MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Network Transcoder with Seamless Switching Function

Naoki Kuwahara, Toshihiko Hata, Toshiharu Nozawa, Anthony Vetro

TR2004-146 December 2004

Abstract

In a surveillance system with a huge number of cameras, the number of videos to be transmitted and displayed is uaually restricted by network bandwidth and the resource of display terminal. Given that the source video is captured at high quality, a network transcoder is used to send video with lower data rate as teh default for ordinary scenes, while only extraordinary or unusual scenes are sent with higher quality. With such a scenario, it is necessary to switch from a low quality version of the video to a higher quality video with low latency and in a seamless manner. This paper presents a network transcoder that is able to change the content and the quality of videos seamlessly and with low latency. The novelty of the proposed scheme is possible to change the quality and camera in the same session. Moreover, this paper describes an RTSP enhancement that enables this dynamic transcoding function. Finally, an evaluation of the results is provided.

SPIE Multimedia Systems and Applications

Network transcoder with seamless switching function

Naoki Kuwahara*[a], Toshihiko Hata[a], Toshiharu Nozawa[a], Anthony Vetro[b]

[a] Mitsubishi Electric Corporation, Advanced Technology R&D Center
8-1-1 Tsukaguchi-honmachi, Amagasaki City, Hyogo Prefecture, 661-8661, Japan
[b] Mitsubishi Electric Research Laboratories, Technology Laboratory
201 Broadway, Cambridge, MA 02139, USA

ABSTRACT

In a surveillance system with a huge number of cameras, the number of videos to be transmitted and displayed is usually restricted by network bandwidth and the resource of display terminal. Given that the source video is captured at high quality, a network transcoder is used to send video with lower data rate as the default for ordinary scenes, while only extraordinary or unusual scenes are sent with higher quality. With such a scenario, it is necessary to switch from a low quality version of the video to a higher quality video with low latency and in a seamless manner. This paper presents a network transcoder that is able to change the content and the quality of videos seamlessly and with low latency. The novelty of the proposed scheme is possible to change the quality and camera in the same session. Moreover, this paper describes an RTSP enhancement that enables this dynamic transcoding function. Finally, an evaluation of the results is provided.

Keywords: Surveillance, Network Transcoder, RTSP, Low Latency, Seamless Switching

1. INTRODUCTION

In the monitoring system of rivers and roads, the number of cameras to be set is dramatically increasing up to some hundreds. In such a system, we can see the state of any remote site of anytime from anywhere by transmitting the videos via IP networks and mobile networks. One key component to realize that system is a network transcoder[1]. A network transcoder reduces the bit-rate of a video encoded at high quality for surveillance use and retransmits it. Therefore, a transcoder is used in browsing under narrow band network and displaying plural videos concurrently at the same terminal without exhausting the resource of networks and terminal[2].

For the human interface of the system, it is impossible to display all of the videos simultaneously. Videos are usually displayed switch by switch periodically or on demand locally. However, it is not allowed to miss an important scene at all for surveillance use. So, in an emergency, the videos on the display must be changed to the corresponding cameras immediately. Additionally, the quality of the video is also required to turn higher without a visually break for important scenes. This kind of requirement is applied to other applications to handle live videos over networks, e.g., sports broadcasting, videoconference, etc. That is, such a seamless switching function at the network transcoder can impact the service of the systems.

In a video streaming system over networks, such a seamless switching service for a realtime video has not been extensively studied, although a relay server that switches among stored videos seamlessly based on the a preset time schedule has been reported. Therefore, we propose a network transcoder that has a seamless switching function for surveillance system.

The rest of this paper describes surveillance system with multiple cameras and the network transcoder, and then provides the architecture of the seamless switching. Then, we propose an RTSP enhancement for this kind of transcoder system. Finally, we present an evaluation of the result.

