International Telecommunication Union

# ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

# H.627

(06/2012)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Broadband, triple-play and advanced multimedia services – Advanced multimedia services and applications

## Signalling and protocols for visual surveillance

Recommendation ITU-T H.627

ITU-T

ITU International Telecommunication Union

ITU-T H-SERIES RECOMMENDATIONS

**AUDIOVISUAL AND MULTIMEDIA SYSTEMS**

| | |
|---|---|
| CHARACTERISTICS OF VISUAL TELEPHONE SYSTEMS | H.100–H.199 |
| INFRASTRUCTURE OF AUDIOVISUAL SERVICES | |
| General | H.200–H.219 |
| Transmission multiplexing and synchronization | H.220–H.229 |
| Systems aspects | H.230–H.239 |
| Communication procedures | H.240–H.259 |
| Coding of moving video | H.260–H.279 |
| Related systems aspects | H.280–H.299 |
| Systems and terminal equipment for audiovisual services | H.300–H.349 |
| Directory services architecture for audiovisual and multimedia services | H.350–H.359 |
| Quality of service architecture for audiovisual and multimedia services | H.360–H.369 |
| Supplementary services for multimedia | H.450–H.499 |
| MOBILITY AND COLLABORATION PROCEDURES | |
| Overview of Mobility and Collaboration, definitions, protocols and procedures | H.500–H.509 |
| Mobility for H-Series multimedia systems and services | H.510–H.519 |
| Mobile multimedia collaboration applications and services | H.520–H.529 |
| Security for mobile multimedia systems and services | H.530–H.539 |
| Security for mobile multimedia collaboration applications and services | H.540–H.549 |
| Mobility interworking procedures | H.550–H.559 |
| Mobile multimedia collaboration inter-working procedures | H.560–H.569 |
| BROADBAND, TRIPLE-PLAY AND ADVANCED MULTIMEDIA SERVICES | |
| Broadband multimedia services over VDSL | H.610–H.619 |
| **Advanced multimedia services and applications** | **H.620–H.629** |
| Ubiquitous sensor network applications and Internet of Things | H.640–H.649 |
| IPTV MULTIMEDIA SERVICES AND APPLICATIONS FOR IPTV | |
| General aspects | H.700–H.719 |
| IPTV terminal devices | H.720–H.729 |
| IPTV middleware | H.730–H.739 |
| IPTV application event handling | H.740–H.749 |
| IPTV metadata | H.750–H.759 |
| IPTV multimedia application frameworks | H.760–H.769 |
| IPTV service discovery up to consumption | H.770–H.779 |
| Digital Signage | H.780–H.789 |

*For further details, please refer to the list of ITU-T Recommendations.*

# Recommendation ITU-T H.627

## Signalling and protocols for visual surveillance


### Summary

Recommendation ITU-T H.627 describes the detailed specifications of reference points, signalling and control methods, and overall protocols of a visual surveillance system, based on the requirements described in Recommendation ITU-T F.743 and the functional architecture described in Recommendation ITU-T H.626.

For the reference points between different functional entities relevant protocols are identified, such as protocols between the premises unit (PU) and service platform, the customer unit (CU) and service platform, and the PU and CU.

For reference points in the same functional entity, the internal signalling is described, including the signalling flow or information elements relevant for each reference point, such as the signalling for the service control flow, and the media distribution flow in the service platform.

Message formats and message text which relate to existing protocols (such as SIP, RTP, RTSP, HTTP) used in this Recommendation, are also provided.



### History

| Edition | Recommendation | Approval | Study Group |
|---|---|---|---|
| 1.0 | ITU-T H.627 | 2012-06-29 | 16 |

### Keywords

Control protocols, signalling, visual surveillance.

## FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

## NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

## INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

## Table of Contents

# Recommendation ITU-T H.627

## Signalling and protocols for visual surveillance

### 1 Scope

As defined in [ITU-T H.626], the visual surveillance service is a telecommunication service focusing on video (but also including audio) application technology, which is used to remotely capture multimedia (such as audio, video, image and alarm signals) and present them to the end user in a user-friendly manner, based on a managed broadband network with ensured quality, security and reliability.

The objective of this document is to describe the detailed specifications of reference points, signalling and control methods, and overall protocols of a visual surveillance system based on the functional architecture defined in [ITU-T H.626].

### 2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T F.743] Recommendation ITU-T F.743 (2009), *Requirements and service description for visual surveillance.*

[ITU-T H.626] Recommendation ITU-T H.626 (2011), *Architectural requirements for visual surveillance*.

[ITU-T Y.101] Recommendation ITU-T Y.101 (2000), *Global Information Infrastructure terminology: Terms and definitions.*

[ITU-T Y.2012] Recommendation ITU-T Y.2012 (2010), *Functional requirements and architecture of next generation networks.*

[IETF RFC 2326] IETF RFC 2326 (1998), *Real Time Streaming Protocol (RTSP)*.

[IETF RFC 2616] IETF RFC 2616 (1999), Hypertext *Transfer Protocol – HTTP/1.1*.

[IETF RFC 3261] IETF RFC 3261 (2002), *SIP: Session Initiation Protocol*.

[IETF RFC 3550] IETF RFC 3550 (2003), *RTP: A Transport Protocol for Real-Time Applications*.

### 3 Definitions

#### 3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

**3.1.1 interface** [ITU-T Y.101]: A shared boundary between two functional units.

NOTE 1 – An interface is defined by various characteristics pertaining to the functions, physical interconnections, signal exchanges and other characteristics, as appropriate.

**3.1.2 reference point** [ITU-T Y.2012]: A conceptual point at the conjunction of two non-overlapping functional entities that can be used to identify the type of information passing between these functional entities.

NOTE 2 – A reference point may correspond to one or more physical interfaces between pieces of equipment.

### 3.2 Terms defined in this Recommendation

None.

## 4 Abbreviations

This Recommendation uses the following abbreviations:

CMU Centre Management Unit

CPU Central Processing Unit

CU Customer Unit

FECC Far-End Camera Control

HTTP Hypertext Transfer Protocol

ID Identity

IIS Instant Image Snapshot

MDU Media Distribution Unit

MSU Media Storage Unit

PTZ Pan/Tilt/Zoom

PU Premises Unit

RTCP Real-time Transport Control Protocol

RTP Real-time Transport Protocol

RTSP Real-time Streaming Protocol

SCU Service Control Unit

SDP Session Description Protocol

SIP Session Initiation Protocol

SOAP Simple Object Access Protocol

VS Visual Surveillance

XML Extensible Markup Language

## 5 Introduction to VS functional architecture and related reference points

[ITU-T H.626] describes the functional architecture framework of visual surveillance (VS), illustrated in Figure 5-1, and introduces the reference points of VS architecture.

The present Recommendation provides a more detailed description of these reference points and introduces some new reference points.

