

The Development of a Multimedia Transcoding System for Mobile Access to Video Conferencing

Hiroyuki KASAI[†], *Regular Member*, Mike NILSSON^{††}, Tim JEBB^{††},
Mike WHYBRAY^{††}, *Nonmembers*, and Hideyoshi TOMINAGA[†], *Regular Member*

SUMMARY Today, many audiovisual delivery systems, including video streaming and video conferencing, are being developed for use over a range of networking technologies, the differing characteristics of which pose problems for service level interoperability. Multimedia transcoding is one means to provide interoperability between different types of audiovisual terminals and between terminals that connect to different networks. In this paper, we will present a multimedia transcoder system, which provides interoperability between video conferencing terminals on IP networks and mobile terminals on mobile networks.

key words: multimedia transcoding system, mobile access, video conferencing, H.323

1. Introduction

Today, many audiovisual delivery systems, including video streaming and video conferencing, are being developed for use over a range of networking technologies, such as PSTN, ISDN, ATM, mobile and IP networks. In particular, recent wide-band mobile communication systems such as General Packet Radio Service (GPRS), Universal Mobile Telecommunications System (UMTS) and International Mobile Telecommunications 2000 (IMT-2000), and the progress of mobile terminals, such as visual mobile phones and Personal Digital Assistants (PDA), enable us to access audiovisual services over mobile networks. However, the differing characteristics of existing systems pose problems for service level interoperability. In order to provide interoperability among different kinds of systems, system protocol switching and transformations of media data are required. Moreover, when those network characteristics are so different, the bit rate of the multimedia stream must be significantly reduced.

Multimedia transcoding is one means to overcome these differences and provide interoperability: for example, transcoding could improve the error resilience of a multimedia stream for transmission over a more error prone network; or transcoding could dynamically adjust the bit rate of the multimedia stream for transmission over a network that can not guarantee availability

of bit rate. There has been a great deal of theoretical research into transcoding, including [1]–[7], but much less effort has been applied to practical implementation.

In this paper, we will present a multimedia transcoder system, which provides interoperability between video conferencing terminals on IP networks and mobile terminals on mobile networks. In particular, we deal with H.323 as a video conferencing terminal. The rest of this paper is organized as follows. We first present a target system outline in Sect. 2. Then, Sect. 3 presents the developed multimedia transcoder in detail. A call message sequence is described in Sect. 4. The system performance is described in Sect. 5. Finally, Sect. 6 gives conclusions of our work.

2. System Overview

This section provides an overview of the system. The objective of this system is to allow an H.323 terminal to connect to a mobile client via a multimedia transcoder. The system outline is shown in Fig. 1. In this figure, some H.323 [8] terminals are connected to the Internet which is more reliable and has broader bandwidth than the mobile network to which some mobile terminals are connected. There are some significant problems to overcome when a mobile terminal connects to an H.323 terminal. In particular, the system interface and media data of the two terminals are different. Moreover, the bit rates of each media can be different. Therefore, some special mechanisms are required to provide the interoperability between these two terminals.

We have developed a transcoder that overcomes these problems. In the following sections, ITU-T H.323, the mobile client and the multimedia transcoder are explained briefly.

2.1 ITU-T H.323 Standard

The H.323 standard provides a foundation for audio, video, and data communications across IP networks, including the Internet. H.323 is an umbrella recommendation from the International Telecommunications Union (ITU) that sets standards for multimedia communications over Local Area Networks (LANs). This recommendation is still being extended. Many commercial PC applications supporting H.323 are available,

Manuscript received February 8, 2002.

Manuscript revised May 10, 2002.

[†]The authors are with Global Information and Telecommunication Institute, Waseda University, Tokyo, 169-0051 Japan.

^{††}The authors are with BText Technologies, Ipswich, IP5 3RE, U.K.

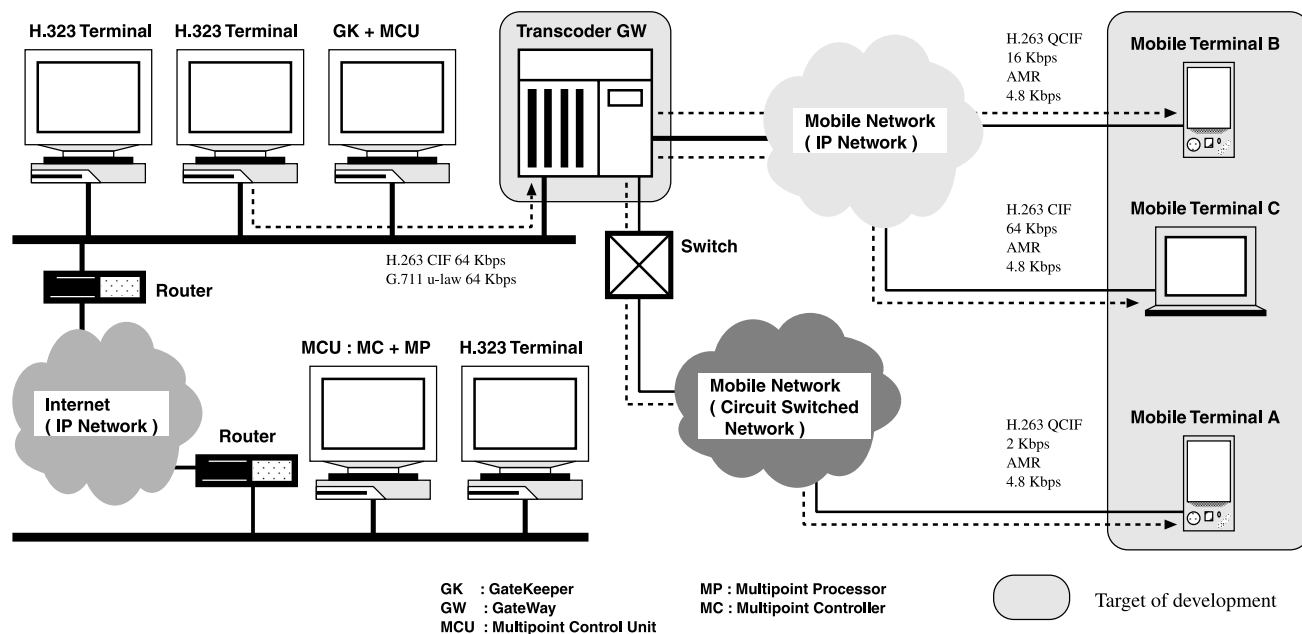


Fig. 1 System outline.

including Microsoft NetMeeting [9], Intel VideoPhone [10] and SmithMicro InternetCommSuite [11].