2. SURVEILLANCE SYSTEM WITH MULTIPLE CAMERAS

2.1. System structure

Fig.1 shows the system structure of river and road monitoring with multiple cameras. Hundreds of cameras with encoder are set at local sites and the encoded video is transmitted to the network. The encoding format is normally MPEG-2 MP@ML or Motion JPEG (M-JPEG) for surveillance use. These imply approximately 6Mbps or 10Mbps in bit-rate for each video stream. A video server stores the videos endlessly or by an alarm and sends the requested video clip as a stream on demand. Watchmen usually monitor the videos by the display terminal inside the surveillance center. Among the encoders, video servers and display terminals are connected by a wideband network such as optical fibers. Since the bit-rate of the video is high, when multiple videos are displayed on the same terminal concurrently, a network transcoder reduces the bit-rate according to the importance of video in realtime. Then, videos can be delivered to remote personal devices by adjusting the video stream to the Internet or mobile network by means of a transcoder. On detecting an unusual scene or triggering an alarm, the video, namely the camera, to be displayed, is switched to the most appropriate camera and the quality of this view is improved.

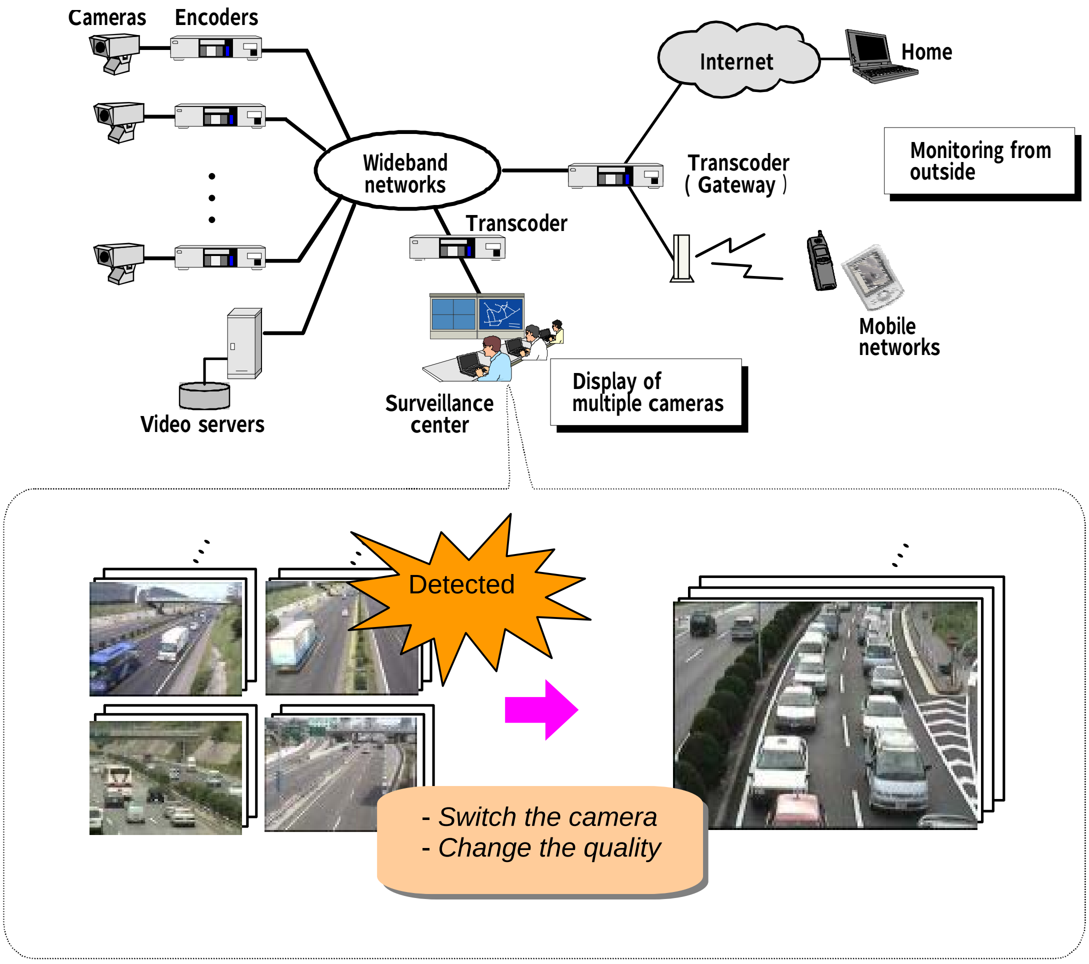

Fig. 1: System structure

2.2. Network transcoder

2.2.1. Architecture

Fig.2 is a diagram of a general network transcoder. It basically consists of 5 sequential parts, (1) receiving (RECV), (2) de-multiplexing (DEMUX), (3) transcoding (TC), (4) multiplexing (MUX), (5) sending. DEMUX part divides the input stream that is multiplexed by video and audio into elementary streams. The transcoding part reduces the bit-rate of the elementary stream by reducing the spatial resolution as well as changing the quantization parameter. Basically, the transcoding part is a combination of decoder and encoder whose operation is fixed by the input and output format. Generally speaking, the complexity of decoding and encoding is too high, so the study of the algorithm to optimize the transcoding operation has been widely conducted[3,4]. The MUX part multiplexes transcoded videos and audios into a stream if necessary. Finally, the SEND part transmits it as an output stream.