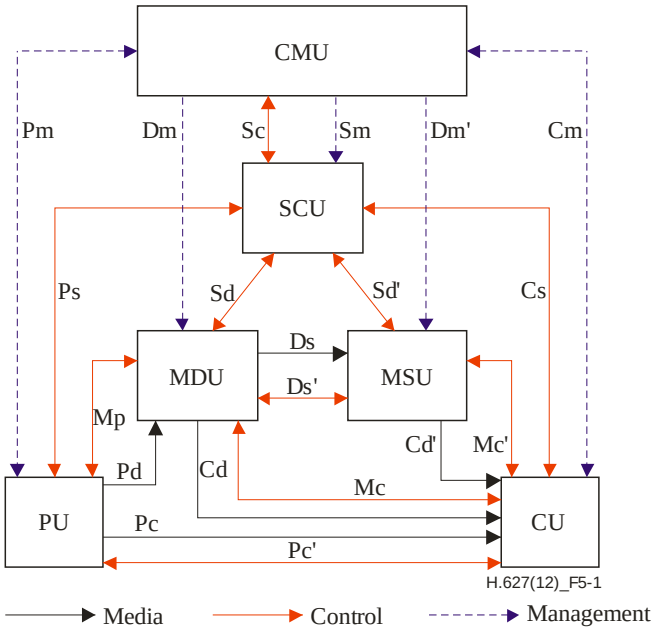


**Figure 5-1 – Decomposition of VS**

## 6 Signalling flows and relevant protocol exchanges

The following are illustrative items:

– the SIP message format should comply with the definition in [IETF RFC 3261]

– the RTSP message format should comply with the definition in [IETF RFC 2326]

– the HTTP message format should comply with HTTP1.1 defined in [IETF RFC 2616]

– the RTP/RTCP signalling and message refer to [IETF RFC 3550].

### 6.1 Service management

#### 6.1.1 PU registration

PU registration uses the SIP protocol [IETF RFC 3261]. The request message-header uses the standard REGISTER method, and the CMU returns a response with a "401 unauthorized" or "200 OK" message. The related signalling flows are shown in Figure 6-1 and the steps are described underneath.

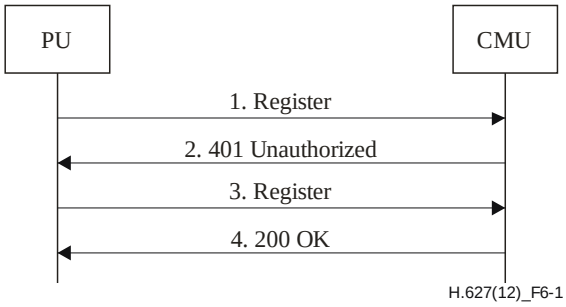


**Figure 6-1 – PU registration**

| | |
|---|---|
| Step 1: | The PU uses the SIP REGISTER method to implement registration to the CMU, the message header (SIP header) contains "From, To, Contract" fields to show the addresses of the PU and CMU. |
| Step 2: | The CMU returns a response with an SIP 401 message and requests the PU to send the authentication information. |
| Step 3: | The PU adds the authentication information and re-sends an SIP REGISTER message to the CMU. |
| Step 4: | The CMU returns an SIP 200 OK message or other error code to finish the registration. |

### 6.1.2 CU registration

CU registration has the same procedural flow and uses the same protocols as PU registration, see Figure 6-2.

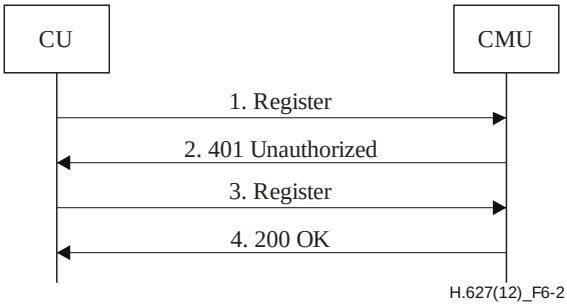


**Figure 6-2 – CU registration**

### 6.1.3 Obtaining the PU list

As illustrated in Figure 6-3, after the CU and PU have been registered to the service platform, the CU needs to request from the SCU in the service platform, the PU list information, in order to provide that list to the end users. The PU list information includes user information, static list information and dynamic list information of PUs. The following steps describe the process.

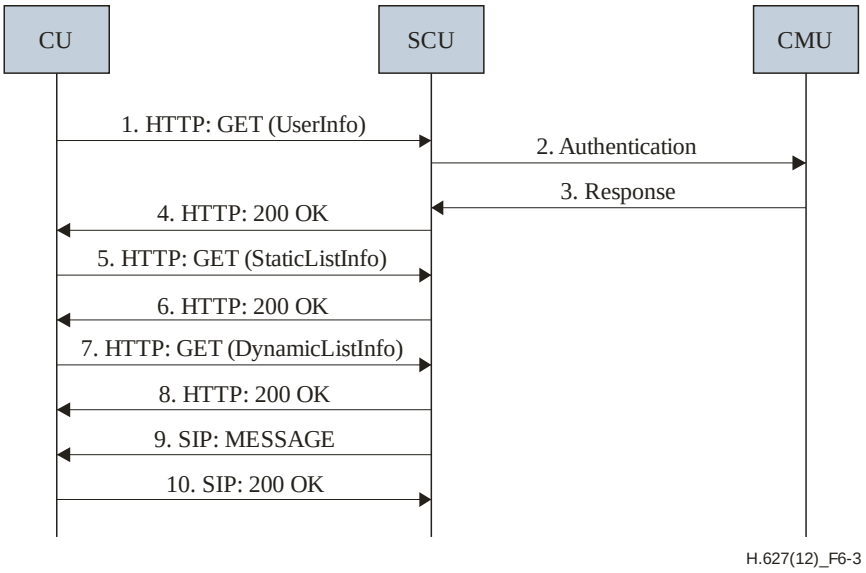


**Figure 6-3 – Obtaining the PU list**

| | |
|---|---|
| Step 1: | The CU sends an HTTP GET message with the Request-URI "UserInfo" to the SCU to request user information. |
| Steps 2, 3: | The SCU needs to check the authentication information from the CMU. The CMU returns a response to the SCU. |
| Step 4: | The SCU returns an HTTP 200 OK message with the user information to the CU. |
| Steps 5, 6: | The CU sends an HTTP GET message with the Request-URI "StaticListInfo" to the SCU to request the static PU list information. The SCU returns an HTTP 200 OK message with the static PU list information to the CU. |
| Steps 7, 8: | The CU sends an HTTP GET message with the Request-URI "DynamicListInfo" to the SCU to request the dynamic PU list information. The SCU needs to get the latest status of the PUs and returns an HTTP 200 OK message as a response to the CU. |
| Steps 9, 10: | The SCU sends an SIP MESSAGE with the updated PU list information to the CU when the SCU gets the relevant status changes event. |

### 6.1.4 PU-related parameters query

The related parameters of PUs include:

1) Serial port parameters, including serial port numbers, baud rate, data bit rate, parity bit, flow control parameters, PTZ decoder type, PTZ decoder address, etc.
2) Input parameters for switching, including channel number, channel state for input and output, etc.
3) Clock parameter in the PU, used by the PU for clock drift correction.
4) Video parameters, including video channel ID, encoding mode, encoding type, picture quality level, bit rate, frame rate, video resolution, I-frame interval, etc.
5) Video display parameters, including contrast ratio, brightness, chromaticity, saturation.
6) Subtitles parameters, including position of text (bitmap X/Y), text content, position of time stamp, etc.
7) Alarm configuration parameters, including alarm type, alarm level, parameters for instant image snapshot, etc.
8) PTZ control parameters, including PTZ ID, preset point index, preset point name, track number, PTZ velocity, etc.
9) PU status parameters, including CPU utilization, channel status, alarm status, etc.
10) PU logs parameters, including type of log, query method, query time, etc.