The functions of H.323 call processing are divided into three basic areas: registration with a gatekeeper using the Registration, authentication, and status (RAS) protocol, call signalling using H.225.0 [12], and media capability exchange with routing of the media channel using H.245 [13]. H.225.0 messages for RAS are transmitted over UDP, and those for call signalling are transmitted over TCP. H.245 runs only over TCP. The basic H.323 call model consists of five phases, which are (i) call setup, (ii) initial communication between H.323 clients and terminal capability exchange, (iii) establishment of audiovisual communication between H.323 clients, (iv) request and negotiation of call services, and (v) call termination. Audiovisual data are transmitted over UDP, and packetised using RTP [14].

2.2 Mobile Client

Prior to the development of the transcoder, BTexact Technologies was involved in Eurescom project P905 [15], known as AQUAVIT - Assessment of QUality for Audio-Visual signals over the Internet and UMTS, one deliverable from which was a real-time audiovisual communication application for the PC. This application supported H.261 [16]/H.263 [17] video codecs and G.711 [18], G.723.1 [19], MPEG-2 AAC (Advanced Audio Coding) [20] and MPEG-4 CELP (Code Excited Linear Prediction) [21] audio codecs over IP networks. Subsequently, this application was extended to support Adaptive Multi-Rate (AMR) speech coding [22], and H.223 [23] multiplexing for transmission over circuit

switched data (modem) connections. The application uses a simple proprietary control protocol derived from H.245. Examples of the control messages are described in Table 1. To simplify implementation, this protocol does not use ASN.1 encoding which is used in H.245. This terminal application has the functionality of both 3G-324M [24] and H.323 except the control protocol. In the following, we refer to this client as the 'mobile client.' Figure 2 shows the diagram of the mobile client.

2.3 Multimedia Transcoder

This transcoder allows a mobile client to connect to an H.323 client as shown in Fig. 1. 'Mobile Client A' connects at 9.6 kbit/s with a GSM modem. The other side of the transcoder connects to an H.323 client using an IP network, over which it receives RTP packetised H.263 CIF video at 64 kbit/s and G.711 at 64 kbit/s. This data is transcoded to H.263 QCIF video at about 2 kbit/s and AMR speech at 4.8 kbit/s, which are multiplexed using H.223 and transmitted using a GSM data connection to a mobile client.

Moreover, the transcoder can operate with multiple H.323/mobile clients with different codec capabilities and/or different network capabilities, accessing the same source. Figure 1 shows the scenario of the transcoder allowing multiple clients to receive different qualities of H.263 video but the same quality of AMR audio from a single H.323 client. 'Mobile Client B' receives H.263 QCIF video at 16 kbit/s and 'Mobile Client C' receives H.263 CIF video at 64 kbit/s simultaneously with 'Mobile Client A.'

In the rest of this paper, we will refer to the flow

Table 1 System control messages used in mobile clients.

Message	Explain
CONNECT / DISCONNECT	CONNECT is used to connect to the Transcoder from the Client or to the Server from the Transcoder, and to inform it of the local IP address in the IP connection case. DISCONNECT is used to disconnect from the remote application.
OPEN_MEDIA_CHANNEL /CLOSE_MEDIA_CHANNEL	OPEN_MEDIA_CHANNEL is used to establish a media channel, specifying the media type (audio or video) and codec etc. CLOSE_MEDIA_CHANNEL is used to close a media channel.
STOP_MEDIA_TRANSMISSION	STOP_MEDIA_TRANSMISSION is used to inform the transcoder that decoding has stopped, and that it should stop transcoding and transmitting. And, it is used to inform the Server that transcoding has stopped, and that it should stop transmitting.
REQUEST_MODE	REQUEST_MODE is used to request the transmission of audio and/or video to the transcoder. Codec type, bit rate and frame rate are specified.
H323_ADDITIONAL_INFO	H323_ADDITIONAL_INFO includes the IP address of the H.323 client that the transcoder client wants to connect to. This also contains the media direction(e.g. Video in Down-link, Audio in Down and Up-link.).
ROUND_TRIP_TIME	ROUND_TRIP_TIME is used to measure the round-trip-time between the transcoder and the client.

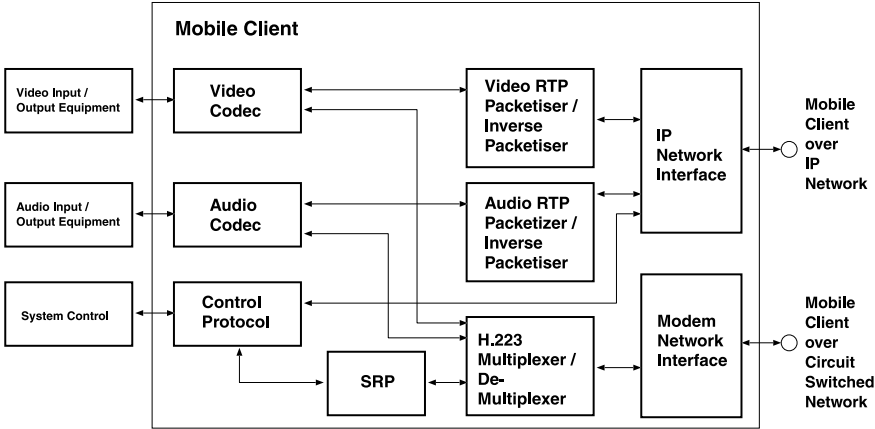


Fig. 2 Mobile client diagram.

