Interface Requirements for HIKARI Services

Ver. 1.2

March 26, 2002

HIKARI Service Architecture Consortium
Technical Committee
## Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date of Revision</th>
<th>Modifications</th>
<th>Reasons for Modifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Jan. 22, 2002</td>
<td>–</td>
<td>First version</td>
</tr>
<tr>
<td>1.1</td>
<td>Jan. 31, 2002</td>
<td>• Figs. 4.2.1 to 4.2.4</td>
<td>Figs. 4.2.1 to 4.2.4 have the form of another drawing.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Correction of misprints</td>
<td></td>
</tr>
<tr>
<td>1.2</td>
<td>Mar. 26, 2002</td>
<td>• Review of a partial description</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Addition of an appendix</td>
<td></td>
</tr>
</tbody>
</table>
Authors (in alphabetical order)

IZUMI Hiroyuki  NTT Software Corporation
HASEGAWA Satoshi  NEC Corporation
HASHIMOTO Junichi  Nippon Telegraph and Telephone East Corporation
MIZUHARA Noboru  Hitachi, Ltd.
MORINISHI Yuji  Nippon Telegraph and Telephone Corporation
MORITA Hideyo  Nippon Telegraph and Telephone Corporation
NAKANO Go  Matsushita Electric Industrial Co., Ltd.
NIKI Teruki  Matsushita Electric Industrial Co., Ltd.
TAKAYAMA Katsunori  Fujitsu Limited
Contents

1. Introduction ................................................................................................................................. 1

2. Outline of This Document ............................................................................................................ 3
   2.1 Configuration of This Document ................................................................................................. 3
   2.2 How to Read This Document ..................................................................................................... 4
       2.2.1 How to see diagram .............................................................................................................. 4
       2.2.2 How to see table ................................................................................................................... 4
       2.2.3 Applicable target protocol .................................................................................................. 5

3. Protocols of Interface Requirements on Broadband Net-live Service ........................................ 7
   3.1 Broadband Net-live Service ..................................................................................................... 7
       3.1.1 Outline of service ................................................................................................................ 7
       3.1.2 Outline of service function and service sequence ................................................................. 8
       3.1.3 Function and issue required for realizing service ................................................................. 10
   3.2 Interface in Broadband Net-live Service .................................................................................... 16
       3.2.1 Content attribute registration function ................................................................................ 16
           3.2.1.1 Outline of function ...................................................................................................... 16
           3.2.1.2 Content attribute registration function sequence ......................................................... 16
           3.2.1.3 Service sequence and applicable protocol set ............................................................... 17
           3.2.1.4 Applicable condition ..................................................................................................... 17
       3.2.2 Ticket purchase function .................................................................................................... 18
           3.2.2.1 Outline of function ...................................................................................................... 18
           3.2.2.2 Ticket purchase function sequence diagram ................................................................. 18
           3.2.2.3 Service sequence and applicable protocol set ............................................................... 19
           3.2.2.4 Applicable condition ..................................................................................................... 19
       3.2.3 Live content feed/distribution function ............................................................................... 21
           3.2.3.1 Outline of function ...................................................................................................... 21
           3.2.3.2 Live content feed/distribution function sequence diagram ........................................... 21
           3.2.3.3 Service sequence and applicable protocol set ............................................................... 22
           3.2.3.4 Applicable condition ..................................................................................................... 22
       3.2.4 Live viewing function ......................................................................................................... 24
           3.2.4.1 Outline of function ...................................................................................................... 24
           3.2.4.2 Live viewing function sequence diagram ................................................................. 24
           3.2.4.3 Service sequence and applicable protocol set ............................................................... 26
           3.2.4.4 Applicable condition ..................................................................................................... 27
       3.2.5 Viewing end function ......................................................................................................... 29
           3.2.5.1 Outline of function ...................................................................................................... 29
           3.2.5.2 Viewing end function sequence diagram ................................................................. 29
           3.2.5.3 Service sequence and applicable protocol set ............................................................... 31
           3.2.5.4 Applicable condition ..................................................................................................... 31

4. Protocols of Interface Requirements on Bi-directional Communication Service ...................... 32
   4.1 Bi-directional Communication Service ..................................................................................... 32
       4.1.1 Outline of service ................................................................................................................ 32
       4.1.2 Service function and outline of service sequence ................................................................. 33
       4.1.3 Function and issue required for realizing service ................................................................. 36
   4.2 Interface in Bi-directional Communication Service .................................................................... 43
       4.2.1 Utilization application function .......................................................................................... 43
           4.2.1.1 Outline of function ...................................................................................................... 43
           4.2.1.2 Utilization application function sequence diagram ....................................................... 43
           4.2.1.3 Service sequence and applicable protocol set ............................................................... 44
           4.2.1.4 Applicable condition ..................................................................................................... 44
       4.2.2 Teaching materials registration function ............................................................................. 45
           4.2.2.1 Outline of function ...................................................................................................... 45
           4.2.2.2 Teaching materials registration function sequence diagram ....................................... 45
           4.2.2.3 Service sequence and applicable protocol set ............................................................... 46
           4.2.2.4 Applicable condition ..................................................................................................... 46
       4.2.3 Trainee recruitment function ............................................................................................... 47
1. Introduction

The HIKARI Service Architecture Consortium (HSAC) is aiming at realization of a video-rich HIKARI Service provided to end users on an ultra-high speed optical access network of about 10 Mbps to 100 Mbps. To attain this aim, the following three items become important.

1) Investigating video-rich HIKARI Services suitable for the age of broadband communication.
2) Making clear the functions and structure of the HIKARI Service platform for distributing/delivering the HIKARI Service from providers to users and constructing Service Network Architecture.
3) Making clear interface requirements between functional modules when the system construction for realizing the HIKARI Service.

Fig. 1.1 shows the basic structure (framework) of the HIKARI Service. In this consortium, the above three items are investigated and systemized as the following documents that correspond to the items respectively.

1) Service Specifications on HIKARI Service Network Architecture
3) Interface Requirements for HIKARI Services

In the Service Specifications on HIKARI Service Network Architecture, five killer HIKARI Services are selected and their outlines are described. At the same time, their business nature, sociability, and technical requirements are investigated. In the SA Technical Report, in addition to a study on the function, structure, and architecture of the above HIKARI Service platform, players for realizing the HIKARI Service are arranged and various technical issues are investigated. Further, the sequence is also analyzed as a case study by selecting two specific services (a net-live and a remote cooperative work) from the services described in the Service Specifications on HIKARI Service Network Architecture. Besides, in the SA Technical Report and this document, the above remote cooperative work is described as a bi-directional visual communication service and interface requirements are specified taking distance learning as an example.
This document makes clear the interfaces that exist between the HIKARI Service platform and user, between the HIKARI Service platform and provider, and exist in the inside of the HIKARI Service platform and specify requested interface requirements. And the interface requirements mean to provide recommended protocols and data formats for functional modules. To specify actual interface requirements, the two specific services of 1) a broadband net-live service and 2) a bi-directional visual communication service were assumed and the interface requirements specified according to the signal transfer on a service sequence were investigated. However, such detailed protocols and content formats that can be implemented are not specified. For the protocols, existing international standards and de-facto standards were adopted. However, when the current protocol or content format standard needs be extended, the extended conditions are shown, and the requisite conditions are shown in this document if there is no applicable protocol.

The purpose of the Interface Requirements for HIKARI Services is to be used as what is called a guideline for the system construction for realizing the HIKARI Service. To enable the practical system development, the expansion into documents, such as more detailed functional specifications, an interface basic specification, and an interface implementation specification, will become necessary assuming the Interface Requirements for HIKARI Services Architecture as the uppermost-based document.
2. Outline of This Document

2.1 Configuration of This Document

This document specifies interface requirements on two specific HIKARI Services.

(1) Broadband net-live service
(2) Bi-directional visual communication service

In Section 3, an outline of a broadband net-live service is described first. The broadband net-live service distributes/delivers a live event, such as sports and entertainment, to many end users. The multicast distribution/delivery function of 1:N is used as the basic function. Additional functions only for HIKARI Service, such as a “multi-angular function” and a “time-shifted function” in which user’s convenience was considered are also investigated. To realize a high quality video service with the same level as a DVD as a distribution video format, MPEG-2 of about 6 Mbps is considered. Following a description of a service outline, concerning major frame functions when a service is provided from the net-live service provider to users, content attribute registration and net-live feed/distribution functions are defined as the service provider side and a ticket purchase function, a live viewing function, and a viewing end function are defined as the user side. Subsequently, for these five individual functions, a description of a service sequence to be realized, a description of an applicable protocol and data on the service sequence, and a description of a condition requested when the protocol is applied are made.

