as in IVS-JPEG. The gateway establishes a participant table T as shown in Table 1 to control video translation. This table is filled with

No.	User id	Host id	Video scheme	Video sender	...
1	X	X	H261/JPEG	Yes/No	...
2	X	X	H261/JPEG	Yes/No	...
3	X	X	H261/JPEG	Yes/No	...
...	...	...	...	...	...
...	...	...	...	...	...

Table 1: Participant table in the gateway

participant information. In the table, an entry is corresponding to a participant. The field "Video scheme" indicates which video scheme the participant uses.

Suppose a video sender is in one video transport session S', the other session is S'', the directional translation pair to be used is P. The algorithm to translate and forward video data for the video sender is as follows:

Begin: the gateway receives all the packets of a video frame in S',

1. unpack the packets;
2. Translate data with the pair P;
3. Pack with RTP payload format;
4. Send the packets to S'';
5. End

4.3 Congestion control in IVS-H261/JPEG

Since in the videoconferencing application, video throughput is much greater than other kind throughputs, congestion control for videoconferencing applications is actually that for video stream.

Both IVS-H261 and IVS-JPEG use a feedback control mechanism [20] [21] to implement congestion control for video stream. We define three network states: unloaded, loaded and congested. A video sender finds out the network state by analyzing the feedback information in QoS packets sent by the participants and then adjusts its video throughput. The feedback information is loss rate of video packets. The feedback information from a participant reflects the network state on the link from the video sender to the participant. The system has to analyze feedback information from all participants to find out the synthetic network state over the multicast tree, then adjusts its video throughput. The algorithm to adjust video throughput is: if the multicast tree is unloaded, the throughput increases linearly; if the multicast tree is congested, the throughput decreases multiplicatively; otherwise the throughput don't change.

Congestion control in IVS-H261/JPEG is a congestion control issue for multiple video transport sessions connected by gateways. When a gateway translates and forwards video data for a participant in one session, it is actually a video receiver in this session and a video sender in another session. Therefore the gateway can be considered as a set of video receivers and senders. Some of them are in one session, and the others in another session. Thus congestion control for multiple video transport sessions can be implemented within each individual session.

IVS-H261/JPEG creates two video transport sessions. Congestion control is implemented within each session. In any session, each video sender adjusts its video throughput in this session as described above. The gateway IVS-SERV uses the same congestion control mechanism as in IVS-H261 and IVS-JPEG. If the gateway acts as a video sender in one session, it adjusts its video throughput in this session. If the gateway acts as a video receiver in one session, it sends QoS packets in this session to report its loss statistics of video packets.

4.4 Application consideration for IVS-H261 /JPEG

The application case of IVS-H261/JPEG is that some participants are in a broadband subnet (and these participants' host workstations are equipped with Parallax cards), but the other participants are in the narrow-band subnets or their subnets are interconnected to the broadband subnet with narrow-band links. IVS-JPEG can be used by the participants of the former type, and the participants of the latter type can only use IVS-H261. The gateway IVS-SERV should be performed at a host in the broadband subnet or close to it.

When in interactive conferences, considering the processing power of current workstations, we only allow one single participant to send video data at any one time. We call the speaker of any given time "the current speaker". When the current speaker is a participant using IVS-JPEG, for those using IVS-JPEG, quality video will be ideal, while for the others, the quality of video is not so high. On the other hand, when the current speaker is a participant using IVS-H261, all the participants will get video data of the same quality. Therefore different participant groups sometimes get different conference service, sometimes get the same conference service.

Obviously, different conference services result from different bandwidth resources and hardware resources.

Non-interactive conferences will be the typical application of IVS-H261/JPEG in practice. The speaker is supposed to be of the former type. In fact, if the speaker is of the latter type, this example will degenerate into an application example of mono-video-scheme videoconferencing system IVS-H261.

5 Conclusion

We have discussed the use of multiple video schemes in videoconferencing applications. The open system architecture for multi-video-scheme videoconferencing applications is proposed. A multi-video-scheme videoconferencing application consists of mono-video-scheme subsystems with some gateways. The gateways translate and forward video data, and forward audio data between the mono-video-scheme subsystems. Translation for different video schemes can be achieved in two ways. One is direct conversion; the other is "first to decode then to encode". Direct conversion is more effective but cannot always be achieved. The translation in the way "first to decode then to encode" is implemented in IVS-SERV, the gateway for IVS-H261 and IVS-JPEG. IVS-SERV enables two existing mono-video-scheme videoconferencing systems to become a multi-video-scheme videoconferencing application IVS-H261/JPEG. IVS-H261/JPEG can make the best use of the specific video hardware and broadband subnets of the Internet, and can also be suitable to narrow-band subnets of the Internet.

