Translation Gateway for Videoconferencing Systems

Bernard COUSIN Junhui ZHOU

Laboratoire IRISA, université de Rennes-1
Campus de Beaulieu – 35042 RENNES CEDEX – FRANCE
bcousin@irisa.fr

Abstract

Videoconferencing systems may use differents digital video and audio formats due to variable environment constraints. So translation processes are necessary. We introduce a distributed architecture using specific translation agents : the disadvantages to translate in each user workstation and, the flexibility and optimization produced by the use of translation gateway are exhibited. But introduction of the translation gateways can disrupt the streams or isolate some participants. In this paper, the translation gateway for two different formats in one videoconferencing system is discussed. The mechanism of the gateway is first discribed. The gateway forwards RTCP packets between the separate RTP sessions, and forwards/translates the speaker's audio/video packets. The gateway also adjusts its video throughput according to the feedback information. Finally, the gateway IVS[1]-SERV for the videoconferencing systems using H261 or JPEG video formats is presented.

1 Introduction

Recent years have been seeing rapid development of videoconferencing technology. Videoconferencing systems acquire, transport and present real time video and audio data between networked computers. They enable with today workstations and data networks to organize a videoconference (e.g. IVS [1, 2], CU-SeeMe, NV and VIC [6], etc).

Videoconferencing systems concerns two principal aspects: video and audio data compression, and real time data transport.

Video data can only be transported after being compressed due to the huge data rate (Uncompressed video data rate in CIF format is 36.25 Mbit/s). There are numerous video compression technics. Amongst them we can find intraframe compression methods used by compression standards such as JPEG [8], or interframe compression methods used by compression standards such as H261 [9] and MPEG [10]. Interframe compression achieves greater compress ratio than intraframe compression because interframe compression takes advantage of similarity of consecutive frames while intraframe compression only treats individual frames. But obviously, interframe compression technics need much more processing power than intraframe compression technics. Trade-off between video quality, data rate, delay and processing power is the main goal of our work in this paper.

For audio, there are also several compression encoding technics: sample based encoding and frame based encoding (PCM, ADPCM, GSM, LPC). Typical coding rates are: PCM – 64kbit/s, ADPCM – 32kbit/s, GSM – 13kbit/s, LPC – 4.8kbit/s. Albeit video and audio compression processeses have similar behaviors, their specific constraints lead to different architecture optimisations. We will exhibit these differences in the following sections.

Videoconference systems usually use only one compression format. But using different formats during one videoconference session is also possible and sometimes required. Each format has their own characteristics. Some formats can achieved low video throughput due to efficient encoding and compression technics. Some station architectures or network infrastructures can process or transmit high frame rate. Some formats can tolerate higher frame loss than other. For example, our teleconference application Ivs proposes two video formats. Ivs-JPEG can achieve high video frame rate (20-30 frame/s) because it is supported by XVideo Parallax hardware card, but Ivs-H261 can sustain comparatively low video throughput (3-10 frame/s) due to its software implementation on usual workstations. However at the same video frame rate Ivs-JPEG generates greater video throughput (1000-1500 kbit/s) than Ivs-H261 (30 kbit/s).

IVS-JPEG and IVS-H261 are suitable to different environments. The first implementation can be used on well equiped workstations connected to high speed

[1] IVS : INRIA Videoconferencing System.

links which enable high frame rate and high quality video to be obtained. While the latter can be used on simple workstations with low speed links ensuring low throughput. To ensure a total interconnection of all reachable workstations, videoconference applications have to deal at the same time with network links with different speeds, with stations with variable architectures, and with video formats with different characteristics. Different video formats can be used at the same time in one videoconference session providing that translation can be achieved. In the following sections we will propose architectures wich optimize this required translation process.

Three solutions come in mind, to deal with the multi format problem. The first obvious solution is to impose only one (video and audio) format. This solution has the following drawbacks : the first drawback has already been introduced, generally a format is a tradeoff so it is not optimized for all environnements. The second drawback is, as technics evolve best formats can be found in the future, but mono format solution will not allow new formats to be used by an already developped application. So a distributed application which tolerates, at raisonnable cost, some adaptivity towards the environnements and the future and potential evolution will be a must.