In Section 4, an outline of the bi-directional visual communication service is described first by focusing the service on distance learning. The bi-directional visual communication service that the HIKARI Service provides aims at providing a service that enables a flexible stream configuration with presence making use of HIKARI broadband transmission. By acquiring a flexible N:M bi-directional stream configuration, face-to-face learning between an instructor and a trainee, group leaning, and information interchange between trainees are enabled. The conditions of the video format are high image quality (exceeding the degree of a VHS)/high sound quality, and a low delay (requisite conditions are less than 300 msec). Following a description of the service outline, nine major frame functions necessary for realizing the distance learning are defined. They are a service registration function, a posting teaching materials function, a class announcement function, a class attendance registration function, an opening class function, trainee’s intermediate participation/early leaving/speech function, a video display switching function, an inter-trainee conversation function (group learning function), and a closing class function. Subsequently, a description of a service sequence to be realized every function is made, a description of an applicable protocol and data is made on the service sequence, and a description of conditions requested for the application of the protocol is made.

In Appendix, taking RTSP into consideration, more practical specification is made. RTSP is one of the well-known protocol for specifying the streaming content control, and has been published as Internet Draft RFC 2326. However, RTSP specified in RFC 2326 cannot fully support interoperability between vendors due to a lack of clear protocol description. In this Appendix, a detailed study has been made from an interoperability point of view, which includes specification of RTSP message details, SDP description, sample sequences, etc.

Concerning the service sequence symbolic convention in Sections 3 and 4, the live content feed/distribution is shown in the next section as an example.
2.2 How to Read This Document

This section shows how to read the sequence diagrams and protocol set table shown in Sections 3 and 4 and technical terms that appear in the text.

2.2.1 How to see diagram

2.2.2 How to see table

The “upper layer” or “lower layer” standard is relative and does not match a layer model of OSI.
2.2.3 Applicable target protocol

The protocols used in this document are listed below.


This protocol automatically performs dynamic allocation of a reusable IP address and various settings in the IP network.

**Diffserv** (Differentiated Services)

Diffserv is a technology that identifies user’s traffic and provides the communication quality that corresponds to individual traffic.

In Diffserv, the ToS (Type of Service) field (8 bits) in the IP header is redefined as:
- 6-bit DSCP (Differentiated Services Code Point) field, and
- 2-bit field that is not used currently, and a band is guaranteed every this DSCP value.

Unlike IntServ, because control by flow is not performed, a router load is lightweight. Further, such signaling as RSVP is also necessary, Diffserv is suitable for the structure of the current Internet and its practical use is advancing. However, such a strict band that is equal to IntServ cannot be guaranteed.

**FTP** (File Transfer Protocol; 959, 1123, 1127, 1579, 1635, 2228, 2389, 2428, 2577, 2640)

This protocol mainly transfers files between a client and a server over the IP network.

**H.323**

This protocol is a communication protocol for multimedia (audio, video, text, data) information over the IP network recommended by the ITU-T.

**HTTP** (HyperText Transfer Protocol; [1.0] RFC 1945, [1.1] RFC 2616)

This protocol is a communication protocol for sending and receiving a document (HTML document) described using HTML between a Web server and a client. A file, such as image, audio, and video related to the HTML document as well as the HTML document (text file) can be transferred including information, such as a representation format.

**HTTPS** (HTTP over SSL)

The HTTP protocol provides cryptographic communication using SSL. The protocol encodes communication between a Web server and a client and can safely transfer information about secrecy and a credit card number. Because a main Web browser corresponds to this protocol, the protocol is a de-facto standard of encryption in a WWW (world wide web).

**LDAP** (Lightweight Directory Access Protocol; [v3] RFC 2251, 2252, 2253, 2254, 2255, 2256)

This client server protocol is used for accessing a directory database in which X.500 is simplified. Unlike the existing DAP (Directory Access Protocol) that uses the OSI protocol, this protocol has the specification specified for utilization over the IP network and a feature that the operation is lightweight.


The MPEG-2 TS is one of the two systems (ISO/IEC 13818-1) of the MPEG-2. Because multiple programs can be constituted in a stream, this system can also be applied to broadcasting and telecommunications (the other is a PS [Program Stream], and is used for a storage type content). It is also adopted in BS digital broadcasts.

Video or audio data is divided an extremely small fixed length packet of 188 bytes and multiplexed. Further, error detection and repetitive transmission are supported assuming that the MPEG-2 TS is applied to the environment in which a transmission error of broadcasting and telecommunications data will occur.
MPEG-4

MPEG-4 is a moving picture coding system that is being standardized by the MPEG (Moving Picture Experts Group) that is a Standardization Activity Group affiliated with the Joint Technology Standardization Activity Group JTC 1 of the ISO and the IEC. The basic standard Version 1 was officially determined as the ISO/IEC 14496 in March 1999 and Version 2 that is the high-level compatibility standard of Version 1 was officially determined in March 2001.

MPEG-4 consists of systems (MPEG-4 systems), video (MPEG-4 Visual), and audio (MPEG-4 Audio) technologies.

At first, MPEG-4 started the work targeting ultralow bitrate coding of less than 64 kilobits per second, such as wireless visual communication. However, at present, coding every object in a video, video editing/processing, integration (SNHC) of natural images and composite images, and a high-level error correction function are introduced including improvements of coding efficiency and is becoming a multimedia coding standard.

MPEG-7

Standardization activity of a content description system and an access interface (being standardized currently in January 2002).

The following standardization is targeted as the content description system.

- Descriptor: An attribute that is represented in a primitive according to the characteristics of a content and its representation value
- Description schema: A higher level attribute than a descriptor obtained by combining descriptors
- Description Definition Language: Language for defining a description schema and a descriptor.
  Investigated based on the XML schema.

MPEG-1, -2, and -4 aim at the compressed coding of a multimedia content, whereas MPEG-7 aims at the standardization of a descriptor for effectively retrieving the multimedia content. In that point, the purposes of both differ greatly. However, the technology of content feature extraction and the configuration of a search engine are beyond the object of the MPEG-7 standard.

RTP (A Transport Protocol for Real-Time Applications/Real-time Transport Protocol; RFC1889)

This transmission protocol is used to playback (streaming playback) real-time media, such as audio and video, over the IP network in real time. In general, the UDP is used as the lower layer protocol.

RTSP (Real Time Streaming Protocol; RFC 2326)

Content transfer control protocol over the IP network

SIP (Session Initiation Protocol; RFC 2543)

This protocol is used for a multimedia communication system and a terminal that handles call signaling and control in a communication service based on the Internet Protocol. The standardization is currently advancing in the IETF. The protocol is used to start, change, and terminate a session in bi-directional communication, such as a VoIP service.

TCP (Transmission Control Protocol; RFC 793)

This protocol is a standard protocol that is mainly used over the IP-based network and corresponds to the transport layer of an OSI reference model. The protocol provides a high-level application for a communication function with reliable and full duplex flow control using a datagram oriented communication function through the IP (network) layer.

UDP (User Datagram Protocol; RFC 768)

This protocol is a standard protocol that is chiefly used over the IP-based network and corresponds to the transport layer of an OSI reference model. The UDP only uses most IP packets on the lower layer from an application as they are. Unlike the TCP, the protocol is lacking in reliability because a connection management function, a response acknowledgement function, a sequence function, and a flow control function are not provided. However, the protocol has a high transfer rate because no overhead is incurred.
3. Interface Requirements on Broadband Net-live Service

3.1 Broadband Net-live Service

3.1.1 Outline of service

The broadband net-live service provides video distribution from a live hall, such as a concert hall and a stadium, via the HIKARI Service platform in real time. When the content of a video to be distributed is determined, necessary content attributes, such as the date of an event, a viewing charge, and a distribution destination are registered previously in a service menu provider site over a network. A user performs the access to this site and selects an event to be viewed from the menu. At this time, right to view is obtained by performing ticket purchase processing over the network. When a live starts, a live video is fed over the network from the hall and the user can view the video by accessing the specified site.

This service has the following features in comparison with a live video distribution service through the current Internet or a TV broadcast.