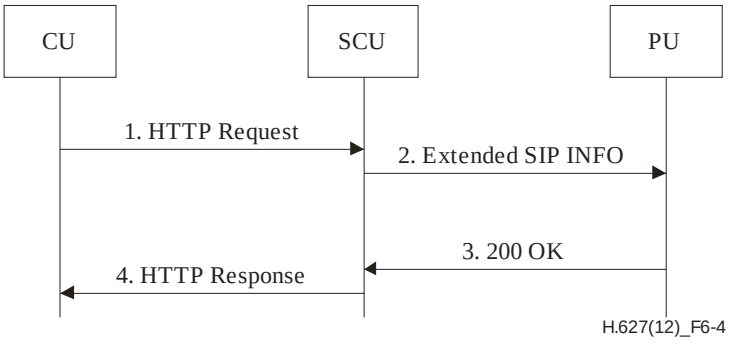


**Figure 6-4 – Procedural flows for a PU-related parameters query**

The procedural flows for a PU-related parameters query are illustrated in Figure 6-4 and described as follows:

| | |
|---|---|
| Step 1: | The CU sends an HTTP Request message to the SCU to request PU-related parameters. |
| Step 2: | The SCU sends an SIP INFO message to the specific PU to query its parameters. The SIP INFO message body uses the XML format to describe specific parameters. |
| Step 3: | The PU returns a 200 OK message to the SCU. |
| Step 4: | The SCU returns an HTTP Response message to the CU with the specific parameters requested by the CU. |

## 6.2 Alarm and linkage action

The alarm and linkage action function is one of the basic capabilities of VS, and it is used to respond to an input alarm signal event, give an output alarm activity and trigger the related actions. When the alarm event occurs, the VS system utilizes sounds, images, or other ways to inform end users of it.

The main procedural flows are indicated in Figure 6-5 and described in the steps below.

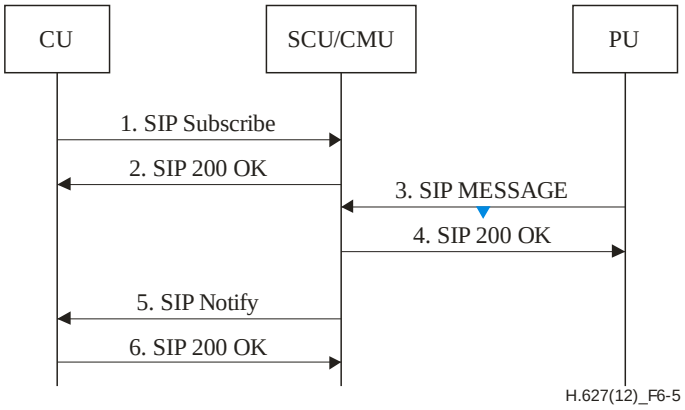


**Figure 6-5 – Procedural flows for alarm and linkage action**

| | |
|---|---|
| Step 1: | The CU uses the SIP Subscribe method to subscribe an alarm service to the SCU. |
| Step 2: | The SCU returns a response with an SIP 200 OK message to the CU. |
| Step 3: | When the PU captures an alarm event it uses the SIP Message method to notify the SCU of this event. |
| Step 4: | The SCU returns a response with an SIP 200 OK message to the PU. |
| Step 5: | The SCU uses the SIP Notify method to inform the CU of the event. |
| Step 6: | The CU returns an SIP 200 OK message to the SCU. |

## 6.3 Video/audio acquisition

The session control uses the SIP protocol and the message body uses the SDP protocol or XML text. The media transmission uses the RTP protocol. The ports for media channels are not defined in this Recommendation.

NOTE – The ports are usually pairs, such as port 7000 is used for RTP and 7001 is then used for RTCP.

### 6.3.1 Media acquisition transferred by the service platform

Figure 6-6 and the steps below illustrate media acquisition being transferred by the service platform.

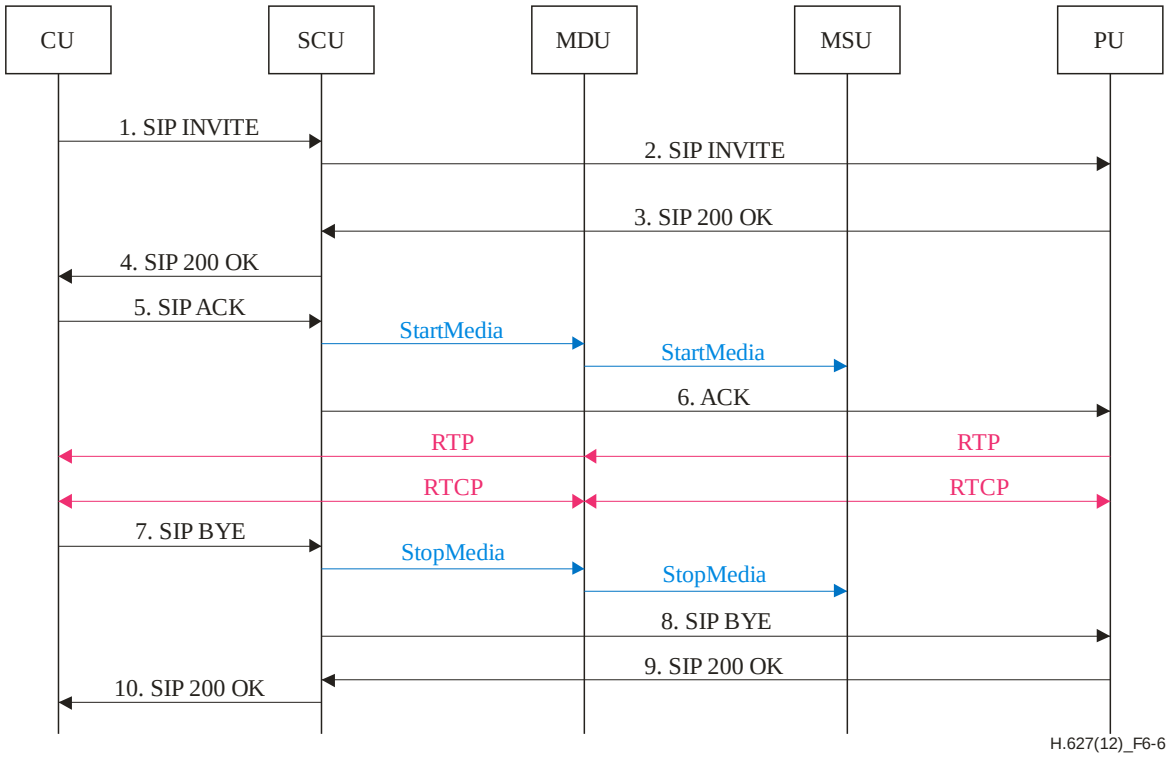


**Figure 6-6 – Media acquisition transferred by the MDU**

Steps 1, 2: The CU sends an INVITE message which includes the CU media format list and the CU address information to the SCU. The SCU transfers this message to the PU.

NOTE – In this step it can also handle the authentication.

Step 3: The PU returns a response with a 200 OK message which includes the media format of the selected PU and the PU address information.

Step 4: The SCU modifies the SDP information and then forwards a 200 OK message to the CU.

Step 5: The CU sends an ACK message including the SDP information to the SCU.

Step 6: The SCU notifies the MDU to start the media stream and the MDU notifies this event to the MSU. The SCU modifies the SDP information in ACK and forwards it to the PU. Media channels between the CU and MDU, MDU and PU are established. The PU sends RTP packages of media stream to the MDU, and the MDU transfers the media stream to the CU. RTCP packages are transported between the CU and MDU, and MDU and PU.

Steps 7, 8: The CU sends a BYE message to end the session. The SCU notifies the MDU to stop the media stream and the MDU notifies this event to the MSU. The MSU stops sending the media stream to the CU. Then the SCU transfers the BYE message to the PU.

Step 9: The PU stops sending the media stream and returns a response with a 200 OK message to the SCU.

Step 10: The SCU transfers the 200 OK message to the CU.

### 6.3.2 Media acquisition directly from the PU

Figure 6-7 and the steps below illustrate media acquisition directly from the PU.