The second solution can be already found in several videoconferencing applications in their audio process. These applications enable one from several possible audio formats to be user selected and encoded. Then the chosen selected format is transmit and automaticaly decoded at each receiver. This solution does not correspond to what our application needs. Because, when one videoconference session runs, only one format is active. These sort of applications are multiformat applications but monoformat session applications. Furthermore these sort of applications have the following drawbacks. First, their code can become too large due to the numerous formats they have to encode and decode to assure total interoperability. Second if a new format appears the application has to be upgrated. To upgrate numerous copies of an application used on many stations distributed over a large network is not easy, can take a long time, and can stop the teleconference service for a while. Third, all the participants to one teleconference session have to agree with the chosen format. Fourth, as told previously, some stations have not the processing power, or the specific video or video card, or the high speed or high quality connection to the network, required by the format chosen for the current session of the application.

In this paper we describe a third solution, which has not the previous drawbacks. This solution proposes multiformat session application and uses translation servers.

Subsequently without lost of generality, we suppose that the teleconference system only deals with two differents video formats. In the same way, we make the assumption that only one speaker is active at any time. We wil reconsider these assumptions in the conclusion.

In next section, we will present brievly the Internet protocols. Then Section 3 we discuss the translation gateway architecture. In Section 4, we present our specific gateway for the Ivs application between H261 and JPEG formats. Section 5 concludes the paper.

2 Internet protocols

Real time data transport is the basic requirement of the videoconferencing system. On Internet, to implement real time data transport, the Internet Audio-Video Transport Working Group has proposed the Real-Time Transport Protocol (RTP) [3]. RTP consists 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 information about the participants within a RTP session and monitors the quality of service (QoS).

For application to multi-participant videoconference, RTP is based on UPD that supports IP multicast. So an RTP session S (*i.e. a videoconference session*) can be defined by an IP group address $@m$ plus a set of three UDP ports (p^v, p^a, p^c) as following:

$$S : \{@m,\ (p^v,\ p^a,\ p^c)\}\ .$$

The IP group adress identify the set of user workstations which participe to the videoconference session. The three different ports enable the video data stream, the audio data stream and RTCP stream to be multiplexed (and demultiplexed). Each stream is uniquely identified by the IP station address of its sender, the IP group address of the videoconference session and the port number associated to the data type of the stream.

In an RTP session created by a typical videoconferencing system, each participant has his own unique synchronization source (SSRC) identifier, and all his packets carry this SSRC identifier. During the session, each participant sends periodically RTCP Source Description (SDES) packets to declare his join in the videoconference. The SDES packets contain information about the participant including his user name, host name, e-mail address and the like. All the other participants receive the SDES packets, and

keep track of his presence by use of a participant table. When a participant is going to leave the videoconference, he sends an RTCP BYE packet. All the other participants will remove his entry in their participant tables. In the same way, if a participant doesn't receive periodically the SDES packet of another participant the entry of the latter in the participant table of the first is removed.

Amongst the participants to a teleconference session, some of them sends video or audio data packets : we call them speakers. Speakers send RTCP Sender Report (SR) packets to issue their video packet transmission statistics to the others participants. All participants send RTCP Receiver Report (RR) packets to the speakers to issue their reception statistics. Report packets contain many statistic information fields. Amongst them the packet lost rate is used to enable speakers (using specific mechanism [19]) to adjust their video throughput. In the last section, we will describe how our proposition influences the adaptative control mechanisms.

3 Translation gateway

In the first section, we have introduce the translation requirements to achieve the interoperability of open multimedia and distributed applications. We propose to put the translation process in specific stations called translation gateways. The translation process can take advantage, first of the processing power of the servers without overloading of the user workstations, second of specific equipements attached to the servers but too rare or costly to be present in each user workstations. Third if new format is developped then the application code of the user workstations may remain unchanged, but only some developpements are required to upgrade the code of the translation servers. Fourth if the underlying network enables broadcasting (which is probable because many local networks propose this function, and the considered applications required multicasting) the server can take advantage of it to optimize the translation process. Actually the presence of the server and the broadcast networking function enable the translated data to be broadcast to all stations on the server domain. So only one translation process and only one translated data stream are required to feed several destination workstations with a different data format.