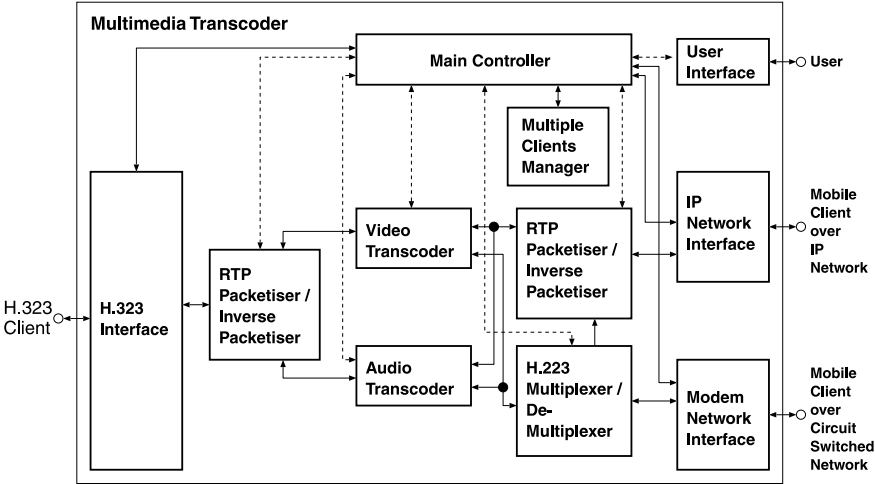


Fig. 3 Multimedia transcoder diagram.

from an H.323 client to a mobile client as the ‘Down-link,’ and the reverse direction the ‘Up-link.’

3. Multimedia Transcoder Description

In this section, the transcoder system is described.

The transcoder is shown in Fig. 3. The transcoder has system interfaces and media processes as well as media transcoders which adopted some basic technologies used in [1]–[7]. It is composed of the Main Controller, Multiple Client Manager, System Interfaces, Video Transcoder, Audio Transcoder, H.223 Multiplexer and

RTP Packetisers. These processes are described in the next section.

3.1 Main Controller and Multiple Clients Manager

The 'Main Controller' is located in the highest layer of the protocol stack. This controller controls (i) parsing and conversion of all call messages to/from the network interfaces, (ii) the operation of all processes, (ii) creation/deletion and start/stop of instances of processes, and (iii) handling of all events from processors and the user interface.

The 'Multiple Clients Manager (MCM)' manages all the information of all clients which are connected to the transcoder. This information consists of the requests of a client (e.g. media direction, media type, media codec, bit rate, frame rate, server address etc.) and all the instances, that a mobile client is using in a session. Moreover, this manager performs registration and cancellation of an instance into/from the multiple buffer lists.

3.2 System Interfaces

On the input side of the transcoder, an H.323 interface is provided to allow communication with terminals complying to H.323. The H.323 interface provides call signalling functions, capability exchange negotiation, and receiving and transmitting of RTP packetised audiovisual data.

On the output side, the modem network interface is supported for communication over circuit switched networks, such as PSTN and GSM. This interface supports receiving and transmitting system protocol messages and media data, multiplexed using H.223. Moreover, the IP network interface is supported for IP networks. This interface supports receiving and transmitting system protocol messages over TCP, and media data which are packetised using RTP, and transmitted over UDP.

3.3 H.223 Multiplexer and RTP Packetiser

When using the IP network interface, system control messages, are sent using TCP, and media data are packetised and transmitted over UDP. Audio frames are packetised with one audio frame per UDP packet with the packet header containing the frame number and timestamp. Video data are packetised using RTP, as defined in RFC 1889, and the H.263 Payload Format defined in RFC 2190 [25].

On the other hand, when using the modem interface, media data and control messages are multiplexed using H.223 (baseline, Annex A [26] or Annex B [27]), which interleaves video, audio and system control streams into a single bit stream. These control messages are transferred using the Simple Re-transmission

Protocol (SRP) defined in H.324 [28], which provides frame error checking. If bi-directional video communication is requested from a mobile client, the raw video bit stream is packetised using RTP in the transcoder in order to forward it to an H.323 client.

3.4 Video Transcoder

The video transcoder performs bit rate reduction of the input bit stream for video transmission over lower bandwidth networks. It can also provide multiple bit rates and frame rates of video for supporting multiple clients. Figure 4 shows the video transcoder diagram.

The transcoder, in the Down-link, can receive H.261 and H.263 video coded at CIF or QCIF resolution, and can produce H.263 at either CIF or QCIF resolution. Many video transcoder architectures have been proposed to achieve fast processing, but at the expense of quality degradation from drift error. As this was considered unacceptable, a cascaded pixel-domain transcoder architecture was implemented [29]. The transcoder is composed of a video decoder and a video encoder. The video transcoder consists of two sub-systems, which are the 'Decoder sub-system' and the 'Encoder sub-system.'

The 'Decoder sub-system' is composed of two processors: 'Video Decoder (VD)' and 'Multiple Decoded-Info Copy (MDC),' and two kinds of buffers: 'Packet Buffer (PB)' and 'DecodedInfo Buffer (DB).' VD decodes the input bit stream from the H.323 Interface into pictures in the pixel-domain, and reduces the resolution of these pictures by filtering and sub-sampling, typically converting from CIF to QCIF. Moreover the decoder stores picture encoding information, including picture types, motion vectors, macro-block modes, and quantisation parameters in DB. MDC provides multiple copy functions which copy the content stored in this DB into other DBs, to allow the support of multiple receivers, each potentially receiving different frame rates of video at different bit rates.

The 'Encoder sub-system' is composed of one process: 'Video Encoder (VE)' and two kinds of buffers: 'DecodedInfo Buffer (DB)' and 'Symbol Buffer (SB).' VE encodes the pictures stored in DB based on the picture coding informations from DB. The bit stream from VE is stored in SB.