1. The high-speed/broadband video distribution that uses the HIKARI Service platform provides a high quality video service with the same level as a DVD to many users (the transfer band assumes to be about 6 Mbps of MPEG-2).
2. A favorite angle of a live video can be viewed by selecting a provided camera angles (multi-angle function).
3. A time-shifted function is provided in addition to real time live viewing. The time-shifted function enables viewing of the live from the live start even after the live started. The time-shifted function stores a live video over a network and distributes this video in accordance with a user request. Therefore, this service includes the VoD service.

Fig. 3.1.1 shows the system configuration of this service. The diagram shows the functions and necessary servers related to this service.

Copyright (C) 2001-2002 HIKARI Service Architecture Consortium. All rights reserved.
3.1.2 Outline of service function and service sequence

The outline of functional element and service sequence of each function for providing this service is described below.

- **Content provider**
  The content provider performs necessary processing, such as encoding, ad. insertion, digital watermarking, and encryption, to a video and requests the content distribution function for distribution.

- **Content distribution function**
  The content distribution function receives a viewing request from a user, assigns right to view to the user, and performs billing at the same time. The content distribution function also receives a video from the content provider when it is distributed and performs distribution to the content delivery function. The content distribution function manages placement of a cache and specifies the optimum cache server for the access from a user who desires viewing using the time-shifted function.

- **Content delivery function**
  The content delivery function sends a video to a user. The function controls the video to be delivered for a special operation request, such as a temporary halt, and an angle switching request.

- **Terminal**
  The terminal decodes the received video and displays video information on the screen. The terminal also sends control signals of the special operation and angle switching requests to the content delivery function.

The service sequence in this service is listed below. Further, Fig. 3.1.2 shows the outline.

- **Content attribute registration**
  (1) The content provider presents attributes concerning live content, such as the date, a viewing charge, and a distribution destination, to the content distribution function and requests live distribution.
  (2) The content distribution function adds the registered content to the service menu and recruits users. It is also considered to limit the number of users to perform video distribution in sufficient quality.

- **Ticket purchase**
  (3) A user accesses a live ticket purchase site from a terminal.
  (4) A user performs ticket purchase processing and acquires an ID and a password.

- **Live content feed/distribution**
  (5) The content provider performs necessary processing to a video, such as encoding, ad. Insertion, digital watermarking, and encryption.
  (6) The content provider requests the content distribution function for distribution (live content feed).
  (7) The content distribution function performs distribution of a live video to the content delivery function. On this occasion, the live video is stored in a cache server installed in the content delivery function. This stored video is used for a user who desires viewing using the time-shifted function.

- **Live viewing**
  (8) A user accesses to the content distribution function site and receives content using the ID and password.
  (9) The delivery of a live stream to a user is started. For live viewing, the live stream delivery from a delivery server is performed, and for time-shifted viewing, the stream delivery from a cache server is performed.
  (10) The operation of a “temporary halt” is enabled in the case of live viewing. While time-shifted viewing is being performed, the delivery is performed using the stored video. Accordingly, the operation of “rapid feed”, “rewinding”, or “jumping” is also enabled.
  (11) When a user makes an angle switching request, the video is switched into the video of the selected camera angle (multi-angle viewing function).

Copyright (C) 2001-2002 HIKARI Service Architecture Consortium. All rights reserved.
Completion of viewing
(12) The live completion time is reached.
(13) A live completion message is sent from a content provider to the content distribution function.
(14) The content distribution function checks the live completion and releases the session during this period.
(15) A live completion message is sent from the content distribution function (content delivery function for time-shifting) to a user.
(16) A user terminates viewing by checking the live completion.

Fig. 3.1.2 Service Sequence of Broadband Net-live Service
### 3.1.3 Required functions and issues

Functions required for providing this service are listed. A billing function and a settlement function are not included in the following functions. Because various operations are considered on these functions and it is difficult to specify a function/implementation method, the two functions are omitted from this document.

- **Content provider (1/2)**

<table>
<thead>
<tr>
<th>Function</th>
<th>Outline</th>
<th>Issue</th>
</tr>
</thead>
</table>
| Ad. insertion function (Ad. management server) | In this service, a user must basically pay a viewing charge. Accordingly, it seems to be improper to insert an advertisement in the middle of a video. A method of displaying the video on the menu screen before or after viewing in the format of banner advertising and establishing a link in a site for allowing the user to browse the details of advertising is considered to be usual. | • Advertisement insertion method  
Concerning advertisement insertion, an implementation method differs according to the advertisement form. |
| Metadata insertion function (Copyright management server) | This function inserts a content ID for preventing an incorrect copy of a content and discriminating a content attribute (however, this function is unnecessary for such a service that cannot store and copy the content).  
(1) Description of metadata  
Concerning the symbolic convention of metadata, this service recommends the content investigated by the ARIB. |                                                                                                 |
| Encryption function                           | This function performs encryption to a content and generates/manages/distributes an encoding key.                                                                                                       | • High-speed encryption and decoding processing  
An encryption technique for a video/audio is provided, but a high performance server is required to perform encryption and decoding to the video at high speed. |
| Encoding function (Encoder)                   | This function encodes an original content to a proper format  
(1) Encoding system  
In this service, we recommend the following system as an encoding system.  
- Video format : MPEG-2 MP@ML, MPEG-2 MP@M14  
- Audio format : ACC  
- Multiplex format : MPEG-2 TS  
For the reason for selecting each system and the outline of another encoding system, refer to Section 9-2 of the SA Technical Report. | • Selection of an encoding system in accordance with a technical trend  
In this service, the encoding system described on the left was recommended considering the current technical level, but MPEG-4 can also be selected according to the future technical trend. |
| Session connection request (Distribution server) | This function requests connection to the content distribution function for content feed.  
(1) Protocol used  
In this service, we recommend the RTSP as the protocol used for the session setting and release. |                                                                                                 |
### Content provider (2/2)

<table>
<thead>
<tr>
<th>Function</th>
<th>Outline</th>
<th>Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content feed function (Distribution server)</td>
<td>This function sends an original content to the content distribution function</td>
<td>(1) Protocol used&lt;br&gt;In this service, we recommend the RTP/UDP as the protocol used for the content feed.</td>
</tr>
<tr>
<td>Band control and management function (Network management server)</td>
<td>This function secures a band necessary for content feed&lt;br&gt;(1) Secured band&lt;br&gt;Because a sufficient network band that will not deteriorate an original video needs to be secured, a leased line must be used in the content feed section.</td>
<td></td>
</tr>
</tbody>
</table>

### Content distribution function (1/2)

<table>
<thead>
<tr>
<th>Function</th>
<th>Outline</th>
<th>Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content provider management (Authentication server)</td>
<td>This function provides the distribution service application screen and manages/authenticates a content provider.&lt;br&gt;(1) Authentication method&lt;br&gt;The simplest system of checking using a password/user ID is considered as a content provider authentication method.</td>
<td></td>
</tr>
<tr>
<td>Distribution content management function (Content management server/portal server)</td>
<td>This function registers content information (EPG information) in the content management (content ID, distribution condition, and billing condition) service menu. When the content distribution function previously registers information in a portal site, a user can obtain content information from the site.</td>
<td></td>
</tr>
<tr>
<td>Viewer (user) management function (Authentication server)</td>
<td>This function provides the user registration screen (viewing application screen) and checks an authenticated/registered user.&lt;br&gt;(1) Authentication system&lt;br&gt;The simplest system of checking using a password/user ID is considered as a user authentication method. To reduce a workload of the user, it is also considered that information about identification is assigned to a terminal itself and the user is identified based on the information. (However, in this case, strictly speaking, this does not indicate authentication of a person). When the terminal has an I/F, it is also considered as a means for improving convenience that an attempt is made to specify the user without depending on the terminal by using an authentication means, such as an IC card.</td>
<td></td>
</tr>
</tbody>
</table>
### Content distribution function (2/2)