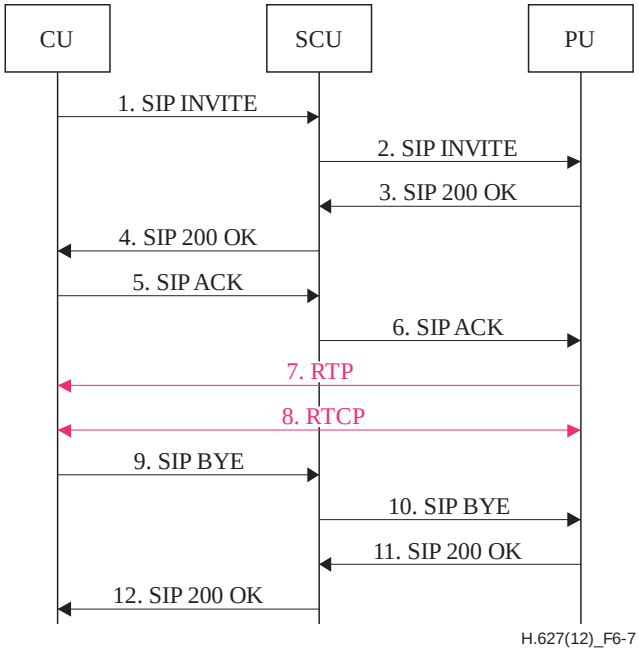


**Figure 6-7 – Media acquisition directly from the PU**

| | |
|---|---|
| Step 1: | The CU sends an INVITE message to the SCU with the CU media format list and CU address information. NOTE – In this step, it can also handle the authentication. |
| Step 2: | The SCU sends the INVITE message to the PU with the CU media format list. |
| Step 3: | The PU handles the INVITE message and returns a 200 OK message including the SDP information to choose the codec and PU address information. |
| Step 4: | After receiving the 200 OK message, the SCU selects the codec type, modifies the SDP and then forwards a 200 OK message to the CU. |
| Step 5: | The CU returns a response to the SCU with an ACK message including an SDP answer. |
| Step 6: | The SCU modifies the SDP in ACK and sends it to the PU. |
| Steps 7, 8: | The PU sends RTP packages of the media stream to the CU. RTCP packages are transported between the PU and CU. |
| Steps 9, 10: | The CU sends a BYE message to end the session and the SCU transfers the message to the PU. |
| Step 11: | The PU stops sending the media stream and returns an 200 OK message to the SCU. |
| Step 12: | The SCU returns the 200 OK message to the CU and stops sending the media stream. |

## 6.4 Instant image snapshot

The instant image snapshot (IIS) function is used to get instant image information based on video from the surveilled object.

The procedural flow in Figure 6-8 involves the implementation of the snapshot action in the PU which then uploads the snapshot image to the service platform from which the CU can access this image. The steps below describe the process.

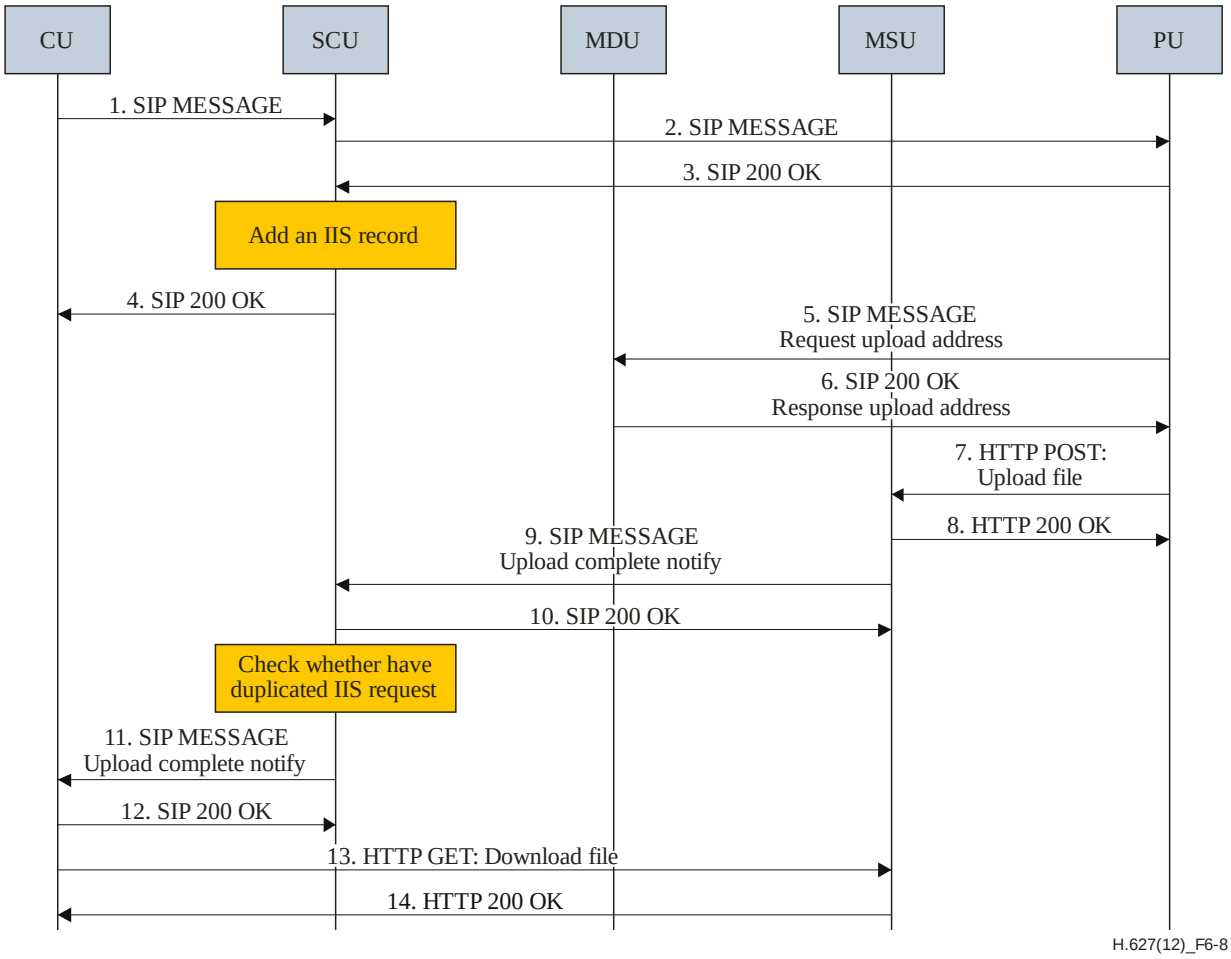


**Figure 6-8 – Procedural flows for the instant image snapshot**

| | |
|---|---|
| Steps 1, 2, 3, 4 | The CU sends an SIP MESSAGE to the SCU to request an instant image snapshot. The SCU transfers this message to the requested PU. The PU handles this request and returns an SIP 200 OK message to the SCU. The SCU adds an IIS (instant image snapshot) record in the service platform and returns an SIP 200 OK message to the CU. |
| Steps 5, 6: | The PU sends an SIP MESSAGE to the MDU to request the image upload address. The MDU handles this request and returns an SIP 200 OK message with the upload address to the PU. |
| Steps 7, 8: | The PU uploads the image file to the MSU through the HTTP POST method with the URL of the upload address, and the MSU returns an HTTP 200 OK message. |
| Steps 9, 10: | The MSU sends an SIP MESSAGE to notify the SCU of the completion of the image upload. The SCU then returns an SIP 200 OK message to the MSU. |
| Steps 11, 12: | The SCU sends an SIP MESSAGE to notify the CU for the completion of the image upload. The CU then returns an SIP 200 OK message to the SCU. |
| Steps 13, 14: | The CU uses the HTTP GET method to request the MSU to download the snapshot image. The MSU handles this request and returns an HTTP 200 OK message with the image. |

## 6.5 Recording and playback

### 6.5.1 Recording

The recording function is used to record multimedia (such as audio and/or video) information from a surveilled object and store this multimedia information either in PUs or in the service platform.