The transcoder reduces the input bit rate by a combination of picture re-sizing, frame dropping and re-quantisation. It has the following features. The transcoder calculates the mode of output macro-blocks using the decoded macro-block mode information. The transcoder uses the decoded motion vectors to avoid the computational complexity of motion estimation in the video encoder. However, as copying vectors can cause picture quality degradation [30], a limited motion search is performed around the decoded motion vectors. The transcoder drops pictures when the symbol buffer

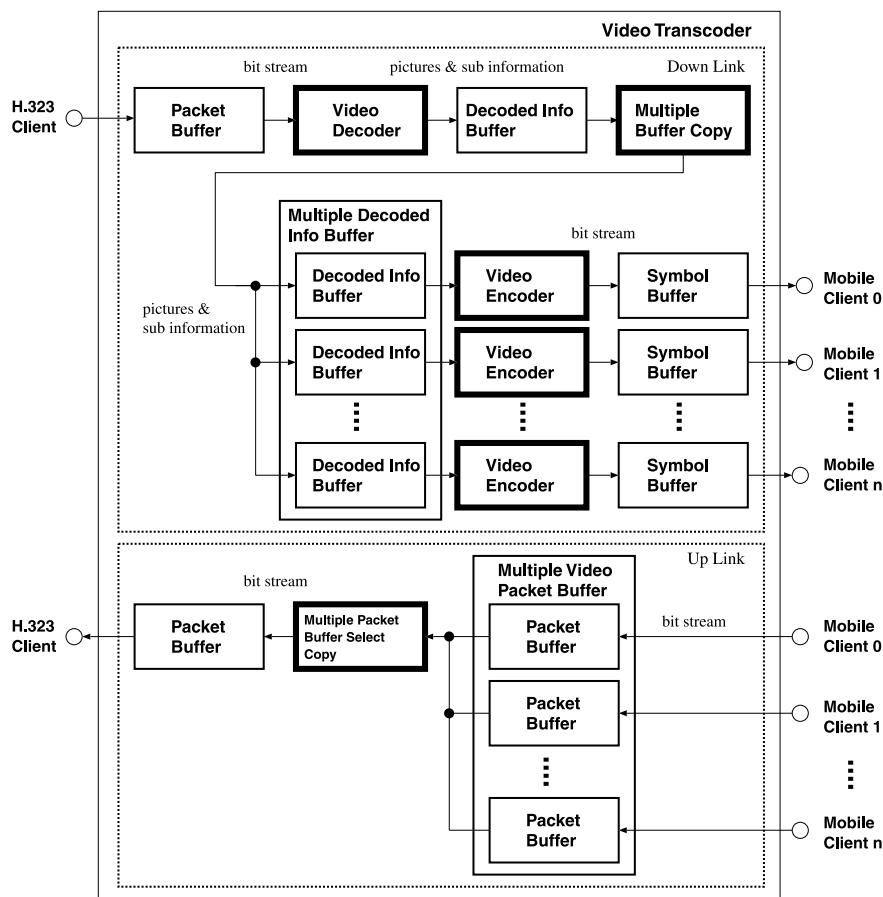


Fig. 4 Video transcoder diagram.

is quite full. This causes the decoded motion vectors to no longer represent movement from the local reference picture to the picture currently being transcoded. So, motion vectors from the dropped pictures are stored, allowing the calculation of new motion vectors using the motion vectors of the dropped pictures as well as the motion vectors of the picture being transcoded, as described in [31]. However, when the number of dropped pictures exceeds a pre-defined threshold, the decoded motion vectors are ignored, and conventional motion estimation is performed.

No video transcoding is performed in the Up-link as the available bandwidth is higher than that required. The video transcoder, in the Up-link, is composed of 'Multiple Packet Buffer Copy' and two buffers. This process selects the bit stream which should be transmitted to an H.323 terminal based on the external information. In the current implementation status, this information is set by a transcoder manager. The further strategy, e.g. audio detection, should be investigated in the future.

3.5 Audio Transcoder

The audio transcoder performs conversion between

G.711, GSM Full-Rate Codec [32], G.723.1 and AMR. It supports multiple clients, each of which can receive audio encoded with different standards and at different bit rates. The audio transcoder is composed of an audio decoder that decodes compressed audio into PCM audio samples and an independent audio encoder that encodes these samples into an output bit stream. Figure 5 shows the audio transcoder.

The audio transcoder consists of two sub-systems, which are 'Decoder sub-system' and 'Encoder sub-system.'

'Decoder sub-system' is composed of two processes: 'Audio Decoder (AD)' and 'Multiple Audio Frame Buffer Copy (MAFBC),' and three kinds of buffer: PB, 'Decode Audio Buffer (DAB)' and 'Multiple Audio Frame Buffer (MAFB).' AD decodes the input audio bit stream into a PCM signal and stores this in DAB. MAFBC copies PCM signals into DAB belonging to the particular MAFB, thus providing support for multiple clients receiving different audio bit streams.

'Encoder sub-system' is composed of two processes: 'Audio Encoder (AE)' and 'Multiple Audio Packet Buffer Copy (MAPBC),' and two kinds of buffer: PB and 'Multiple Audio Packet Buffer (MAPB).' AE encodes PCM signals into an output bit stream, and