<table>
<thead>
<tr>
<th>Function</th>
<th>Outline</th>
<th>Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session connection request/acceptance function</td>
<td>This function requests and accepts connection from a content provider and requests connection to the content delivery function</td>
<td>• Multicast system selection</td>
</tr>
<tr>
<td>(Distribution server)</td>
<td>(1) Protocol used</td>
<td>The multicast system includes IP multicast, splitter multicast, and Xcast.</td>
</tr>
<tr>
<td></td>
<td>In this service, we recommend the RTSP as the protocol used for the session setting and release.</td>
<td>It is realistic that an applicable system is determined according to a service request condition (the number of users) and a technical trend when a service is provided. For details, refer to Section 5-3 of the SA Technical Report.</td>
</tr>
<tr>
<td>Content distribution function (Distribution server)</td>
<td>This function performs the optimum content distribution to multiple cache sites. Both a multicast system and a unicast system are considered. (1) System when multicast is performed Several systems are considered as the system when multicast is performed, but a specific system is not selected.</td>
<td></td>
</tr>
<tr>
<td>Band control and management function</td>
<td>This function secures a band required for applying a video/audio to a distribution network. (1) Guaranteed band implementation system The guaranteed band implementation system other than Diffser is difficult at the current point. A certain degree of a stable band is secured according to a combination of “Diffserv” and “secured fairness in the same priority”. Further, when the band is insufficient, it seems to be realistic to cover the insufficient band using an error correction function at the receiving terminal side. It is desirable to perform dynamic sending control matching the band in the above environment. There is no control method established as a general method. However, the RTCP is provided as a framework in which information necessary for control is transferred.</td>
<td>• Guaranteed band protocol Because of a high-speed access network, a guaranteed band in a core network is requested. There is a system that secures a band using the RSVP. However, to realize it in the practical HIKARI Service platform, there is an issue in the aspect of facilities (installation of corresponding facilities) and the aspect of operation (evaluation when the absolute value of the band may be insufficient and processing and a billing system when it is insufficient).</td>
</tr>
<tr>
<td>(Network management server)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Content provisioning &amp; optimum server retrieval function</td>
<td>This function monitors the operating states of multiple delivery servers and selects the optimum delivery server.</td>
<td>• Optimum site selection that matches terminal attributes</td>
</tr>
<tr>
<td>(Content management server)</td>
<td></td>
<td>Because there are various user terminals that use this service (difference in a video format that can be played back and difference of a throughput), the optimum site selection method that matches these terminal attributes is an issue. For details, refer to Section 5-2 of the SA Technical Report.</td>
</tr>
</tbody>
</table>
Content delivery function (1/2)

<table>
<thead>
<tr>
<th>Function</th>
<th>Outline</th>
<th>Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Session connection request/acceptance function (Delivery server)</strong></td>
<td>This function requests and accepts connection from the content distribution function and requests and accepts connection from a terminal (1) Protocol used In this service, we recommend the RTSP as the protocol used for the session setting and release.</td>
<td></td>
</tr>
<tr>
<td><strong>Content delivery function (Delivery server)</strong></td>
<td>This function performs the optimum content delivery to multiple user terminals (1) Application of a unicast system It is considered that a multicast technique is also used in the same manner as a distribution section concerning a delivery section. However, this service used a unicast system as a candidate from the standpoint that the delivery to each user is flexibly performed.</td>
<td></td>
</tr>
<tr>
<td><strong>Time-shifted function (Cache server)</strong></td>
<td>This function delivers a video to an on-demand viewing request of a time-shifted viewing user. Because the time-shifted function is used, delivery is performed from the cache provided in a network.</td>
<td></td>
</tr>
<tr>
<td><strong>Band control and management function (Network management server)</strong></td>
<td>This function secures a band required for applying a video/audio to a delivery network (1) Guaranteed band implementation system Same content described in the content distribution function</td>
<td>• Guaranteed band protocol Same content described in the content distribution function</td>
</tr>
<tr>
<td><strong>Stream switching (Delivery server)</strong></td>
<td>This function changes an audio/video stream in accordance with a user angle switching request.</td>
<td>• Angle switching implementation system The following items are considered as the implementation system. (1) Method of switching a stream delivered to a user with a control signal. One stream is delivered to the terminal. (2) Multiple streams are delivered to the terminal and switched at the terminal side (a few angles of two or three streams are possible) For details, refer to Section 8-3 of the SA Technical Report.</td>
</tr>
</tbody>
</table>
Content delivery function (2/2)

<table>
<thead>
<tr>
<th>Function</th>
<th>Outline</th>
<th>Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Special operation control (Delivery server)</td>
<td>This function realizes a special operation, such as a temporary halt, rapid feed, rewinding, and jumping. (1) Protocol used In this service, we recommend implementation using the RTSP as the protocol that realizes the special function.</td>
<td>• Integration of the RTSP specification undetermined part Only the current RTSP (RFC 2326) will produce various interpretations. Accordingly, the point in which clarification is necessary in the current RTSP specification was investigated. For details, refer to Section 9-4 of the SA Technical Report and the appendix of this document. • Combination with the time-shifted function A live video needs to be stored in a cache and delivered at the same time. This is technically possible. However, when this processing is performed at the same time with a special operation, such as rapid feed or rewinding, it is considered that there is a limit in the number of streams that can be delivered from the standpoint of a throughput. For details, refer to Section 5-5 of the SA Technical Report.</td>
</tr>
</tbody>
</table>
### Terminal

<table>
<thead>
<tr>
<th>Function</th>
<th>Outline</th>
<th>Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection and disconnection to network</td>
<td>This function connects/disconnects with a network.</td>
<td>• System of connection and disconnection to a network</td>
</tr>
<tr>
<td></td>
<td>(1) Terminal IP address</td>
<td>When a large number of terminals perform always-on connection and a fixed IP address, the IP address may be insufficient in IPv4.</td>
</tr>
<tr>
<td></td>
<td>In this service, we recommend the always-on connection and a fixed IP address.</td>
<td></td>
</tr>
<tr>
<td>Session connection request</td>
<td>This function requests connection to the content delivery function.</td>
<td>• Synchronization of a video and an audio</td>
</tr>
<tr>
<td></td>
<td>(1) Protocol used</td>
<td>How much strictly synchronization must be obtained is changed according to the content of a service.</td>
</tr>
<tr>
<td></td>
<td>In this service, we recommend the RTSP as the protocol used for the session setting and release.</td>
<td></td>
</tr>
<tr>
<td>Content control (play, stop, fast, etc.)</td>
<td>This function receives and decodes the encoded video/audio stream.</td>
<td>• Trade off between quality and a reduced delay</td>
</tr>
<tr>
<td></td>
<td>• Synchronization of a video and an audio</td>
<td>The optimum buffering size will be decided by the trade off between guarantee of quality (guarantee of a fixed refresh rate and image quality) and a reduction of a delay. The trade off differs according to an application. However, it is considered that priority is given to the quality for a broadband net-live.</td>
</tr>
<tr>
<td>Buffering function</td>
<td>This function absorbs a fluctuation and a delay of a video signal.</td>
<td>• Multi stream screen display</td>
</tr>
<tr>
<td></td>
<td>• Trade off between quality and a reduced delay</td>
<td>Refer to Section 8-3 of the SA Technical Report.</td>
</tr>
<tr>
<td>Video stream screen display</td>
<td>This function properly displays the received video stream on the screen using a Web browser or a dedicated application.</td>
<td></td>
</tr>
<tr>
<td>Terminal event sending function</td>
<td>This function notifies an event to the delivery server by pressing a button or a key on a terminal.</td>
<td></td>
</tr>
<tr>
<td>Terminal security</td>
<td>This firewall function is used to prevent intrusion into home equipment. However, a video/audio stream needs to reach a terminal.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(1) Realization of a firewall function</td>
<td>It is desirable that the firewall function should be realized by a gateway function (Proxy function) at the RTSP and HTTP application levels.</td>
</tr>
</tbody>
</table>
3.2 Interface in Broadband Net-live Service

The specific interface requirements of a broadband net-live service are shown here in accordance with the following flow.

1. Content attribute registration function
2. Ticket purchase function
3. Live content feed/distribution function
4. Live viewing function
5. Completion of viewing function

3.2.1 Content attribute registration function

3.2.1.1 Outline of function

This function registers metadata of a live content. The function consists of three phases of “executing and terminating of the registration wizard processing”, “ID and password input processing”, and “content metadata registration processing”.

The content provider starts the content registration wizard that the content distribution function provides and reaches the content metadata registration processing via the user authentication (ID and password input processing). In this sub-section, the content metadata indicates content attributes, such as a content title and outline.