Figure 6-9 shows the procedural flows for recording in the PU. It uses the SIP MESSAGE method and the XML format to specify parameters. The steps below describe the process.

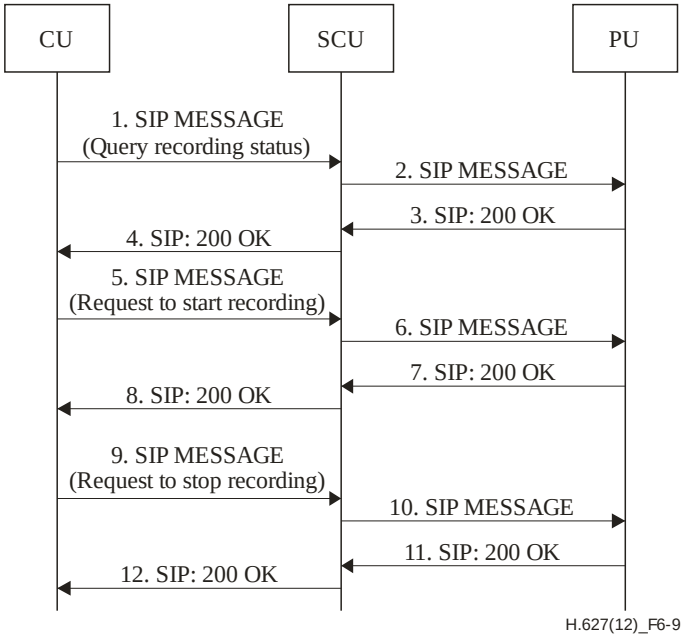


**Figure 6-9 – Procedural flows for recording in the PU**

| | |
|---|---|
| Steps 1,2: | The CU sends an SIP MESSAGE to the SCU to query the recording status of the requested PU. After authentication, the SCU transfers this message to the requested PU. |
| Steps 3, 4: | The PU returns an SIP 200 OK message to the SCU with the recording status of the requested PU. The SCU handles this response and returns an SIP 200 OK message to the CU. |
| Steps 5, 6: | The CU sends an SIP MESSAGE to the SCU to request to start recording in the PU with the parameters of recording conditions. The SCU handles this request to start a new recording task or change the existing recording task according to the new request conditions. Then the SCU sends an SIP MESSAGE to the PU to request the PU to start a recording task. |
| Steps 7, 8: | The PU returns an SIP 200 OK message to the SCU and the SCU transfers an SIP 200 OK message to the CU. |
| Steps 9, 10: | The CU sends an SIP MESSAGE to the SCU to request recording is stopped in the PU. The SCU handles this request to stop the recording task. Then the SCU sends an SIP MESSAGE to the PU to stop recording. |
| Steps 11, 12: | The PU returns an SIP 200 OK message to the SCU and the SCU transfers an SIP 200 OK message to the CU. |

Figure 6-10 shows the procedural flows of recording in the service platform (media stored in the MSU). It uses the HTTP protocol between the CU and SCU and the XML format to specify parameters. It uses the RTSP protocol between the PU and MDU to perform video or audio recording. It uses the SOAP protocol among units in the service platform, such as the SCU, CMU, MDU and MSU. The steps below describe the process.

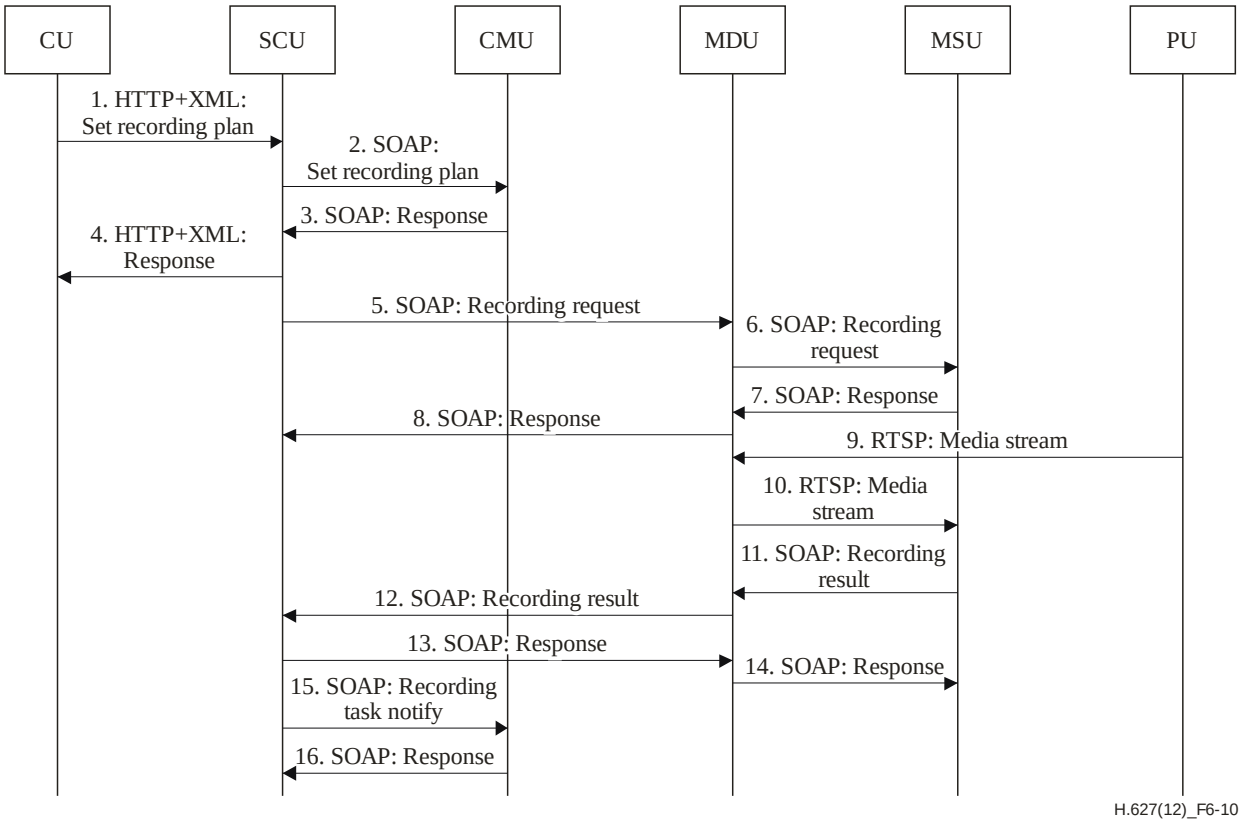


**Figure 6-10 – Procedural flows for recording in the service platform**

| | |
|---|---|
| Step 1: | The CU sets a recording plan in the service platform and sends an HTTP+XML message to the SCU. |
| Steps 2, 3, 4: | The SCU transfers the plan using a SOAP message to the CMU. The CMU saves and confirms the plan, then uses a SOAP message to return a response to the SCU. The SCU uses an HTTP+XML message to return this response to the CU. |
| Step 5: | The SCU gets a recording event and uses a SOAP message to send a recording request to the MDU. |
| Steps 6, 7, 8: | The MDU uses a SOAP message to transfer this request to the MSU, and the MSU uses a SOAP message to return the response to the MDU. Then the MDU returns the response to the SCU. |
| Steps 9,10: | Then the media stream is sent from the PU to the MDU. After some pre-processing (such as transcoding, creation of suitable file format, etc.), the MDU transfers the media to the MSU for storage and management using the RTSP protocol. |
| Steps 11 to 14: | When the recording task has ended, the MSU notifies the MDU of the recording result using a SOAP message and the MDU transfers this notification to the SCU. After the SCU receives the notification, it returns the response to the MDU, and the MDU transfers the response to the MSU. |
| Steps 15, 16: | The SCU uses a SOAP message to notify the CMU that the task has ended and the CMU saves the related logs. Then the CMU returns a response to the SCU to finish the whole recording procedure. |

### 6.5.2 Playback

The playback function is used to play a media resource which has been previously recorded and stored.