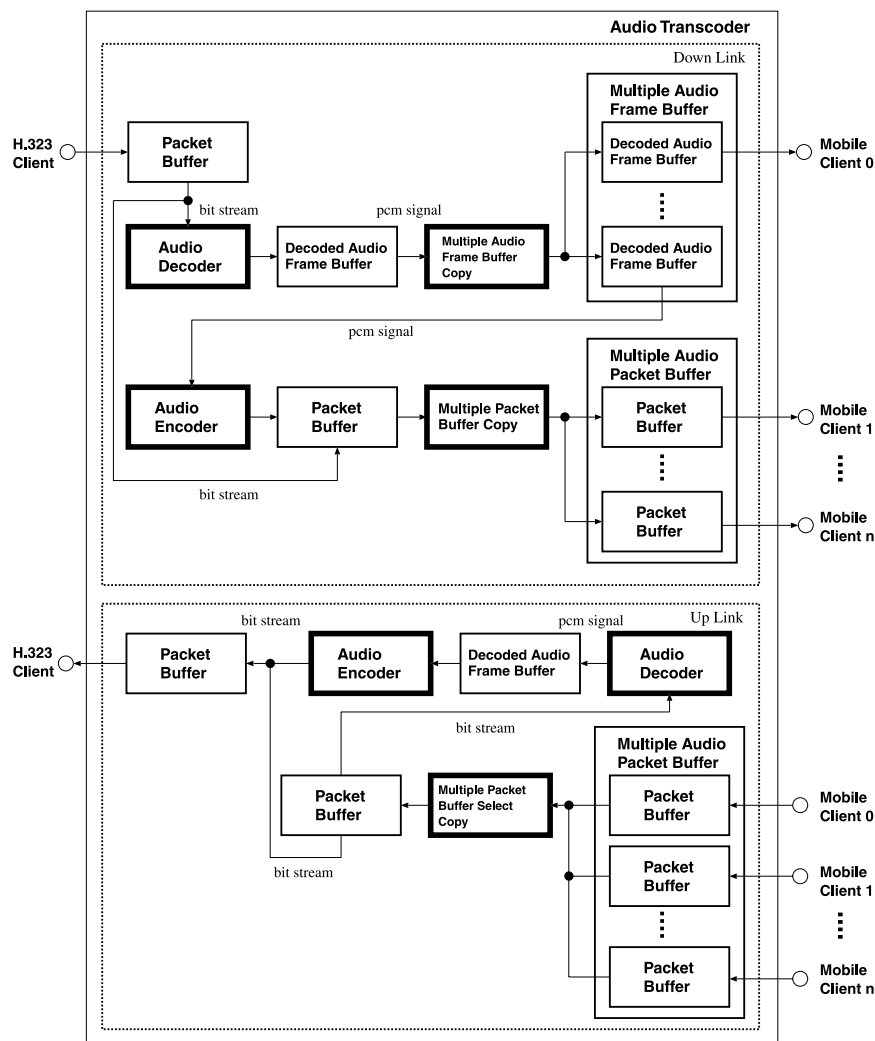


Fig. 5 Audio transcoder diagram.

stores this in PB. MAPBC copies the bit stream from this PB into the PB, within the MAPB, used for the given client.

When the requested audio coding algorithm is the same as one that is being used between an H.323 terminal and the transcoder, no audio transcoding is performed. In this case, the input bit stream, which is input into PB in front of AD, is directly copied into PB at the back of AE.

3.6 Low Delay Audio Communication

This transcoder can provide a low delay path for audio communication using an ISDN interface, and enhancement of the session by using a data network for the transmission of video, for which some delay may be tolerated. This interface is compliant with COMMON-ISDN-API (CAPI) 2.0 [33], [34]. In this case, G.711 audio codec is used between the transcoder and the audio terminal. (e.g. GSM phone or PSTN phone.) In addition, the audio codec used between the transcoder

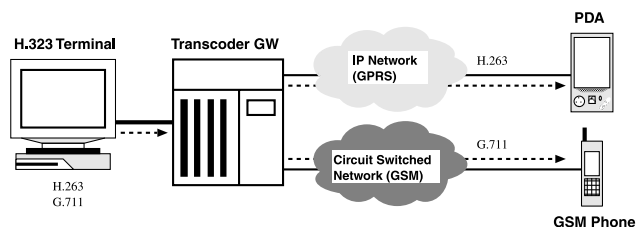


Fig. 6 Low delay audio communication.

and the H.323 client is also selected to G.711 by restricting the audio codec capability of the transcoder indicated in the H.323 capability exchange. Therefore, no audio transcoding is needed. One example of the transcoder architecture for mobile video conferencing is that speech is transmitted over GSM, as in normal voice communication, and GPRS is used to enhance the service by delivering video to a PDA as shown in Fig. 6.

4. Call Message Sequence

Figure 7 shows signal flows for a call between a mobile client and an H.323 client via the transcoder. We will describe two call phases separately. Firstly, we will describe the phase between call setup and establishment of audiovisual communication (Phase A), and secondly, the phase between stopping the audiovisual communication and call termination (Phase B). Phase A is as follows for uni-directional audiovisual communication in the Down-link.

1. A mobile client initiates communication by making a call to the transcoder and sends CONNECT message. Then, the client also sends H323_ADDITIONAL_INFO message for the call to the H.323 client. The transcoder registers the client with MCM.
2. The transcoder makes a connection to send H.225.0 SETUP message to the H.323 client using the H.323 destination IP address received from the mobile client. After H.225.0 ALERTING and CONNECT messages are received, an initial H.245 communication takes place between the H.323 client and the transcoder. This performs Master Slave Determination and Capability Exchange, then Logical Channel Signalling, allowing audiovisual communication to begin.
3. The transcoder also receives REQUEST_MODE message from the mobile client. The requested mode is registered with MCM.
4. Once the transcoder has received the confirmation

of the establishment of the media connection with the H.323 client, it will try to create and register the instances of each process. In this stage, the main controller queries MCM whether the instance(e.g. video decoder instance) to be created had been already created by other client requests. If previous other clients had accessed the same media stream in the same H.323 client, no new media decoder instance is created. In this case, that instance is registered with MCM as the instance the new mobile client is sharing with existing clients.

5. Once the transcoder sends OPEN_MEDIA_CHANNEL message, and receives OPEN_MEDIA_CHANNEL_ACK message from the mobile client, the transcoder starts the transcoding process by starting the instances that had been created in the previous stage by querying MCM about the instances to be started.
6. Finally, the bit stream from the H.323 client is transcoded into the desired format, and transmitted to the mobile client.

Phase B is as follows.