Within a multiformat videoconference session, two given different systems A and B may use either the same IP group address but different UDP port set or different IP group address. Since A and B are definitly run in separate hosts, to use the same IP

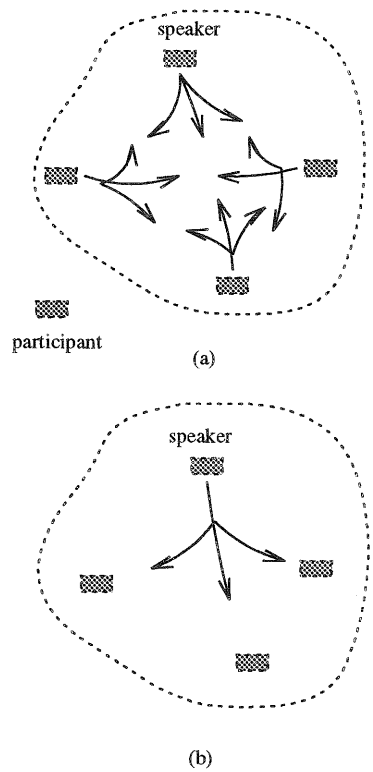

Figure 1: (a) RTCP packet streams in S_A; (b) Data packet streams in S_A

group address will reduce the bandwidth efficiency and increase the overhead of IP layer. So the latter is optimal. We denote the two separate sessions S_A, S_B created respectively by A, B as

$$S_A : \{@m_A, (p_A^v, p_A^a, p_A^c)\} ;$$

$$S_B : \{@m_B, (p_B^v, p_B^a, p_B^c)\} ;$$

$$m_A \neq m_B .$$

Without loss of generality, we always assume that the speaker is in S_A. Any participant sends RTCP packets including SDES and SR or RR packets. There will be a bidirectional stream of RTCP packets between every two participants in each session. The speaker sends video and audio data packets. Therefore in S_A, there is a one-directional stream of RTP data packets from the speaker to every other participant. Fig.1 shows the packet streams in the session S_A. Within the other session S_B, we can only find RTCP packet streams. No one sends data packets there.

Within each session, the participants know each other through transmission of SDES packets. And within S_A, the participants can see and hear the speaker. But the participants in S_A don't know those in S_B, vice versa, and the participants in S_B are not able to see and hear the speaker.

We setup a mapping in S_B for each participant in S_A including the speaker. Similarly, we setup a mapping in S_A for each participant in S_B. We state that the information about a participant and that about his mapping are the same, and that the speaker's mapping has the same original video and audio data as the speaker's. Thus, the participants in S_B can know those in S_A, and see and hear the speaker; the participants in S_A can also know those in S_B.

The gateway is to establish the mappings for all the participants. It will be in both S_A and S_B. For any given participant p in S_A, the gateway forwards his SDES packets from his corresponding session S_A to the other S_B. Then it seems to the participants in S_B that there appears a new participant in their session. This "new participant" is actually the participant's mapping. Since the gateway forwards his SDES packets without changing them, these packets will carry the original information to the participants in S_B. Then the participants there know p by his mapping. If p is the speaker, the gateway will also forward his audio packets to the session S_B without changing, and translate video packets from the video scheme V_A (corresponding to the system A) to the scheme V_B (corresponding to the system B) then sends to the session S_B. Thus the speaker's mapping is formed in S_B. It sends data packets as a speaker whose original audio and video data are the same as the real speaker's. The participants in S_B see and hear the speaker by his mapping. For each participant in S_B, the gateway will establish his mapping in S_A by forwarding his SDES packets from S_B to S_A. Now, all the participants in both sessions know each other, and see and hear the speaker.