According to the location where the recorded media is stored, there are three different storage modes: storing it in the service platform, the PU and the CU.

Figure 6-11 shows the procedural flows for playback directly from the PU. The steps below describe the process.

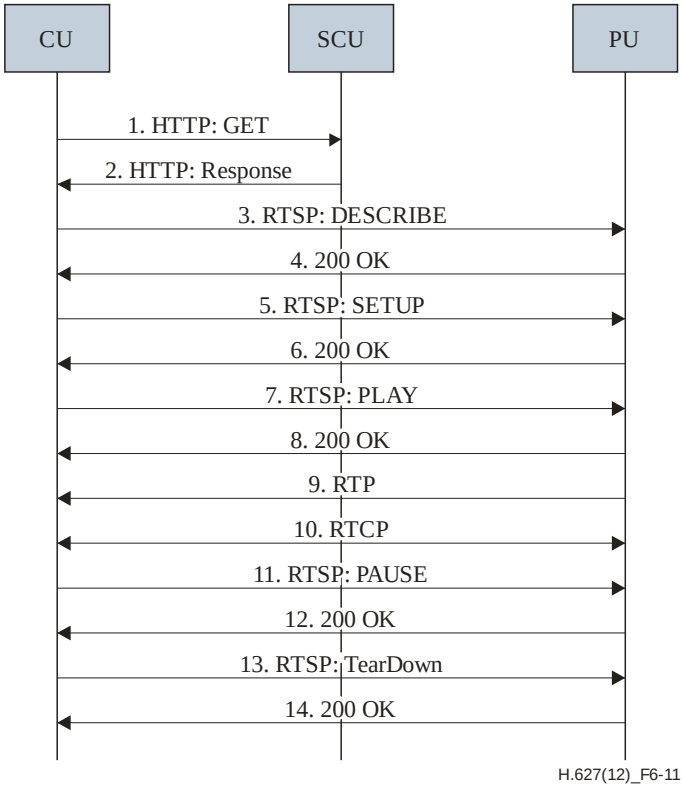


**Figure 6-11 – Procedural flows for playback directly from the PU**

Step 1: The CU sends an HTTP GET request to the SCU to get the URL of the media resource to be played.

Step 2: The SCU in the service platform returns the URL to the CU with an HTTP response.

Step 3: The CU sends an RTSP DESCRIBE request to the PU to get the session description information of the PU.

Step 4: The PU returns a response with an RTSP 200 OK message, including the SDP information.

Step 5: The CU initiates an RTSP SETUP request to the PU, which at the least includes the following information:

– URL of the requested media resource
– request sequence number
– transport parameters for the CU client.

Step 6: The PU returns a response with an RTSP 200 OK message, including the following information:

status line: RTSP/1.0 200OK

sequence number

session ID

transport parameters for the CU client.

Step 7: The CU sends an "RTSP PLAY" request to start getting the media packets, and this request includes the following information:

– request line, including the URL of the requested media resource
– request sequence number
– session ID
– range (this parameter is optional).

NOTE – The RTSP server (here it is the PU) is recommended to support fast forward/fast rewind and random access functions through the RTSP PLAY request with the standard parameters defined in [ITU-T RFC 2326].

To implement the forward/fast rewind function, "Scale" and/or "Speed" parameters are used.

A scale value of 1 indicates normal play or record at the normal forward viewing rate. If not 1, the value corresponds to the rate with respect to the normal viewing rate. An example of playing in reverse at 4 times the normal rate:

Scale: –4

The parameter value is expressed as a decimal ratio, e.g., a value of 2.0 indicates that data is to be delivered twice as fast as normal. Example:

Speed: 2.0

To implement the random access function: use the "Range" parameter to indicate the starting point requested by the CU. Example:

Range: clock=19960213T143205Z-;time=19970123T143720Z

Step 8: The PU returns a response with an RTSP 200 OK message, including the following information:

status line: RTSP/1.0 200 OK,

sequence number,

session ID,

range,

RTP-Info: URL of the media resource, first RTP package sequence number, RTP timestamp.

Step 9: The PU sends RTP packages of the media to the CU.

Step 10: RTCP packages are sent between the PU and CU.

Step 11: If the media stream needs to be paused, the CU sends an RTSP PAUSE request to the PU, and this request includes the following information:

request line, including the URL of the requested media resource

request sequence number

session ID

range (optional).

Step 12: The PU returns a response with an RTSP 200 OK message, including the following information:

status line: RTSP/1.0 200 OK

sequence number.

Step 13: If the media stream needs to be stopped, the CU sends an RTSP TEARDOWN request, and this request includes the following information:

request line, including the URL of the requested media resource

request sequence number

session ID.

Step 14: The PU returns a response with an RTSP 200 OK message, including the following information:

status line: RTSP/1.0 200 OK

sequence number.

Figure 6-12 shows the procedural flows for playback from the MSU in the service platform. The steps below describe the process.

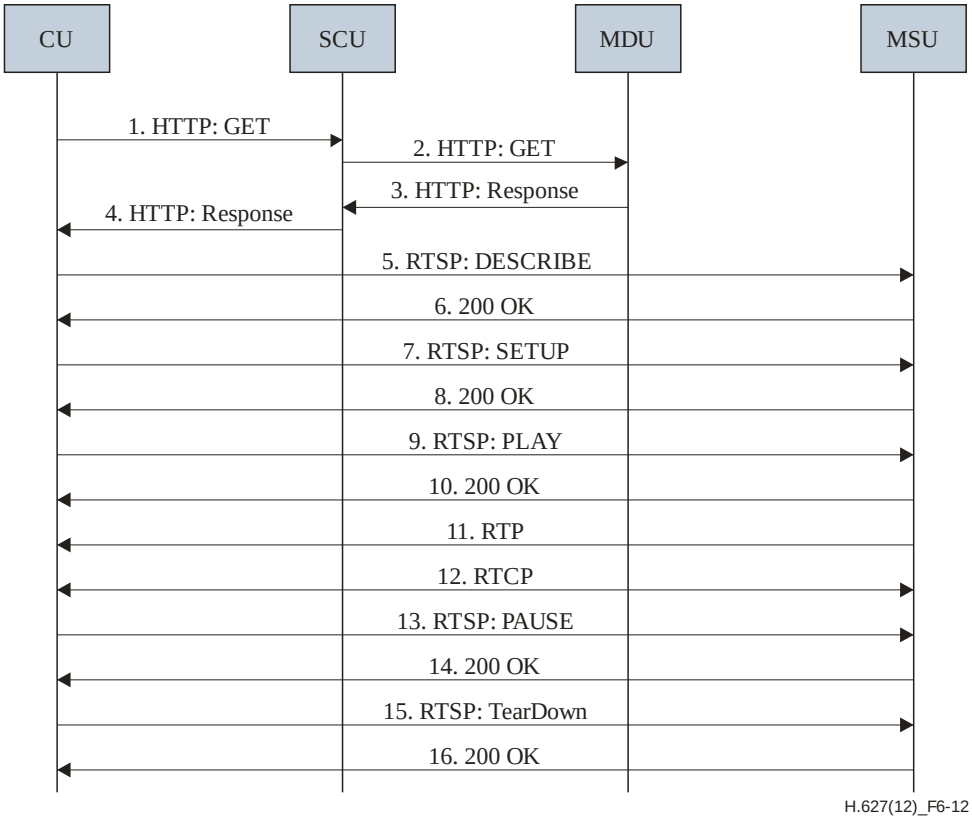


**Figure 6-12 – Procedural flows for playback from the service platform**

| | |
|---|---|
| Step 1: | The CU sends an HTTP GET request to the SCU to get the URL of the media resource to be played. |
| Step 2: | The SCU transfers this request to the MDU. |
| Step 3: | The MDU in the service platform returns the URL to the SCU with an HTTP response. |
| Step 4: | The SCU transfers this response to the CU. |
| Steps 5 to 16: | For steps 5 to 16, signalling and messages are the same as for steps 3 to 14 in the procedural flows for playback directly from the PU, but the RTSP media server for the CU is the MSU more than the PU, and the media server is recommended to support fast forward/fast rewind and random access functions. |