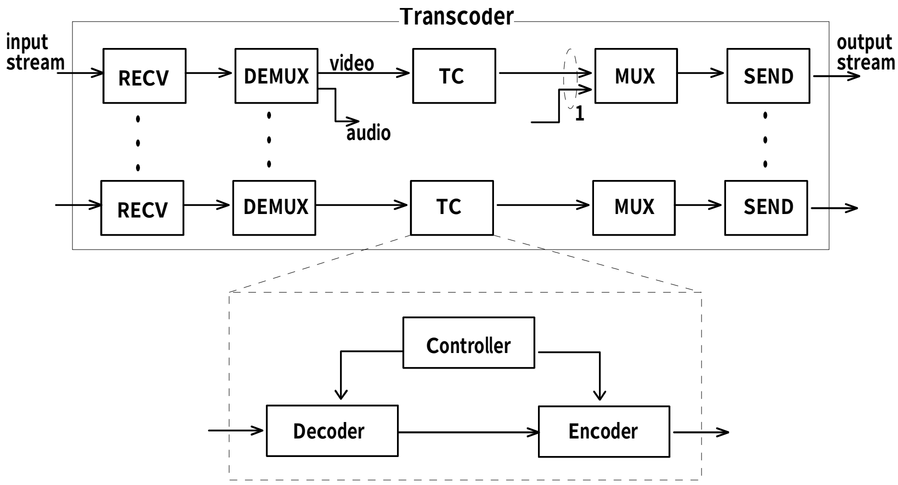

Fig. 2: Diagram of the transcoder

2.2.2. Transcoder configuration

The transcoder supports a variety of configuration parameters to accomplish the bit-rate reduction according to various uses and contests. Table 1 shows the configuration parameters of transcoding.

Table 1: Configuration parameters of transcoding

Parameter	Semantics
Encoding format	Converts to suitable format for an objective and content. For example, converting to JPEG for monotonous scene, while converting to MPEG-4 as the motion is important. Then, converting to the acceptable format to the display terminal. (Our prototype supports MPEG-2, -4 and M-JPEG as an output)
Down-sampling	Reduces the spatial resolution in video (Our prototype can change to half or quarter)
Re-quantization	Changes the quantization parameter to the rougher.
Frame rate	Drops some frames periodically. B and P picture are usually candidate to be skipped
Region of Interest (ROI) Position and size of ROI Encoding format Down-sampling rate Re-quantization Frame rate	Encodes the image at higher quality for some important areas than other background areas. All of the listed parameters to the left can be set up for each area respectively.

3. IMPLEMENTATION OF SEAMLESS SWITCHING

Two types of seamless switching function are required in the network transcoder. One is the switching of a source (camera switching) and the other is for transcoder configuration. The switching transcoder configuration seamlessly is important technique in dynamic rate control to adjust the bit-rate according to the conditions of the network bandwidth for output.

3.1. Camera switching

To switch a source, the conventional transcoder would terminate the connection for session control of the current stream and establish a new connection for new source. In this case, the video is interrupted visually during the switch and the response until the video is switched back is typically delayed. The reason why the conventional transcoder performs this way is that it does not provide the specific function (interface) to switch only a source for the clients, although one important aim of the transcoder is to relay a stream.