In the executing and terminating of the registration wizard processing, the content provider creates content metadata and provides it to the content distribution function prior to content distribution. The content distribution function stores the provided metadata in the content database. Some part of the data is used as EPG when a ticket is purchased prior to the content distribution, and the other part of the data is distributed and delivered together with the content and is used to control content viewing.

3.2.1.2 Content attribute registration function sequence

The sequence of this function is shown in Fig. 3.2.1.

![Fig. 3.2.1 Content Attribute Registration Function Sequence](http://example.com/image.png)

Copyright (C) 2001-2002 HIKARI Service Architecture Consortium. All rights reserved.
3.2.1.3 Service sequence and applicable protocol set

The protocol set used in this function is shown in Table 3.2.1.

<table>
<thead>
<tr>
<th>Processing</th>
<th>Interface function</th>
<th>Data item</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executing and terminating of the registration wizard processing</td>
<td>(1) Registration wizard access</td>
<td>Registration wizard display request event</td>
<td>Upper layer: HTTP Lower layer: TCP</td>
</tr>
<tr>
<td></td>
<td>(6) Registration completion notification</td>
<td>Registration completion notification event</td>
<td></td>
</tr>
<tr>
<td>ID and password input processing</td>
<td>(2) ID/password request</td>
<td>ID/password request event</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(3) ID/password reply</td>
<td>- Content registrant ID - Password</td>
<td></td>
</tr>
<tr>
<td>Content metadata registration processing</td>
<td>(4) Registration wizard display</td>
<td>Registration wizard display screen information</td>
<td>Upper layer: HTTPS Lower layer: TCP</td>
</tr>
<tr>
<td></td>
<td>(5) Metadata registration</td>
<td>Content metadata</td>
<td>(Note: What information is to be registered as metadata basically conforms to the ARIB proposals. For details, refer to the SA Technical Report.)</td>
</tr>
</tbody>
</table>

3.2.1.4 Applicable condition

(1) ARIB proposals concerning content metadata

The ARIB proposals indicate the content of the proposal document “Proposals of General Broadcasting System based on Home Servers” from the Association of Radio Industries and Businesses (ARIB) for the technical requirements on a broadcast system that is edited by the Committee for Broadcasting System based on Home Servers in Information Communication Council based on Requests for Advice from the Minister of Public Management.


(a) Description Scheme of metadata

The proposal document proposes the following description scheme specified in “TV Anytime Forum TVA SP003 v1.1” as the description scheme of content metadata.

- Content Description DS (Description Scheme)
- Instance Description DS
- Segmentation DS

(b) Description language of metadata

The proposal document proposes the Description Definition Language (DDL) specified in MPEG-7 as the description language of content metadata.

The DDL is a language based on XML Schema that is the Schema language in which the standardization of MPEG-7 was decided on by W3C, and is adopted even in the TV Anytime Forum.
3.2.2 Ticket purchase function

3.2.2.1 Outline of function

In this service, a viewer (user) needs to purchase a ticket prior to the viewing of a live content. This function provides an HI (human interface) when viewers purchase this ticket.

This function consists of three phases of “user authentication processing”, “ticket selection processing”, and “ticket purchase processing”.

When a user turns on the terminal power supply, the terminal starts a browser, automatically connects it to the service portal, and makes a user authentication request. Further, the service portal transfers this access (authentication request) to the content distribution function.

The content distribution function performs the user authentication from a (fixed) IP address and a MAC address of the terminal, or an ID and a password set previously in the terminal.

After the terminal is authenticated, the user purchases the ticket by selecting a target item from the menu. In this service, the content distribution function provides a series of procedures when the ticket is purchased.

When the ticket purchase procedure is completed, the content distribution function forwards the ID and password required when the live content is viewed.

3.2.2.2 Ticket purchase function sequence diagram

The sequence of this function is shown in Fig. 3.2.2.
3.2.2.3 Service sequence and applicable protocol set

The protocol set used in this function is shown in Table 3.2.2.

<table>
<thead>
<tr>
<th>Processing</th>
<th>Interface function</th>
<th>Data item</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>User authentication processing</td>
<td>(1) User authentication request (portal)</td>
<td>• User ID</td>
<td>Upper layer: HTTPS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Password</td>
<td>Lower layer: TCP</td>
</tr>
<tr>
<td></td>
<td>(2) User authentication request (distribution function)</td>
<td>• User ID</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Password</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(3) User authentication response (distribution function)</td>
<td>Authentication result</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(4) User authentication response (portal)</td>
<td>Authentication result</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(5) Initial menu request</td>
<td>Initial menu request event</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(6) Initial menu display</td>
<td>Initial menu request event</td>
<td></td>
</tr>
<tr>
<td>Ticket selection processing</td>
<td>(7) Service menu request</td>
<td>Service menu request event</td>
<td>Upper layer: HTTP</td>
</tr>
<tr>
<td></td>
<td>(8) Service menu display</td>
<td>Service menu screen information</td>
<td>Lower layer: TCP</td>
</tr>
<tr>
<td></td>
<td>(9) Live ticket list request</td>
<td>Live ticket list request event</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(10) Live ticket list display</td>
<td>Live ticket list screen information</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(11) Live event specification</td>
<td>Ticket category</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(12) Live ticket purchase screen display</td>
<td>Live ticket purchase screen information</td>
<td></td>
</tr>
<tr>
<td>Ticket purchase processing</td>
<td>(13) Ticket purchase request</td>
<td>Ticket purchase request event</td>
<td>Upper layer: HTTPS</td>
</tr>
<tr>
<td></td>
<td>(14) Customer ID, password notification</td>
<td>• Reserved ID (for viewing time)</td>
<td>Lower layer: TCP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Password (for viewing time)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(15) Ticket purchase</td>
<td>Ticket purchase agreement event</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(16) Ticket purchase completion notification</td>
<td>Ticket purchase completion event</td>
<td></td>
</tr>
</tbody>
</table>

3.2.2.4 Applicable condition

(1) Terminal IP address

The premise of this function is that the terminal has a global fixed IP address and is ready for being able to be connected to a network. The processing content and sequence are the same as even when an IP address is dynamically allocated using the DHCP.

(2) Relationship between the service portal and content distribution function

The relationship between the service portal and content distribution function depends on each service. As an actual service, in many cases, the content distribution provider has also the service portal function at the same time. However, both functions are different as a player model.

In this service, a description was made so that the service portal for simplicity can display the initial menu and subsequent services can be executed using the content distribution function.
(3) Authentication with single sign-on

The authentication method in this service is assumed on single sign-on in the service portal. Further, to obtain the presence of the relevant content in the service portal, a directory server is provided in a service portal system. This directory service uses LDAP (Lightweight Directory Access Protocol) v3 (RFC 2251, 2252, 2253, 2254, 2255, 2256) as the candidate of an applicable protocol.

The service portal first authenticates whether the accessed terminal is a terminal of which connection may be accepted (or a user to whom a service may be provided). After the authentication is completed, the service portal sends an authentication request from the terminal to the content distribution function via a cookie based on the information obtained by the directory service. The content distribution function authenticates the usability of the service that this function provides based on the information transferred from the service portal.

For example, even if a video-dedicated content distribution function is provided, when a user contracts only live viewing, a distribution service, such as the cinema menu is not displayed (or even if it is displayed, at least the menu is not made active) or any menu of a predetermined improper content is not displayed when the user is a child), can be chosen.

After the content distribution function returns an authentication result to the service portal after it performs authentication in the above service level. Similarly, the service portal returns the authentication result to the terminal in the same manner.

In actual, multiple times of authentication allow the user to be recognized as one-stop or one-time (single) authentication in the service portal in this manner, thereby securing security and serving convenience of the user.

(4) Sequence up to service menu display

In Fig. 3.2.2, the terminal makes an authentication request to the service portal at the same with the browser start like a live viewing service dedicated terminal. Normally, the terminal may not be connected to the service portal in the extension of the power-on, and the user operation may be performed before the terminal is connected to the content distribution function. In that meaning, the sequence of Fig. 3.2.2 differs from the sequence of an actual service in that sense, but such notation was made here for simplicity.
3.2.3 Live content feed/distribution function

3.2.3.1 Outline of function

This function starts live content distribution. The function consists of three phases of “delivery preparation status acknowledgement processing”, “content feed processing”, and “content distribution processing”.

The delivery preparation status acknowledgement processing checks whether the delivery preparation is completed in the content delivery function prior to the distribution of a live content. Further, in the content feed processing and the content distribution processing, the distribution function accepts distribution from the content provider waiting for the preparation completion of the delivery function and relays it to the delivery function.