## 6.6 Voice (audio) communication

### 6.6.1 Voice communication

Voice communication is one of the advanced capabilities of visual surveillance and it is used for communication between the end user and the surveilled object (when the surveilled object is a human being). Figure 6-13 describes the procedural flows for voice communication and the steps below describe the process.

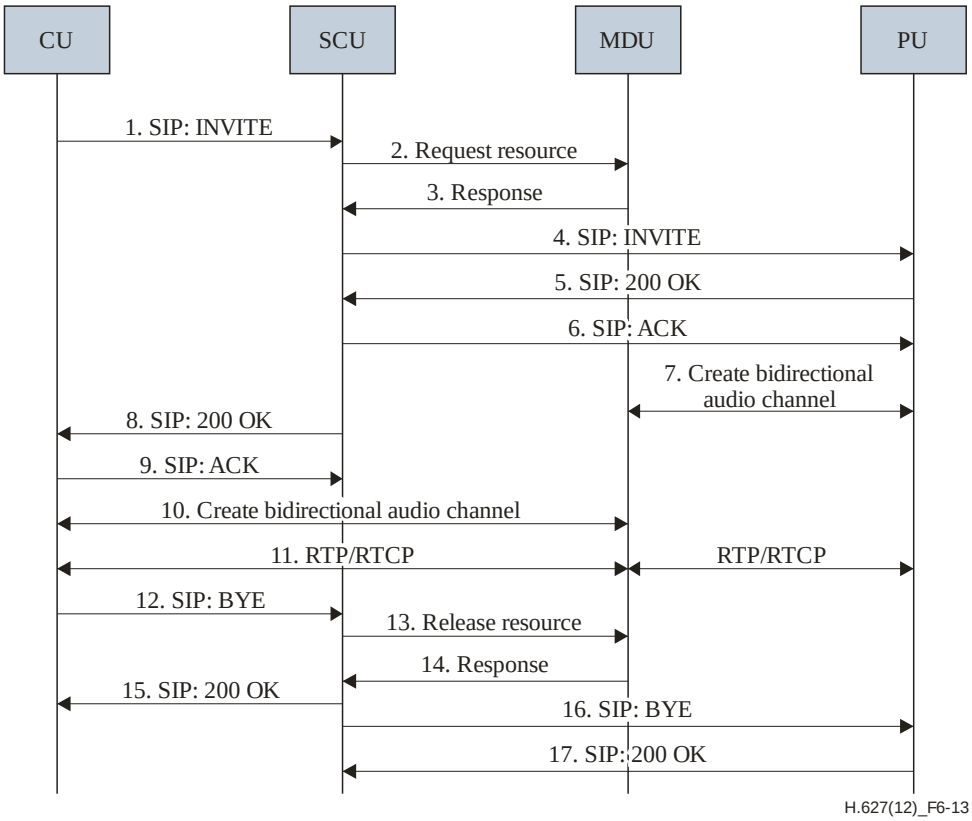


**Figure 6-13 – Procedural flows for voice communication**

Step 1: The CU sends an SIP INVITE message with the SDP information of the CU to the SCU to request voice communication.

Steps 2, 3: The SCU needs to apply resources from the MDU for media stream forwarding.
NOTE – This is an internal interface in the service platform. The MDU returns a response to the SCU with the SDP information of the MDU. This may be implemented based, for example, on the TCP protocol, but no protocol is recommended here.

Steps 4, 5: The SCU sends an SIP INVITE message with the SDP information of the MDU to the PU to request voice communication. The PU returns an SIP 200 OK message with the SDP information of the PU.

Step 6: The SCU sends an SIP ACK message to the PU to confirm this session.

Step 7: Bidirectional audio channels are created between the PU and MDU.

Step 8: The SCU returns an SIP 200 OK message to the CU with the SDP information of the MDU.

Step 9: The CU sends an SIP ACK message to the SCU to confirm this session.

Step 10: Bidirectional audio channels are created between the CU and MDU.

Step 11: RTP packages of audio media stream are transferred by the MDU between the CU and PU.

Step 12: To end the voice communication, the CU sends an SIP BYE message to the SCU.

Steps 13, 14: The SCU notifies the MDU to stop transferring the media stream and release resources. The MDU returns a response to the SCU.
NOTE – This is also an internal interface in the service platform. No protocol is recommended here.

Step 15: The SCU returns an SIP 200 OK message to the CU to end the communication between the CU and MDU.

Steps 16, 17: The SCU sends an SIP BYE message to the PU which returns an SIP 200 OK message to the SCU to end the communication between the PU and MDU.

### 6.6.2 Voice/audio broadcast

The voice/audio broadcast function is one of the advanced capabilities of a visual surveillance system, and it is used to broadcast audio stream only from the CU to multiple surveilled objects (human beings) through the PUs.

The procedural flow in Figure 6-14 illustrates the implementation of the voice/audio broadcast action. The steps below describe the process.