Consequently, a simple means to achieve this switching function in the transcoder is needed. Fig.3 shows the procedure of switching from camera #m to camera #n. On receiving the request message for switching, the transcoder first establishes a new session to the requested new camera. When the random access point in the new bitstream is found, the transcoder changes to the new source, and then the session for previous camera is terminated. The random access point is the next intra picture when the bitstream is inter-frame coded, while it is a next frame when the bitstream is intra-frame coded. Thus, our new transcoder switches the video without a break, i.e., seamlessly. Additionally, it is expected to raise the probability of improving the response time (see Fig.4). Since the processing time of proposed method t_2 is shorter than the conventional method t_1, the response time becomes better when the intra picture arrives in the period of t_1 - t_2.

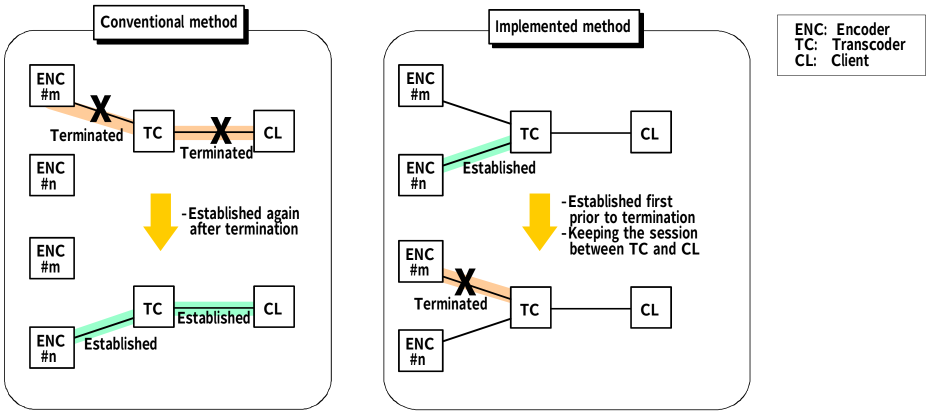

Fig. 3: Procedure of switching cameras

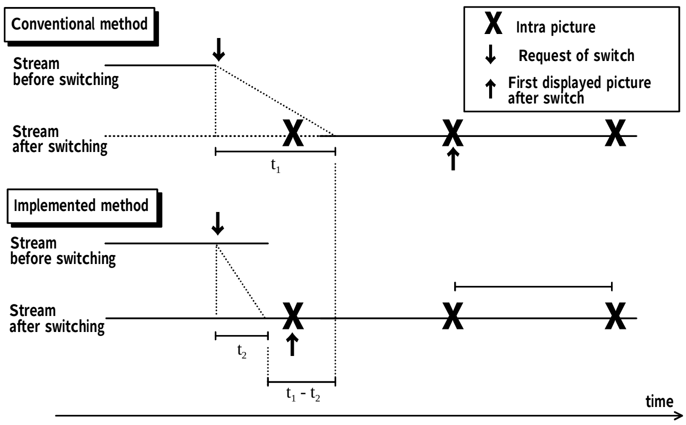

Fig. 4: The timing of switching

As for the system clock of the decoder at the client side, since the bitstream that the transcoder outputs uses its own clock which is produced by multiplexing part in the transcoder, the gap of the clock occurring by switching a source does not impact the display of video at the client.

3.2. Configuration switching

3.2.1. Transcoder configuration

Our transcoder has been implemented so as to accept the request for the configuration change at any time. In transcoding, the new set of configuration parameters takes effect from the start of the next Group of Pictures (GOP).

3.2.2. Notification to client

When the configuration parameters of transcoder are changed, the client should know it in synchronization with bitstream. At least, the point that encoding format has changed must be given. One solution for that is to embed the notification within the bitstream. Our prototype uses RTP[5] as a transport protocol for an audio-video source, and then we used the payload type (PT) field in RTP header to convey the encoding format. On detecting the change in the PT field, the client switches the decoder.

3.2.3. Preparation at client

As a preparation at a client side, a client launches as many decoders as needed so as not to break in video by launching a process and initializing it.

.