1. At the end of communication, the mobile client sends STOP_MEDIA_TRANSMISSION message to the transcoder.
2. Once the transcoder receives STOP_MEDIA_TRANSMISSION message, the transcoder will only stop process instances that are not shared with other clients. Then, those instances and corresponding buffers are deleted.
3. Call termination is initiated by sending DISCON-

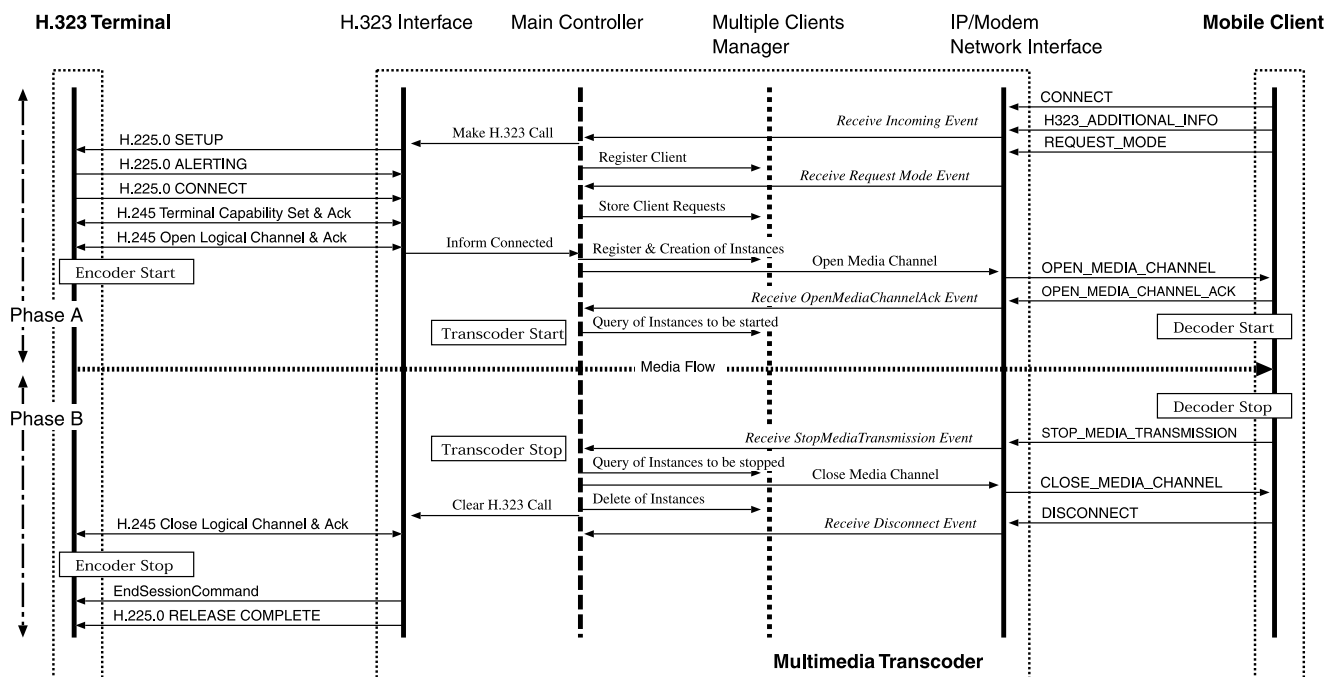


Fig. 7 Call message sequence.

NECT message from the mobile client. Alternatively, the mobile client could send a new REQUEST.MODE message to start receipt of another audiovisual combination from the same source.

4. Once the transcoder receives DISCONNECT message, the transcoder sends H.245 CloseLogicalChannel messages to the H.323 client, which closes the video/audio channels. These messages are then followed by an endSessionCommand message. The H.323 client will also send those messages in the reverse direction. Finally H.225.0 RELEASE COMPLETE message is sent by the transcoder, and the call is terminated.

5. System Performance and Implementation

In this section, we describe the performance of our systems, and the platforms for which it has been developed.

5.1 Objective Video Transcoding Performance

This section gives objective coding results for transcoding the CIF Salesman sequence, originally encoded at 128 kbit/s, into QCIF at 8–24 kbit/s. The three methods for evaluation are ± 8 pel full search followed by a half pel refinement (Method A), re-use of the decoded motion vector (Method B) and rounds the decoded motion vector to an integer vector, and then performing a ± 1 pel search around it, followed by a half pel refinement (Method C).

Table 2 shows luminance SNR and, in parenthesis, the average number of bits to code each picture, for three different motion vector generation methods.

From these results, it can be seen that method C provides a good compromise between picture quality and computational complexity, and so was used in our implementation.

5.2 Transcoding Delay

The extent to which media data is delayed in the transcoder was measured for the H.323 to IP mobile client scenario. The delay between a picture header being received at the H.323 interface and the corresponding picture header being transmitted by the IP network interface was measured. The results are shown in Fig. 8. It is thought that the higher delay of around 50 ms for some pictures is due to the behavior of the

PC running the transcoder, which runs other unrelated processes from time to time. However, we conclude that the transcoding delay is quite low as the average delay was 26 ms, and quite low even if it is compared with other delays such network delay and buffering delay in decoder.

5.3 Transcoding Processing Power Requirements

The processing power requirements of the transcoder were assessed by measuring CPU usage on a 600 MHz Pentium III. Figures 9 and 10 show the relationship between the number of clients and CPU usage for uni-directional media. Figure 9 relates to the scenario of one H.323 client and multiple mobile clients. Figure 10 relates to the scenario of multiple H.323 clients and one mobile client.

From both figures, CPU usage is directly proportional to the number of terminals. This is because additional instances of processes are created according to each request. It can be seen that about 15% of CPU is required for video transcoding and the H.323 interface, that about 4.5% of CPU is required for the video encoder sub-system, and that about 10% of CPU is required for the audio transcoder.