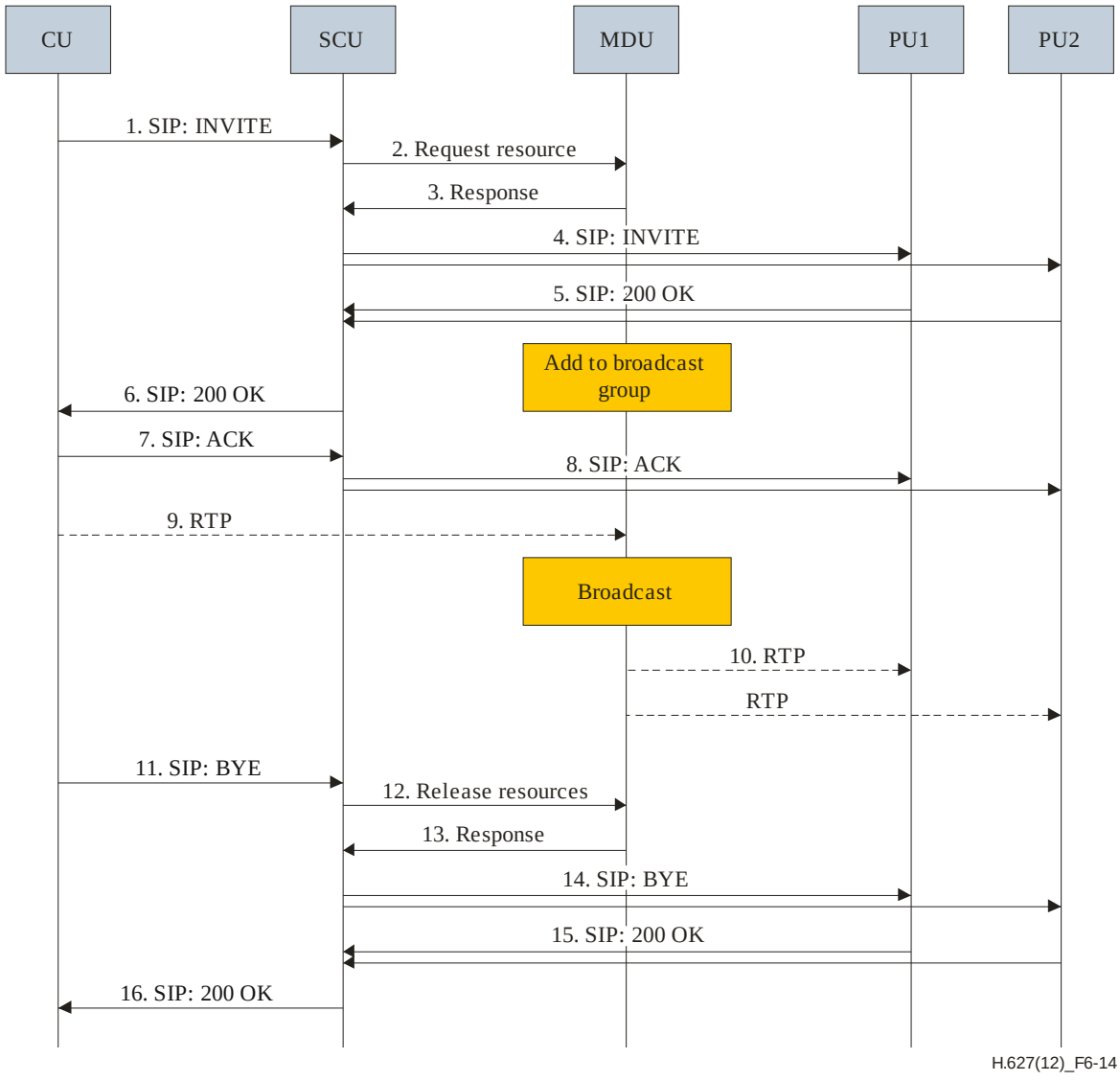


**Figure 6-14 – High level procedural flows for voice/audio broadcast**

| | |
|---|---|
| Step 1 | The CU sends an SIP INVITE to the SCU to request voice/audio broadcast action with the information of the requested PU list. |
| Steps 2, 3: | After authentication, the SCU requests the MDU to assign resources for broadcast action, and the MDU returns a response with the result of resources reservation to the SCU. |
| Steps 4, 5: | The SCU sends an SIP INVITE to each PU of the requested list for broadcast connection.<br>Each PU creates a unidirectional media channel with the MDU and returns a response separately with an SIP 200 OK message to the MDU. |
| Steps 6, 7: | The SCU returns a response with an SIP 200 OK message to the CU. The CU creates a unidirectional media channel with the MDU and sends an SIP ACK message to the SCU. |
| Step 8: | The SCU returns a response with an SIP 200 OK message to each PU separately. |
| Step 9: | The CU sends the media stream to the MDU through the created media channel. |
| Step 10: | The MDU broadcasts the media stream to each PU through separate media channels. |
| Step 11: | To stop broadcasting, the CU sends an SIP BYE message to the SCU and releases the media channel with the MDU. |
| Steps 12, 13: | The SCU requests the MDU to stop broadcasting and release resources, and the MDU returns a response to the SCU. |
| Steps 14, 15: | The SCU sends an SIP BYE message to each PU separately. Each PU stops receiving the media stream and releases the unidirectional media channel with the MDU, and then sends an SIP 200 OK message to the SCU. |
| Step 16: | The SCU sends an SIP 200 OK message to the CU to complete the whole procedure. |

## 6.7 Far-end camera control

The far-end camera control (FECC) function is used to pan, tilt and zoom cameras, switch auxiliary peripherals (such as flashlight, fan and rain brush), and to preset the position of related physical devices.

Figure 6-15 shows the procedural flows and the steps below describe the process.

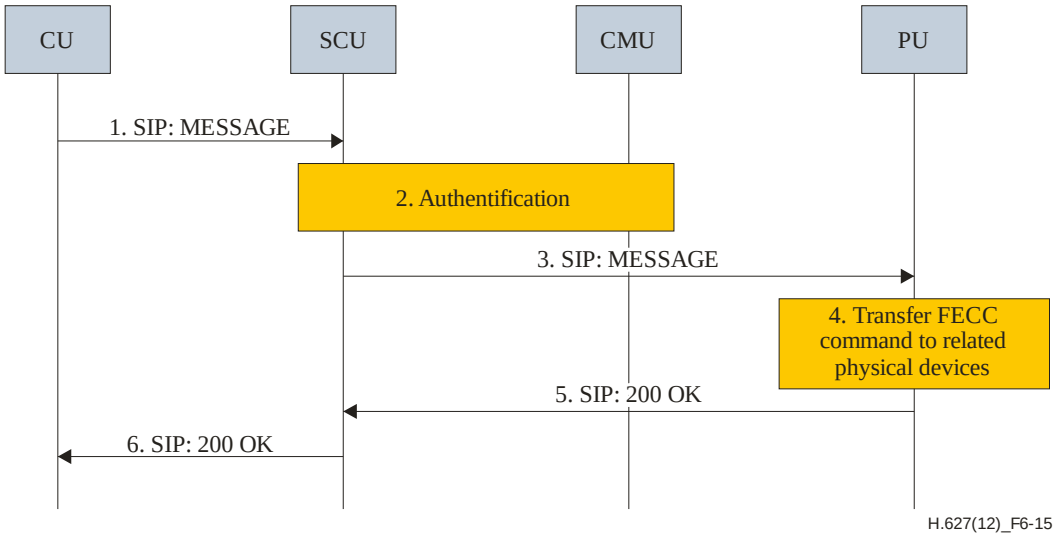


**Figure 6-15 – Procedural flows for the FECC function**

Step 1: The CU sends an SIP MESSAGE to the SCU with information of the far-end camera control commands in order to request remote control of the requested camera or auxiliary peripherals (PU).

Step 2: The SCU handles authentication with some user information from the CMU.

Step 3: After authentication, the SCU transfers the SIP MESSAGE to the requested PU.

Step 4: The PU parses these control commands and sends them to the related physical devices in order to implement the far-end camera control operations.

Step 5: The PU returns a response to the SCU with an SIP 200 OK message.

Step 6: The SCU transfers this SIP 200 OK message to the CU to complete this operation.

## SERIES OF ITU-T RECOMMENDATIONS

| | |
|---|---|
| Series A | Organization of the work of ITU-T |
| Series D | General tariff principles |
| Series E | Overall network operation, telephone service, service operation and human factors |
| Series F | Non-telephone telecommunication services |
| Series G | Transmission systems and media, digital systems and networks |
| **Series H** | **Audiovisual and multimedia systems** |
| Series I | Integrated services digital network |
| Series J | Cable networks and transmission of television, sound programme and other multimedia signals |
| Series K | Protection against interference |
| Series L | Construction, installation and protection of cables and other elements of outside plant |
| Series M | Telecommunication management, including TMN and network maintenance |
| Series N | Maintenance: international sound programme and television transmission circuits |
| Series O | Specifications of measuring equipment |
| Series P | Terminals and subjective and objective assessment methods |
| Series Q | Switching and signalling |
| Series R | Telegraph transmission |
| Series S | Telegraph services terminal equipment |
| Series T | Terminals for telematic services |
| Series U | Telegraph switching |
| Series V | Data communication over the telephone network |
| Series X | Data networks, open system communications and security |
| Series Y | Global information infrastructure, Internet protocol aspects and next-generation networks |
| Series Z | Languages and general software aspects for telecommunication systems |

Printed in Switzerland
Geneva, 2013