4. STREAM CONTROL PROTOCOL

4.1. Stream control protocol in this system

Fig.5 shows the relationship of the stream control protocols in this system. All the request messages are issued from clients and transcoder issues the request messages to the corresponding encoder or video server if necessary. We used RTSP[6], which is a standardized protocol at IETF as a stream control protocol between transcoder and camera or video server. But, the protocol between client and network transcoder does not meet the standardized ones because they do not assume a transcoder in general, especially, novel functions such as transcoding and seamless switching. So, we made enhancement of RTSP for a general transcoder. To define the protocol for a general transcoder, we build a typical model of the transcoder function.

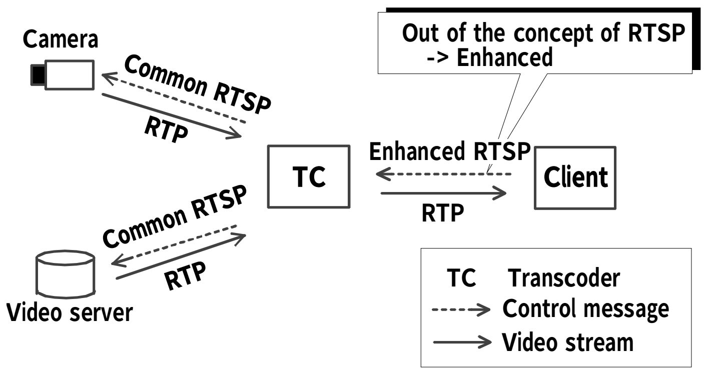

Fig. 5: Stream control protocols in this system

4.2. General model of network transcoder

4.2.1. Model for Input / Output

The assumed model for input and output of transcoder is illustrated in Fig.6 and Table 2. In this model, a network transcoder allows the system to input multiple streams and transcode to multiple output streams concurrently.

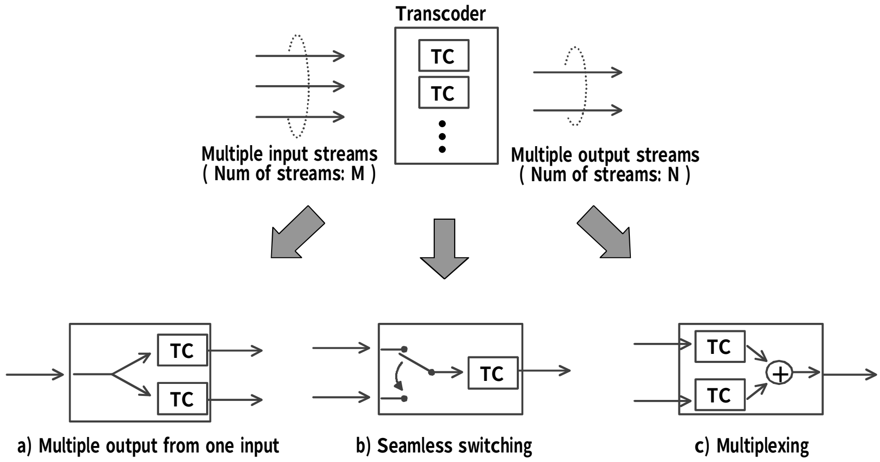

Fig. 6: I/O model

Table 2: I/O model

Multiple output from one input	Duplicates an input stream and converts each stream with different configuration parameters and transmits them
Seamless switching	Switches sources and quality rapidly and without a break in the same session
Multiplexing	Multiplexes plural input streams or parts of them

4.2.2. Model for transcoding

Assumed configuration parameters were shown in Table 1. In handling multiple streams, these parameters could be set for each stream separately. Besides, for ROI transcoding, the parameters could be set for each region as well. Then, for seamless switching, the configuration can be changed in the same session.

4.2.3. Model for stream control

For stream control set, it is expected to support the following control requests; play, reverse play, speed control, seek and pause.

4.3. RTSP enhancement

Based on the above models, we have enhanced RTSP for the network transcoder.

4.3.1. URL

In usual RTSP, a URL specifies the source to access. But, in this system, there is a possibility to alter sources within the same session. So, we decided not to set information about an accessing source in URL. An accessing source is specified in ANNOUNCE method as described later. Each session is identified with session ID in RTSP header. Only the name of transcoder to connect is described in URL.