Within the session S_A, the speaker sends SR packets, and all the other participants send RR packets. As a receiver in S_A, the gateway must send RTCP RR packets in S_A to report its reception statistics. Within the session S_B, the gateway is a sender, so it should send SR packets. The other participants in S_B send RR packets, and the gateway will adjust its video throughput according to the reception ststistics received. The SR packets and RR packets reflect the internal transmission and reception states in a session. It is not required to forward them from one session to the other.

If a participant want to leave the videoconference, he will send a BYE packet. The gateway must forward this BYE packet in order that all the participants in the other session would know his leave and remove his entry from the participant table.

We show in Fig.2 the packet streams in a videoconference. Besides the packet streams within each session, there is a bidirectional RTCP packet stream between the gateway and each participant, and there are a one-directional data packet stream from the speaker to the gateway and a one-directional data packet stream from the gateway to each participant in S_B.

Translation of video packets from the scheme V_A to V_B can be implemented in two ways. One way is to transform directly video data from V_A to V_B. The other way is first decoding video data to get original data then encoding in V_B. The first way cannot always be implemented for certain schemes. But the second way can always be implemented. In next section we will discuss the second way in detail. The first way is beyond the scope of this paper.

For the gateway, being transparent at RTP layer is an important requirement. (Since RTP is integrated into the application, RTP layer is in effect the application layer.) The packets forwarded by the gateway will carry the IP source address of the gateway. However, the packets are intact at RTP layer, and the video packets translated and retranmitted by the gateway will carry the speaker's SSRC identifier, no participant in any session will find the existence of the gateway. Therefore the gateway is transparent at RTP layer.

4 Gateway for IVS-H261 and IVS-JPEG

IVS-H261 and IVS-JPEG are two videoconferencing systems developed by INRIA in France. The two systems use RTP to transport video and audio data, and RTP is integrated into the systems. Both IVS-H261 and IVS-JPEG are based on UDP. Compared with the typical RTP session (See Section 2), IVS-H261 and IVS-JPEG use 4 UDP ports for video packets, audio packets, SDES/BYE packets and quality of service (QoS) packets. QoS packets are actually reception report (RR) packets, but they are sent in unicast from the participants to the speaker. Sending QoS packets in unicast saves the bandwidth, however except for the speaker, no other participant can receive QoS packets. The speaker sends no sender report (SR) packet. So in the session created by either IVS-H261 or IVS-JPEG, each participant sends periodically SDES packets, and sends BYE packet when he leaves the videoconference. The speaker sends video and audio packets. All the other participants send periodically QoS packets in unicast to the speaker, and the speaker adjust his video throughput according to the feedback information carried in QoS packets. Such an RTP session can be considered to be a simplification of the typical RTP session.

The difference between IVS-H261 and IVS-JPEG is that IVS-H261 uses the video compression scheme