In this case, content should be distributed to the plural number of mirror/cache servers. In order to distribute content in an efficient way, several multicasting schemes exist, such as IP multicasting and application splitting, etc. However, this function is premised on the push splitting for simplicity. Besides, the feed from the content provider to the content distribution function is premised on a unicast, but another method will not be forbidden.

3.2.3.2 Live content feed/distribution function sequence diagram

The sequence of this function is shown in Fig. 3.2.3.
### 3.2.3.3 Service sequence and applicable protocol set

The protocol set used in this function is shown in Table 3.2.3.

**Table 3.2.3 Protocol Set Applied to Live Content Feed/Distribution Function**

<table>
<thead>
<tr>
<th>Processing</th>
<th>Interface function</th>
<th>Data item</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delivery preparation status acknowledgement processing</td>
<td>(1) Delivery preparation acknowledgement request</td>
<td>Delivery preparation completion notification event (ANNOUNCE method of RTSP)</td>
<td>Upper layer: RTSP Lower layer: TCP</td>
</tr>
<tr>
<td></td>
<td>(2) Delivery preparation acknowledgement response</td>
<td>Delivery preparation completion acknowledgement event (response event of ANNOUNCE method)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(3) Session setting request</td>
<td>SETUP method of RTSP</td>
<td>Upper layer: RTSP Lower layer: TCP</td>
</tr>
<tr>
<td></td>
<td>(4) Session setting response</td>
<td>Response event of SETUP method</td>
<td></td>
</tr>
<tr>
<td>Content feed processing</td>
<td>(5) Feed request</td>
<td>Feed request event</td>
<td>Upper layer: HTTPS Lower layer: TCP</td>
</tr>
<tr>
<td></td>
<td>(6) Feed acknowledgment</td>
<td>Feed request response event</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(7) Session setting request</td>
<td>SETUP method of RTSP</td>
<td>Upper layer: RTSP Lower layer: TCP</td>
</tr>
<tr>
<td></td>
<td>(8) Session setting response</td>
<td>Response event of SETUP method</td>
<td></td>
</tr>
<tr>
<td>Content distribution processing</td>
<td>(9) Live material feed</td>
<td>Content data</td>
<td>Upper layer: RTP Lower layer: UDP</td>
</tr>
<tr>
<td></td>
<td>(10) Live material distribution</td>
<td>Content data</td>
<td>Upper layer: RTP Lower layer: UDP</td>
</tr>
</tbody>
</table>

### 3.2.3.4 Applicable condition

1. **Video format and audio format**
   - Video: MPEG-2 MP@ML, MP@M14
   - Audio: MPEG-2 AAC
   - Multiplex format: MPEG-2 TS

   Besides, in the implementation of MPEG-4 that is currently put into practical use, a video having the quality with the same level as a DVD of the requirements of this service cannot be distributed/delivered. However, the MPEG-4 protocol setup work is also currently advancing and the protocol that corresponds to the same high image quality as the current MPEG-2 is going to be put into practical use in near future.

2. **Video control protocol between servers**
   - In this live content feed/distribution function, RTSP is applied as a streaming control protocol between a content provider and content distribution servers, and between content distribution servers and content delivery servers as well. However, other streaming control protocols must be also investigated and discussed, since the current RTSP does not support push control, reliability control, multicast control, which are supposed to be important functions in the distribution networks.

3. **Quality of network between content provider and content distribution function**
   - In this service, because the UDP is used for the protocol of the transport layer in the transmission of a live content, retransmission of a packet is not performed. Accordingly, the network between the content provider and content distribution function that transfer an original content needs to secure sufficient quality in which a band or packet loss will not be generated.

   Further, to increase durability in the transport layer level, a method of using the TCP instead of the UDP is also considered after a buffer is provided at the receiving side. However, a difficult problem, such as multicast cannot be used, will continue occurring toward its realization when the TCP is used. Besides, even when the TCP is used, needless to say, reasonable network quality is required.
(4) Delivery preparation acknowledgement sequence
   It is also considered that delivery preparation acknowledgement is actively posted from the delivery function side to the distribution function side contrary to Fig. 3.2.3. However, in this service, because the distribution function functionally becomes the nucleus, it is assumed to be usual to make an inquiry from the content distribution function side. Accordingly, this sequence is described as shown in the diagram.

(5) Multicast
   This function is premised on an assumption that multicast distribution is performed from the content distribution function to multiple content delivery functions. However, as described in the outline of the function, there are various systems to perform multicast in a broad sense. This function is premised on an assumption that the mechanism of a multicast group is established.

(6) Multi-angle streaming
   In this service, one feature is a multi-angle. The service is premised on a method of multiplexing a video/aural signal of each angle and distributing/delivering it in a single stream as the implementation form. There is a method of allocating a stream to the individual angles respectively and distributing/delivering it in a multi stream as another system.
   In the former, a function of decoding a multiplexed signal is requested to a terminal and a certain degree of performance is requested to the terminal. On the other hand, in the latter, the signal that the terminal received may be reproduced as is. However, because a necessary band is increased every viewing, a load is applied to a network.
   Besides, in the former, real time synchronization needs to be obtained between streams in some method.
3.2.4 Live viewing function

3.2.4.1 Outline of function

This function selects a content that a user views and performs actual viewing. This function consists of four phases of “user authentication processing”, “menu selection processing”, “ticket authentication processing”, and “live viewing processing”.

The “user viewing processing” and the “menu selection processing” have the similar content of the ticket purchase function of Section 3.2.2. In the “ticket authentication processing”, a user obtains the authentication of a content provider using the ID or password posted when a ticket was purchased. The “live viewing processing” accesses the delivery site of an actual content and performs viewing. Further, a special operation, such as a temporary halt, rapid feed, rewinding, or jumping, can be performed by performing control operation to the delivery server in the course of viewing. Further, the switching of a camera angle is also enabled.

3.2.4.2 Live viewing function sequence diagram

The sequence of this function is shown in Fig. 3.2.4.
4. Live viewing

Service menu screen
Please choose from following menus.

- Movie
- Web
- Mail
- Content
distribution
- Live
- End

Competition card viewing screen
Japan vs. Italy
Please enter ID and PW.

ID: ****
PW: ****
OK
Cancel

Authentication screen
Japan vs. Italy
Please enter ID and PW.

ID: ****
PW: ****
OK
Cancel

Video control (play, stop, fast, ...) screen

- Camera A
- Camera B
- Camera C

- MPEG-2 TS over UDP
- RTSP
- HTTP

Terminal
Browser start

Service Portal
User authentication request (portal)

Content Provider
Content distribution function

Content Delivery Function
User authentication request (distribution function)

User authentication request response (portal)

User authentication request (distribution function)

Authentication

HTTP

Authentication screen display (customer ID, password request)

HTTP

ID, password notification

User authentication request response (portal)

User authentication request (distribution function)

Live distribution request

Request routing (Selection of the optimum delivery server)

Delivery server, IP address notification

Contents description in the “Live content function of Section 3.2.3”

Live delivery request

RTSP(PLAY)

Live delivery

MPEG-2 TS over UDP

Store

Special operation (temporary halt)

RTSP(PAUSE)

Special operation acknowledgement

RTSP

Temporary halt

Store

Rapid feed/rewinding/jumping

RTSP(PAUSE)

Special operation acknowledgement

RTSP

Live delivery request

RTSP(PLAY)

Video of rapid feed/rewinding/jumping

MPEG-2 TS over UDP

Store

Angle switching

RTSP

Another angle video

MPEG-2 TS over UDP

Store

Fig. 3.2.4 Live Viewing Function Sequence Diagram

Copyright (C) 2001-2002 HIKARI Service Architecture Consortium. All rights reserved.
### 3.2.4.3 Service sequence and applicable protocol set

The protocol set used in this function is shown in Table 3.2.4.