【URL】 *rtsp://transcoder*
transcoder = <name of the network transcoder>

4.3.2. Method

Table 3 shows the methods that are used in enhanced RTSP.

Table 3: List of the methods

Method	Description
ANNOUNCE	Specifies streams to include inside the session
SETUP	Creates a new session defined by ANNOUNCE in advance, and specifies the delivery way and destination, too
SET_PARAMETER	Specifies the configuration parameters in transcoding
PLAY	Sets the start position (seek), requires to start to play the video, specifies the parameters of play such as a speed and direction
PAUSE	Requires the stop of the play
TEARDOWN	Terminates the session

Some of methods correspond to the internal processing of transcoder such as Fig.7. For example, SET_PARAMETER is used only for setting the transcoding configuration. SETUP is used for specifying the parameters related stream delivery as well. Then, ANNOUNCE is used to specify audio-video sources that construct the session to be requested. On the contrary, PLAY, PAUSE are not for a specific processing but for controlling the status of streams.

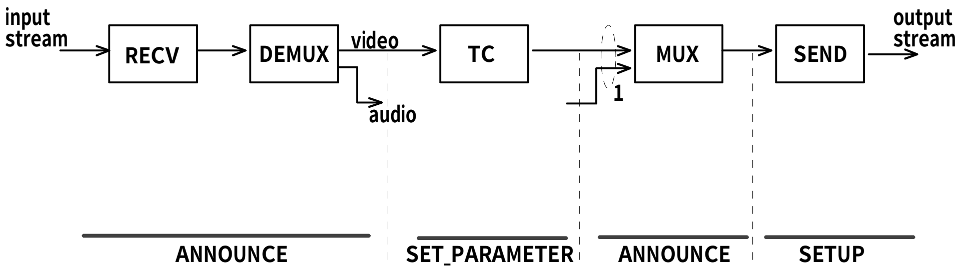

Fig. 7: Relation between method and internal processing of transcoder

4.3.3. Definition of transmitting streams

The attribute tag of SDP[7], "a=control:", is used to list input sources at the message-body in ANNOUNCE method. For multiplexing plural sources to include in the session are described with a source ID which is assigned for each source. The sample message is shown in List 1.
Then, the destination to deliver a stream is described at the transport-header of SETUP method as well as RTSP standard. Of course, the destination address is not limited to the client address for multicast.

4.3.4. Set of transcoding configuration

Configuration parameters of transcoding are described at the message-body of SET_PARAMETER by using the original tag. When plural sources are multiplexed, the configurations can be defined for each source. The configuration is connected to the source by a source ID that has been defined in the previous ANNOUNCE method. Besides, with ROI transcoding, the parameters are set for each region independently. Then, the default value is adopted if not specified. List 2 is a sample message of SET_PARAMETER.

4.3.5. Switching of cameras

To request a switching of a source seamlessly, a client issues an ANNOUNCE message in that a new set of sources are specified. Even if the change is a part of the set, all of the sources should be described so that reducing sources from the set can be implemented. ANNOUNCE message for seamless switching has a session ID to distinguish from the ANNOUNCE message issued prior to the SETUP message.

```
ANNOUNCE rtsp://transcoder1.melco.co.jp
  Cseq:1
  Session:12345678                  /* sessionID */
                                    /* Described only after setup */
  Content-type:text/parameters

    num_of_source:2                 /* The number of input sources */
    source:1                        /* Source identified as 1 */
      a=control:rtsp://yodogawa-river.go.jp/live/1
                                    /* The description for the first source */

    source:2                        /* Source identified as 2 */
      a=control:rtsp://video_server.go.jp/movie/7
                                    /* The description for the second source */

    source:3                        /* Source identified as 3 */
      a=control:rtsp://video_server.go.jp/audio/7
                                    /* The description for the third source */
```