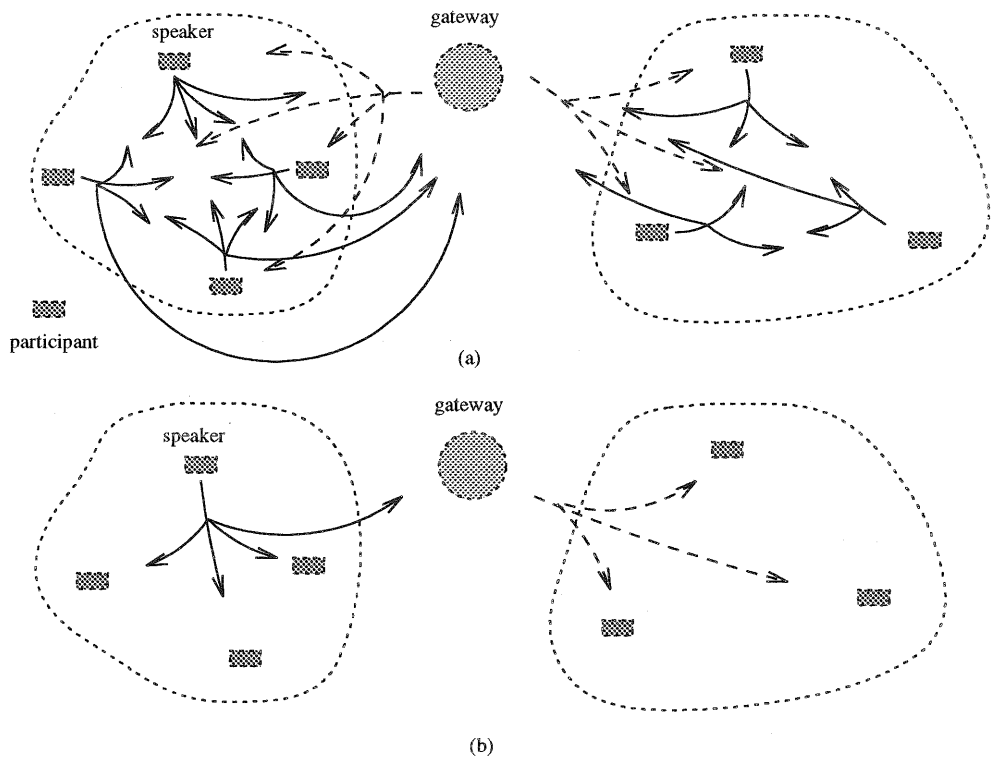

Figure 2: (a) RTCP packet streams in a videoconference; (b) Data packet streams in a videoconference

H261 to compress video data, while IVS-JPEG uses JPEG to do so. Therefore these two sytems haven't video interoperativity. In IVS-H261, the video codec is software codec; and in IVS-JPEG, the video codec is hardware codec in XVideo Parallax card. Since H261 is an interframe compression scheme, it gains a great compression ratio. The video throughput of IVS-H261 is about 30kbit/s (in CIF). But the video frame rate is relatively low (5 frame/s) because the video cedec is implemented in software. JPEG is an intraframe video compression scheme. Its compression ratio is not so high as that of H261. When IVS-JPEG operates at the same video frame rate as IVS-H261, its video throughput is about 200kbit/s. However the codec in IVS-JPEG is implemented in hardware, it can arrive at 30 frame/s, and meanwhile the video throughput is 1500kbit/s.

Both IVS-H261 and IVS-JPEG operate in three audio encoding schemes: Pulse Code Modulation (PCM), Adaptative PCM (ADPCM) and variable rate ADPCM. They can have audio interoperativity by using the same audio scheme.

5 Conclusion

We have discussed the gateway for two different videoconferencing systems on the Internet. The gateway makes all the participants using different audio and video formats be able to communicate, and see and hear the speakers. It also implements the adaptive adjustment of its video throughput. The gateway is robust to the cases of different speakers at different times and multi-speaker at a time since the gateway is informed of the speaker through the user interface. In the case of multi-speaker at a time, we will run the gateway at different hosts to work for the speakers one by one. This may generate unacceptable resource requirements if the number of the speakers is great. Fortunately, there is only one speaker in non-interactive videoconferences such as lectures, and at most two or three speakers at a time in interactive videoconferences. So in practice, the number of the speakers is always small.

Translation of video data in the gateway may require certain time, while retransmission of audio packets requires little time. The latency of video packets may increase much more than that of audio packets. This might influence the timing relation of video and audio stream across the gateway. So at the receivers, synchronization between video and audio

steams becomes very important. Moreover, comparative great latency of data transmission may make the participants feel uncomfortable in interactive videoconferences. Therefore, it is necessary to minimize the latency of video packets across the gateway. To reduce the latency through more effective algorithms for video translation is our further work.

References

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

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

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

[4] 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,

[5] 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.

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

[7] 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.

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

[9] 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.

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

[11] W. Fenner, L. Berc, R. Frederick, S. McCanne, *RTP Encapsulation of JPEG-compressed video*, Internet-Draft, March 23, 1995.

[12] T. Turletti, C. Huitema, *RTP payload format for H261 video streams*, Internet-Draft, July 10, 1995.

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

[14] 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.

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

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

[17] 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.

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

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