<table>
<thead>
<tr>
<th>Processing</th>
<th>Interface function</th>
<th>Data item</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User authentication processing</strong></td>
<td>(1) User authentication request (portal)</td>
<td>• User ID</td>
<td>Upper layer: HTTPS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Password</td>
<td>Lower layer: TCP</td>
</tr>
<tr>
<td></td>
<td>(2) User authentication request (distribution function)</td>
<td>• User ID</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Password</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(3) User authentication request response (distribution function)</td>
<td>• Authentication result</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(4) User authentication request response (portal)</td>
<td>• Authentication result</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(5) Initial menu request</td>
<td>• Content URL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(6) Initial menu display</td>
<td>• Initial menu request event</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Initial menu screen information</td>
<td></td>
</tr>
<tr>
<td><strong>Menu selection processing</strong></td>
<td>(7) Service menu request</td>
<td>Service menu request event</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(8) Service menu display</td>
<td>(Access the URL acquired in (4))</td>
<td>Upper layer: HTTP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Service menu screen information</td>
<td>Lower layer: TCP</td>
</tr>
<tr>
<td></td>
<td>(9) Competition card request</td>
<td>Competition card request event</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(10) Competition card screen display</td>
<td>Competition card screen information</td>
<td></td>
</tr>
<tr>
<td><strong>Ticket authentication processing</strong></td>
<td>(11) Live distribution request</td>
<td>Live distribution request event</td>
<td>Upper layer: HTTPS</td>
</tr>
<tr>
<td></td>
<td>(12) Authentication screen display (customer ID, password request)</td>
<td>• Reserved ID (for viewing time)</td>
<td>Lower layer: TCP</td>
</tr>
<tr>
<td></td>
<td>(13) ID, password notification</td>
<td>• Password (for viewing time)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(14) Delivery server, IP address notification</td>
<td>• Delivery server IP, port number</td>
<td></td>
</tr>
<tr>
<td><strong>Live viewing processing</strong></td>
<td>(15) Session setting request</td>
<td>• SETUP method of RTSP</td>
<td>Upper layer: RTSP</td>
</tr>
<tr>
<td></td>
<td>(16) Setup acknowledgement</td>
<td>• SETUP method acknowledgement event</td>
<td>Lower layer: TCP</td>
</tr>
<tr>
<td></td>
<td>(17) Live delivery request</td>
<td>• PLAY method of RTSP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(19) Special operation (temporary halt)</td>
<td>• PAUSE method of RTSP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(20) Special operation acknowledgement</td>
<td>• PAUSE method acknowledgement event</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(21) Special operation (rapid feed/rewinding/jumping)</td>
<td>• PAUSE method of RTSP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(22) Special operation acknowledgement</td>
<td>• PAUSE method acknowledgement event</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(23) Live delivery request</td>
<td>• PLAY method of RTSP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(25) Angle switching request</td>
<td>• PLAY method of RTSP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(26) Another angle video</td>
<td>• Content data</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Content data</td>
<td>Upper layer: MPEG-2 TS over UDP</td>
</tr>
</tbody>
</table>

Copyright (C) 2001-2002 HIKARI Service Architecture Consortium. All rights reserved.
3.2.4.4 Applicable condition

(1) Terminal IP address
The premise of this function is that the terminal has a global fixed IP address and is ready for being able to be connected to a network. The processing content and sequence are the same as even when an IP address is dynamically allocated using the DHCP.

(2) Authentication with single sign-on
The authentication method in this service is assumed on single sign-on in the service portal. Further, to obtain the presence of the relevant content in the service portal, a directory server is provided in a service portal system. This directory service uses LDAP (Lightweight Directory Access Protocol) v3 (RFC 2251, 2252, 2253, 2254, 2255, 2256) as the candidate of an applicable protocol.

The service portal first authenticates whether the accessed terminal is a terminal of which connection may be accepted (or a user to whom a service may be provided). After the authentication is completed, the service portal sends an authentication request from the terminal to the content distribution function via a cookie based on the information obtained by the directory service. The content distribution function authenticates the usability of the service that this function provides based on the information transferred from the service portal.

For example, even if a video-dedicated content distribution function is provided, when a user contracts only live viewing, a distribution service, such as the cinema menu is not displayed (or even if it is displayed, at least the menu is not made active) or any menu of a predetermined improper content is not displayed when the user is a child), can be chosen.

After the content distribution function returns an authentication result to the service portal after it performs authentication in the above service level. Similarly, the service portal returns the authentication result to the terminal in the same manner.

In actual, multiple times of authentication allow the user to be recognized as one-stop or one-time (single) authentication in the service portal in this manner, thereby guaranteeing security and serving convenience of the user.

(3) Simplification of ticket authentication processing
In the above sequence, when a user views a live, the user reenters the ID/password of a ticket. To save the labor of this input, the following two methods are considered.

(a) A method of recording the ID/password in the terminal and automatically displaying it when the user views the live. However, this method has a security problem.

(b) A method of allowing a content provider to record the ID/password of the user and automatically confirm whether the user purchases the ticket from the correspondence with the user ID. This method requires the user ID in addition to the ID/password (premised on a membership service).

(4) Necessity of RTSP specification extension
In the RTSP, particularization is currently advancing by the IETF. The particularization in implementation levels is not reached and only the current specification (RFC 2326) will produce various interpretations. Accordingly, the part in which the integration of the specification may be required was investigated. The RTSP is formed with a combination of methods and headers. The main content to be investigated is designation of a header required for each method, designation of the content to be described by each header, and definition of the method/header for specifying a timeout. For the details of the investigation concerning the RTSP, refer to Section 9-4 of the SA Technical Report.
(5) Selection of transmission protocol
As the protocol that performs the transmission of video/audio media, we compare MPEG-2 TS over UDP and RTP and select the former as the candidate. This is performed for the following reasons.
- The RTP is premised on a non-guaranteed band Internet network (IP multicast network in particular). If a sufficient band can be secured on the HIKARI Service platform, the control of RTP is unnecessary.
- If an overhead part of the RTCP is sufficiently installed, the processing is heavy.
- When a stream that already includes a time stamp and multiplex information is handled, such as the MPEG, a common header of the RTP becomes unnecessary.
- In the streaming in which compression data is handled at a high bit rate, the RTP is not used in most cases.
However, the above is conditional content of “if a sufficient band can be secured on the HIKARI Service platform”. If a guaranteed band is insufficient, the RTP needs to be used. For the details of the investigation concerning a transmission protocol, refer to Section 9-3 of the SA Technical Report.

(6) Selection of video format and audio format
The following video format and audio format are recommended respectively.
- Video format : MPEG-2 MP@ML, MPEG-2 MP@M14
- Audio format : AAC
- Multiplex format : MPEG-2 TS
For the content of investigation regarding the video format and the audio format, refer to Section 9-2 of the SA Technical Report.

(7) Special operation implementation method
This service enables a special operation, such as “rapid feed”, “rewinding”, or “jumping” to a user who views a live using the time-shifted function. It is technically possible to write a live video to a cache and deliver it at the same time. However, this processing is performed simultaneously with the special operation, such as rapid feed and rewinding, it is considered that there is a limit in the number of streams that can be delivered from the standpoint of a throughput. For details, refer to Section 5-5 of the SA Technical Report.

(8) Multi-angle implementation method
The following methods of realizing a multi-angle are considered.
(a) A method of switching a stream delivered to a user with a control signal. One stream is delivered to a terminal. (In this case, because the video content of another angle are not sent to the terminal, the user cannot know what video is applied at another angle. It is considered that the video of the sub screen is set to about less than 1 Mbps (a band to be used is reduced) and is also delivered in the same stream).
(b) Multiple streams are delivered to the terminal and switched at the terminal side (possible for a few angles of about two or three streams)
Fig. 3.2.4 is the example of (a). The realization of system (a) is easy because a network band is saved and terminal processing is easy. Further, such issues as the transfer of information about the interrelationship between the main screen and the sub screen and synchronization between multiple screens will occur according to the implementation method. For the details of these issues, refer to Section 8-3 of the SA Technical Report.

(9) Quality of network between content delivery function and terminal
In this service, because the UDP is used for the protocol of the transport layer in the transmission of a live content, retransmission of a packet is not performed. Accordingly, the network between the content provider and content distribution function that transfer an original content needs to secure sufficient quality in which a band or packet loss will not be generated. For the guaranteed band/priority control implementation method, refer to Section 5-4 of the SA Technical Report.
3.2.5 Completion of Viewing

3.2.5.1 Outline of function

This function provides a function necessary for the completion of live viewing. Two cases are considered as the completion of viewing procedure. One is the case where a live is terminated and a completion request is made from the content provider side. In this case, completion notification and session release processing are performed sequentially in each section from the content provider side to the terminal side. The processing in each section is basically performed as the same processing. Another case is where a user suspends viewing in the course of a live. In this case, the completion notification and session release processing are performed from the terminal side.