As the PC used in this experiment can support

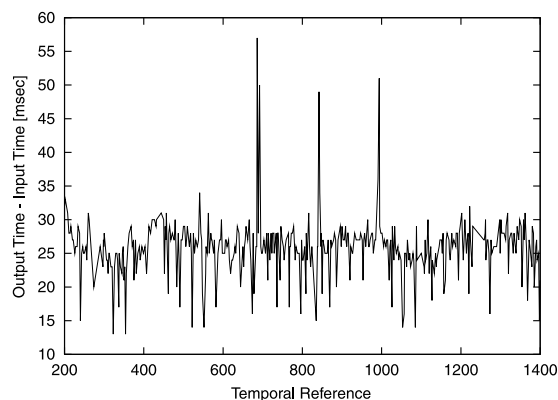


Fig. 8 Delay imposed by the transcoder vs picture number.

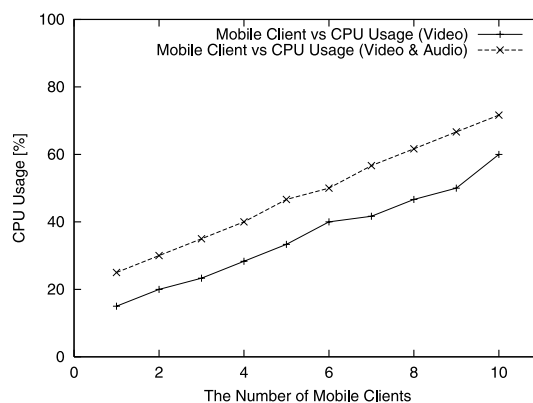


Fig. 9 Transcoder CPU usage vs number of mobile clients.

Table 2 SNR and bits/picture.

Bit Rate	Method A	Method B	Method C
24	34.74 (4502)	33.73 (4539)	34.51 (4536)
16	33.04 (3061)	31.94 (3051)	32.67 (3048)
8	29.92 (1525)	28.78 (1528)	29.40 (1541)

more than 10 clients, we can conclude that the implemented transcoder can satisfy the requirements of video conferencing.

5.4 Implementations

The multimedia transcoder application runs on PC platforms. The application interface is shown in Fig. 11(a). It can display the decoded images from H.323 clients and the locally encoded images sent to mobile clients. It can also show the call messages between clients, statistics of packets and SNR, the requests of all clients etc.

The mobile client application runs on PC platforms, and requires negligible processing power to decode audio and low bit rate QCIF video. The mobile client application has been adapted for PocketPC, which, when running on a Compaq iPaq H3630 or a Casio CASSIOPEIA E-750, is able to receive and decode AMR audio and QCIF H.263 baseline video simultaneously. The application interface is shown in Fig. 11(b).

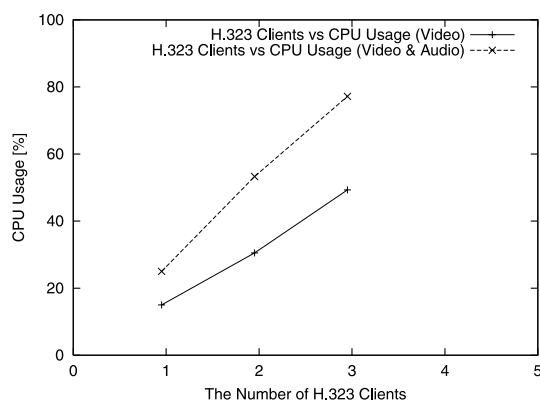


Fig. 10 Transcoder CPU usage vs number of H.323 clients.

6. Conclusion

We have developed a multimedia transcoder that enables mobile access to video conferencing for practical use purposes. This multimedia transcoder system, provides interoperability between the video conferencing terminals on IP networks and mobile terminals on mobile networks. This transcoder supports simultaneous connections from multiple clients on both the IP and mobile networks. The system performance was described. It was shown that the video quality and transcoding delay is sufficient to make the system usable, and hence we have achieved our aim of providing service level interoperability across heterogeneous networks. Such transcoding devices are now practical and enable a range of benefits for service interoperability, including video-enhanced mobile access to teleconferencing. Future issues include how to address processing power limitations of the transcoder, including sharing load between an array of transcoding PCs.

Acknowledgement

The research and development of the transcoder described in this paper was performed as part of the cooperation between Waseda University and BTextact Technologies. Authors thank Dr. Geoff Morrison of BTextact Technologies and Prof. Wataru Kameyama, Prof. Mitsuji Matsumoto and Prof. Sakae Okubo of GITI, Waseda University for enabling the cooperation between Waseda University and BTextact Technologies, and for supporting the transcoder project.



(a) Transcoder application on PC.



(b) Mobile client on PocketPC.

Fig. 11 Application interfaces.