References

[1] Çaglan M. Aras, James F. Kurose, Douglas S. Reeves, H. Schulzrinne, *Real-Time Communication in Packet-Switched Networks*, Proceedings of the IEEE, Vol. 82, NO.1, January 1994,

[2] Mark J. Handley, Peter T. Kirstein, M. Angela Sasse, *Multimedia integrated conferencing for European researchers (MICE): piloting activities and the conference management and multiplexing centre*, Computer Networks and ISDN Systems, No. 26, 1993, pp. 275-290.

[3] Michael R. Macedonia, Donald P. Brutzman, *Mbone provides audio and video across the Internet*, IEEE COMPUTER magazine, April 1994, pp. 30-36.

[4] R. Aravind, G. L. Cash, D. L. Duttweller, H. M. Hang, B. G. Haskell, A. Puri, *Image and Video Coding Standards*, AT&T Technical Journal, January-February, 1993, pp. 66-89.

[5] Gregory K. Wallace, *The JPEG still picture compression standard*, Communications of the ACM, April 1991, Vol.34, No.4, pp.31-44.

[6] Ming Liou, *Overview of the* $p \times 64kbits/s$ *video coding standard*, Communications of the ACM, April 1991, Vol.34, No.4, pp.59-63.

[7] LeGall, D.J. , *MPEG: A video compression standard for multimedia applications*, Communications of the ACM, April 1991, Vol.34, No.4, pp.47-58.

[8] Baker, S., *Multicasting for sound and video*, Unix Review, Feb. 1994, pp. 23-29.

[9] K. Jeffay, D.L. Stone, F. D. Smith, *Transport and display mechanisms for multimedia conferencing across packet-switching networks*, Computer Networks and ISDN Systems, July 1994, Vol.26, pp. 1281-1304.

[10] Shiro Sakata, *Development and Evaluation of an In-House Multimedia Desktop Conference System*, IEEE JSAC, Vol. 8, NO. 3, April. 1990, pp. 340-347.

[11] Cosmos Nicolaou, *An Architecture for Real-Time Multimedia Communication Systems*, IEEE JSAC, Vol. 8, NO. 3, April. 1990, pp. 391-400.

[12] T. Turletti, J. C. Bolot, *Issues with multicast video distribution in heterogeneous packet networks*, Proc. 6th. International Workshop on Packet Video, Porland, Oregon, Sept. 1994, pp. F3.1-3.4.

[13] J. Pasquale et al., *Filter propagation in dissemination trees: Trading off bandwidth and processing in continuous media networks*, Proc. NOSSDAV'93, Lancaster, UK, Oct. 1993, pp. 259-268.

[14] Don Hoffman, Michael Speer, Gerard Fernando, *Network support for dynamically scaled multimedia data streams*, Proc. NOSSDAV'93, Lancaster, UK, Oct. 1993, pp. 240-251.

[15] D. Ferrari, *Client requirements for real-time communication service*, IEEE Communications, November 1990, pp. 65-72.

[16] Xu Mengxia, *Advances in Picture Coding*, Journal of China Institute of Communications, March 1993, Vol.14, No. 2, pp 40-47.

[17] Thierry Turletti, *H.261 software codec for videoconferencing over the Internet*, Technical Report 1834, INRIA Sophia Antipolis, January 1993.

[18] Le Coq Patrice, Le Moulec Hervé, Lhermitte Richard, Remondeau Christophe, *Analyse et modification d'un système multimédia*, Rapport de projet DIIC 3 LSI, IFSIC, Février 1994.

[19] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, *RTP: A Transport Protocol for Real-Time Applications*, Internet-Draft, March 21, 1995.

[20] J. Bolot, T. Turletti, I. Wakeman, *Scalable feedback control for multicast video distribution in the Internet*, Proc. ACM/SIGCOMM'94, Vol. 24, No 4, Oct. 1994, London, UK, pp. 58-67.

[21] J. Bolot, T. Turletti, , *A rate control mechanism for packet video in the Internet*, Proc. IEEE INFOCOM'94, June 1994, Toronto, pp. 1216-1223.