List 1: A sample of ANNOUNCE message

```
SET_PARAMETER rtsp://transcoder1.melco.co.jp
  Cseq:3
  Session:12345678
  Content-type:text/parameters

    num_of_source:2                 /* The number of sources */
    source:1                        /* Source identified as 1 */
      format:MPEG4                  /* Output format is MPEG4 */
      scaleX:1/2                    /* Horizontal resolution to be half */
      scaleY:1/2                    /* Vertical resolution to be half */
      framerate:10                  /* Frame rate is 10 [frame/sec] */

    source:2                        /* Source identified as 2 */
      num_of_region:2               /* Transcoding with 2 ROIs */
      regionID:1                    /* The configuration for first region */
        rectangle:                  /* The shape of ROI is rectangle */
          coordinate:0,0            /* Coordinates for this region */
          coordinate:719,479
        format:MJPEG                /* Output format for this region */
        framerate:2                 /* Frame rate for this region */

      regionID:2                    /* The configuration for second region */
        polygon:4                   /* The shape of ROI is polygon with 4 apices */
          coordinate:10,10          /* Coordinates for this region */
          coordinate:200,10         /* Apices are defined at clockwise rotation */
          coordinate:200,300
          coordinate:10,300
        format:MJPEG                /* Output format for this region */
        framerate:15                /* Frame rate for this region */
```

List 2: A sample of SET_PARAMETER message

4.3.6. Example of message sequence

Fig.9 shows a sample of message sequence in this system. In this diagram, the stream of camera A is played first and then camera is switched to B. First of all, ANNOUNCE message is issued, then, SETUP message is issued. Since the first ANNOUNCE message does not have the session ID, these message are issued at the same TCP connection to associate with continuous SETUP message. Once setup is completed, any messages other than SETUP can be issued in any order.

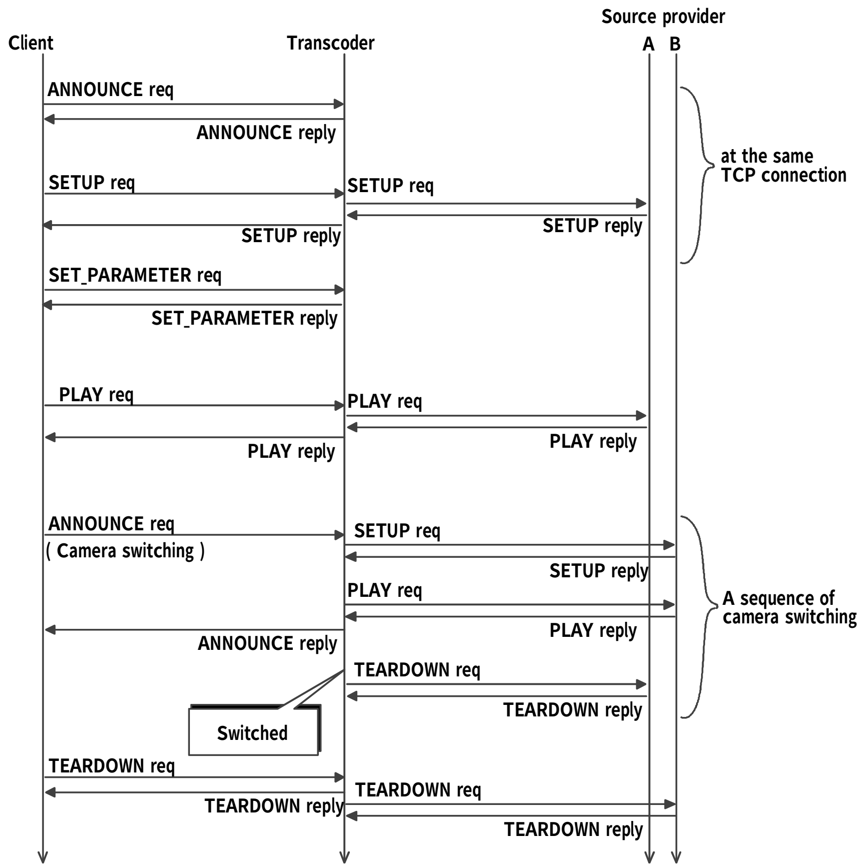

Fig. 8: A sample of message sequence

5. EXPERIMENTAL RESULTS

5.1. Measurement environment and items

To evaluate the effectiveness of the seamless switching function, we measured the consuming time 1) to switch a source and 2) to switch a transcoding configuration in the conventional system and the proposed system respectively. Then, we checked the smoothness in switching with eyes. We measured the time from issuing a request message until passing the first image data to the decoder after switching by using windows' system call, PerformanceCounter(). For 1), we issued the request message to switch from a video server to an encoder, and for 2) the request to change the encoding format from MPEG-4 to MPEG-2 is issued. The average of 5 times measurements is calculated as the result. In the conventional system, we measured the time to transact a message sequence of TEARDOWN => SETUP => SET_ PARAMETER => PLAY.