References

- [1] D.G. Morrison, M.E. Nilsson, and M. Ghanbari, "Reduction of the bit-rate of compressed video while in its coded form," Packet Video Workshop, Sept. 1994.
- [2] H. Sun, W. Kwok, and J.W. Zdepski, "Architecture for MPEG compressed bitstream scaling," IEEE Trans. Circuits and Systems for Video Technology, vol.6, no.2, pp.191-199, April 1996.
- [3] P.N. Tudor and O.H. Werner, "Real-time transcoding of MPEG-2 video bits streams," International Broadcasting Convention, pp.286-301, Feb. 1997.
- [4] H. Kasai, H. Tominaga, T. Hanamura, and W. Kameyama, "Rate control scheme for low delay MPEG-2 video transcoder," IEICE Trans. Commun. (Japanese Edition), vol.J83-B, no.2, pp.151-164, Feb. 2000.
- [5] H. Kasai, I. Nagayoshi, W. Kameyama, and H. Tominaga, "Rate control method in MPEG bitstream bitrate scaling considering drift propagation," IEICE Trans. Inf. & Syst. (Japanese Edition), vol.J83-D-II, no.2, pp.509-524, Feb. 2000.
- [6] H. Kasai, M. Sugiura, T. Hanamura, W. Kameyama, and H. Tominaga, "Video rate control scheme for MPEG-2 multi-program TS transcoder," IEICE Trans. Commun. (Japanese Edition), vol.J84-B, no.10, pp.1877-1889, Oct. 2001.
- [7] H. Kasai, I. Nagayoshi, T. Hanamura, W. Kameyama, and H. Tominaga, "Quantization parameter control scheme for MPEG-2 video transcoder," IEICE Trans. Commun. (Japanese Edition), vol.J85-B, no.2, pp.278-287, Feb. 2002.
- [8] International Telecommunication Union, "Packet-based multimedia communications systems," Recommendation H.323, Nov. 2000.
- [9] <http://www.microsoft.com/>
- [10] <http://www.intel.com/>
- [11] <http://www.smithmicro.com/>
- [12] International Telecommunication Union, "Call signalling protocols and media stream packetization for packet-based multimedia communication systems," Recommendation H.225.0, Nov. 2000.
- [13] International Telecommunication Union, "Control protocol for multimedia communication," Recommendation H.245, July 2001.
- [14] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," RFC 1889, Jan. 1996.
- [15] <http://www.eurescom.de/>
- [16] International Telecommunication Union, "Video codec for audiovisual services at $p \times 64$ kbit/s," Recommendation H.261, 1993.
- [17] International Telecommunication Union, "Video coding for low bitrate communication," Recommendation H.263, June 2001.
- [18] International Telecommunication Union, Pulse code modulation (PCM) of voice frequencies, Nov. 1988.
- [19] International Telecommunication Union, "Speech coders: Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s," March 1996.
- [20] MPEG. MPEG-2 advanced audio coding, AAC. International Standard IS 13818-7, ISO/IEC JTC1/SC29 WG11, 1997.
- [21] MPEG. MPEG-4 Audio. International Standard IS 14496-3, ISO/IEC JTC1/SC29 WG11, 1998.
- [22] European Telecommunications Standards Institute (ETSI) Adaptive Multi-Rate (AMR) speech coding. 1998.
- [23] International Telecommunication Union, "Multiplexing protocol for low bit rate multimedia communication," Recommendation H.223, March 1996.
- [24] <http://www.3gpp.org/>
- [25] C. Zhu, "RTP payload format for H.263 video streams," RFC 2190, Sept. 1997.
- [26] International Telecommunication Union, "Multiplexing protocol for low bit rate multimedia mobile communication over low error-prone channels," Recommendation H.223 Annex A, Feb. 1998.
- [27] International Telecommunication Union, "Multiplexing protocol for low bit rate multimedia mobile communication over moderate error-prone channels," Recommendation H.223 Annex B, Feb. 1998.
- [28] International Telecommunication Union, "Terminal for low bit rate multimedia communication," Recommendation H.324, March 2002.
- [29] J. Youn, M.T. Sun, and J. Xin, "Video transcoder architecture for bit rate scaling of H.263 bit streams," ACM International Multimedia Conference, pp.243-250, Oct. 1999.
- [30] J. Youn, M.T. Sun, and C.-W. Lin, "Motion estimation for high performance transcoding," IEEE Trans. Consumer Electronics, vol.44, no.3, pp.649-658, Aug. 1998.
- [31] A. Shimizu, A. Sagata, H. Watanabe, and N. Kobayashi, "Bitstream scaling by reference frame altering," IEICE Trans. Inf. & Syst. (Japanese Edition), vol.J83-D-II, no.3, pp.967-976, March 2000.
- [32] European digital cellular telecommunications system (Phase 1), Full rate speech; Transcoding (GSM 06.10).
- [33] <http://www.capi.org/>
- [34] <http://www.eicon.com/>



Hiroyuki Kasai received the B.E., M.E., and D.E. degrees in Electronics, Information and Communication Engineering from Waseda University, Tokyo, Japan, in 1996, 1998 and 2000, respectively. He was a research associate at the Global Information and Telecommunication Institute (GITI) from 1998 to 2002. He was also a visiting researcher at BText Technologies, United Kingdom, from 2000 to 2001. He is currently working in

Network Laboratories at NTT DoCoMo, Inc. from 2002. He is a member of the Institute of Image Electronics Engineers Japan (IEEEJ). His interests include video compression and video transcoding, and delivery of multimedia services over heterogeneous networks.



Mike Nilsson received the B.A. degree in Engineering Science from Exeter College Oxford in 1988. He is a chartered engineer currently working in the Advanced Video Services team at BText Technologies, whom he represents in the ITU-T, where he is the editor of Recommendation H.245. His interests include video compression and video transcoding, and the provision of video services over heterogeneous networks.



Tim Jebb has worked for British Telecommunications plc since 1984, during which time he was awarded a B.A. degree in Physics from the Open University. He has worked in a number of areas, all related to computer communications, including maintaining IBM SNA networks, network management, OSI, and distributed databases. He is currently working in the Advanced Video Services team at BTexact Technologies. His inter-

ests include video compression, the provision of video services over heterogeneous networks, and audioconferencing.



Mike Whybray received a B.Sc. degree in Electronics from the University of Birmingham, UK, in 1977, and an MSc 1983. He has worked on the reliability of CMOS and magnetic bubble memories, development of videophones using DSPs, and image interpretation. He currently leads a team at BTexact Technologies developing video coding and multimedia streaming applications. He is a Member of the IEE, and a Chartered En-

gineer.



Hideyoshi Tominaga received the B.E., M.E., and D.E. degrees from Waseda University, Tokyo, Japan, in 1962, 1964 and 1974, respectively. From 1964 to 1971, he worked for NTT Electric Communications Laboratories. Since 1971 he has been with Waseda University. Currently he is a professor of Department of Electronics, Information and Communication Engineering. His research area includes picture coding, switching sys-

tems and information security. He has been involved in standardization activities, such as MPEG, DAVIC, ITU-T, and ISO-SGs. He founded the Global Information and Telecommunication Institute (GITI), Waseda University in 1996, and also Graduate School of Global Information and Telecommunication Studies, Waseda University in 2000. He is a senior member of IEEE and a member of Information Processing Society of Japan (IPSJ), the Institute of Image Electronics Engineers Japan (IEEEJ) and the Institute of Image Information and Television Engineers (ITE).