Table 4: Spec of measurement environment

Encoder	MPEG-2 encoder: MPE, produced by Mitsubishi Electric Corp. 6Mbps, GoP period = 0.5 [sec]
Video server	Pentium4, 1.6GHz, RAM128MB, Windows 2000
Network transcoder	Pentium4, 2.0GHz, RAM1GB, Windows 2000 Transcoding configuration: MPEG-4, 10fps, 384Kbps
Client	Pentium4, 2.0GHz, RAM512MB, Windows XP MPEG-2 decoder: PCI board, MPEG-4 software decoder
Network	100Base Ethernet switching hub

5.2. Results

The measurement results are shown in Table 5. It can be seen that the proposed method contributes to reduction of the switching time in both cases, although there is a possibility to include an error of the GoP period at most.

In subjective tests, the impression of view also improved for the switching of camera. But, two problems were found in switching the encoding format. Both problems depend on the MPEG-2 hardware decoder that is used at this time. The first problem is to take about 2 seconds for the decoder to restart from the pause mode that is set while being not in use to avoid underflow. Table 6 shows the measurement results including the decoder's processing time. The delay of restarting is less than that of initializing (the number of conventional method in Table 6), but it is not acceptable in practical use. The second problem is internal buffer within the hardware decoder. Since the decoder does not start until the buffer is filled to some extent, the display of the video was interrupted for a while. Thus, to avoid such a decoder-specific problem, the proper selection of decoder is indispensable in designing the system to realize the seamless switch of encoding format. Especially, the use of a software decoder seems to be preferable.

Table 5: Measurement result (without decoder processing time)

	Conventional method	Proposed method
Switching of camera	922 ms	223 ms
Switching of encoding format	922 ms	201 ms

Table 6: Measurement result (with decoder processing time)

	Conventional method	Proposed method
Switching of camera	4531 ms	223 ms
Switching of encoding format	4531 ms	2215 ms

6. SUMMARY & FUTURE WORK

This paper presented a network transcoder with seamless switching function for surveillance use. We proposed the method of switching with the same session to accomplish the seamless switching. Then, we enhanced RTSP to implement the functions for transcoder. Finally, we showed the effectiveness of proposed method by our prototype. For camera switching, we described that the response and subjective impression became better. Besides, the choice of the decoder is also important to change the encoding format seamlessly.

In this paper, we supposed that a client has enough information about streaming, namely, it has been assumed that the list of accessible sources and the configuration parameters in multicasting is obvious. But in a real system, we have to provide a way to give a client such information. One of our interests is the construction of the system that can deliver such information in realtime and in synchronization with bitstream.

REFFERNCE

1. Vetro, A. et al., *Video Transcoding Architectures and Techniques: An Overview*, IEEE Signal Processing Magazine, 20, Issue 2, 18~29, 2003
2. Vetro, A. et al., *Complexity-Quality Analysis of MPEG-2 to MEPG-4 Transcoding Architectures*, IEEE International Conference on Consumer Electronics (ICCE), pp.130-131, June 2002.
3. Kuwahara, N. et al., *Map-Top Video Surveillance with Adaptive Video Delivery*, World Multi-Conference on Systemics, Cybernetics and Informatics (ISAS-SCI), 2002
4. Nishimura, S. et al., *A Development of the Real-time MPEG-2 Video Transcoder*, IPSJ SIGNotes Audio Visual and Multimedia information processing No.25-005
5. Schulzrinne, Casner, Frederick and Jacobson, "RTP: A Transport Protocol for Real-Time Applicaions", RFC1889, IETF, Jan. 1996
6. Schulzrinne, Rao and Lanphier, "Real Time Streaming Protocol (RTSP)", RFC2326, IETF, April 1998
7. Handley and Jacobson, "SDP: Session description protocol", RFC2327, IETF, April 1998