3.2.5.2 Completion of Viewing function sequence diagram

The sequence of this function is shown in Fig. 3.2.5 (a) for the live completion and in Fig. 3.2.5 (b) for viewing suspension.

---

Fig. 3.2.5 (a) Completion of Viewing Function Sequence (for Content Completion)
Fig. 3.2.5 (b) Completion of Viewing Function Sequence (for Viewing Suspension)
### 3.2.5.3 Service sequence and applicable protocol set

The protocol set used in this function is shown in Table 3.2.5.

<table>
<thead>
<tr>
<th>Processing</th>
<th>Interface function</th>
<th>Data item</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(1) Live material feed end notification</td>
<td>• ANNOUNCE method of RTSP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(2) Completion acknowledgement</td>
<td>• ANNOUNCE method acknowledgement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(3) Session release</td>
<td>• TEARDOWN method of RTSP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(4) Release acknowledgement</td>
<td>• TEARDOWN method acknowledgement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(5) Live distribution end notification</td>
<td>• ANNOUNCE method of RTSP</td>
<td></td>
</tr>
<tr>
<td>Completion of viewing (for content completion)</td>
<td>(6) Completion acknowledgement</td>
<td>• ANNOUNCE method acknowledgement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(7) Session release</td>
<td>• TEARDOWN method of RTSP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(8) Release acknowledgement</td>
<td>• TEARDOWN method acknowledgement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(9) Delivery end notification</td>
<td>• ANNOUNCE method of RTSP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(10) Completion acknowledgement</td>
<td>• ANNOUNCE method acknowledgement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(11) Session release</td>
<td>• TEARDOWN method of RTSP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(12) Release acknowledgement</td>
<td>• TEARDOWN method acknowledgement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(13) Completion notification</td>
<td>• PAUSE method of RTSP</td>
<td>Upper layer: RTSP</td>
</tr>
<tr>
<td></td>
<td>(14) Completion acknowledgement</td>
<td>• PAUSE method acknowledgement</td>
<td>Lower layer: TCP</td>
</tr>
<tr>
<td></td>
<td>(15) Session release</td>
<td>• TEARDOWN method of RTSP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(16) Release acknowledgement</td>
<td>• TEARDOWN method acknowledgement</td>
<td></td>
</tr>
</tbody>
</table>

### 3.2.5.4 Applicable condition

(1) Necessity of RTSP specification extension

In the RTSP, particularization is currently advancing by the IETF. The particularization in installation levels is not reached and only the current specification (RFC 2326) will produce various interpretations. Accordingly, the part in which the integration of the specification may be required was investigated. The RTSP is formed with a combination of methods and headers. The main content to be investigated is designation of a header required for each method, designation of the content to be described by each header, and definition of the method/header for specifying a timeout. For the details of the investigation concerning the RTSP, refer to Section 9-4 of the SA Technical Report.
4. Interface Requirements on Bi-directional Video Communication Service

4.1 Bi-directional Video Communication Service

4.1.1 Outline of service

In this section, an asymmetric visual communication service of N:M is investigated as one of bi-directional video communication services. In the past, there were a TV telephone and a TV conference system in the real-time visual communication systems. In those systems, technical emphasis was laid on how to achieve a good quality audio and video, using a restricted band for their bi-directional flow, but the communication which brings reality to people could not necessarily be realized by the use of dedicated equipment. However, because a broadband can be secured greatly by the spread of the HIKARI Service Network Architecture, it is considered that the future bi-directional video communication service will have the following features.

1. High image quality (exceeding the degree of a VHS), high sound quality (exceeding fixed telephone sound quality)
2. The bi-directional video communication services are not only symmetrical ones between N points but also asymmetrical ones where a flow of audios/videos is asymmetrical. (Example: video/audio of only an instructor for a trainee, and videos/audios of all trainees for the instructor)
3. A communication party changes according to a scene. (Example: Lecture by an instructor → group learning)
4. Participation of a new user in the service between the N points and withdrawal from the service are enabled. (Example: coming home earlier than usual)
5. The delivery of content (delivery of teaching materials data) is enabled during visual communication.
6. A low delay in such a degree in which communication can be realized smoothly is obtained (it is desirable that an audio and a video should reach the remote terminal within 300 msec one-way).

In this section, a distance learning service is investigated as one of bi-directional video communication services. There may also be various forms in the distance learning. Here, we investigate the service which provides not only the mutual communication between an instructor and trainees, but also some communication modes, such as trainee’s late participation/early leaving and a group learning. Here, the details on 1) registration on a distance learning service, 2) posting of teaching materials, 3) class attendance registration, 4) opening the class, 5) trainee’s intermediate participation/early leaving/speech, 6) video display switching, 7) inter-trainee conversation, and 8) closing the class are investigated.

The configuration in which such service is realized is roughly divided into two configurations. One configuration is that a stream is mutually sent between terminals and another configuration is that streams from all terminals are collected to the MCU and a video to be displayed there is sent to each terminal. Here, the configuration of the latter is premised and the streams from trainees and an instructor are collected to the MCU by unicast. Subsequently, the stream is sent to the terminal of the instructor/trainee by the unicast after processing, such as synthesis, is performed at the MCU.

Fig. 4.1.1 is the system configuration of this service. The diagram shows the provider related to this service and servers that each provider requires.
4.1.2 Service function and outline of service sequence

Each function for providing this service and the outline of service sequence are shown below.

- Application service (distance learning service) provider
  The application service (distance learning service) provider has functions of the management of a user who receives the service (user management server), the management of a service state (distance learning server function), a request for the connection and disconnection of a session to which the service is provided, the control of a of streams (MCU function), and the management of teaching materials (teaching materials server).

- Service portal function
  This function checks whether the initial menu is provided to a terminal and the terminal is registered in the portal.

- Session management provider function
  The session management provider establishes a session based on an instruction from the MCU (session management server function). On the occasion, a connection destination is specified from a name (directory service function).

- Content delivery function
  The content delivery provider has a video content delivery function and a main communication pass setting function in bi-directional video communication. When video materials are used in the form of VoD, a cache/mirror management function may be used. (Basically, the same function used by the live content delivery function).
Terminal
The terminal has an I/F that displays an instructor, another trainee, and teaching materials and allows a trainee to return a response to the teaching materials and the instructor using audio/video/menu selection/button/key input. The great difference from a net-live is that the terminal for a bi-directional video communication service has some video input means, such as a camera, and a video encodes function.

The service sequence of this service is shown below.

* Registration on a distance learning service
  1. An instructor (distance learning service executor) requests the distance learning function from the Web-based menu screen. When the application of time and the number of trainees is executed, user registration is performed.

* Posting of teaching materials
  2. An instructor registers teaching materials used in an instruction in the distance service provider (teaching materials server).

* Class announcement
  3. A class announcement is displayed in the portal.
  4. The (video) teaching materials to be used are registered in the delivery function server.

* Class attendance registration
  5. A trainee registers a class from the Web-based menu.

* Opening the class
  6. Sessions to the terminals of an instructor and all trainees are established from the MCU. The videos and audios from each terminal are collected to the MCU and the videos/audios displayed on each terminal are created by synthesis and coupling, then sent to each terminal.

* Trainee’s intermediate participation/early leaving/speech
  7. Intermediate participation: The trainee who could not participate at the opening of the class joins in the class from the Web-based menu. A session to the participant who came late is established from the MCU.
  8. Early leaving: A trainee in a class exits from the Web-based menu. The MCU disconnects a session to the trainee.
  9. Trainee speech: A trainee requests to make a speech from the Web-based menu. After an executor checks the request, the MCU switches the stream.

* Display video switching
  10. The video/audio/menu screen displayed on each terminal is changed according to a request (selection on the Web-based menu) from an instructor. The video/audio is changed when a flow of a stream and the synthesized content of the stream is changed.

* Conversation between trainees (group learning)
  11. A flow of a stream is switched by the MCU according to a request from an instructor and the formation of group learning is produced.

* Closing the class
  12. The class can be terminated by selecting the Web-based menu (a trainee can exit from the class).
Fig. 4.1.2 Service Sequence of Bi-directional Video Communication Service

Copyright (C) 2001-2002 HIKARI Service Architecture Consortium. All rights reserved.
4.1.3 Required functions and issues

The every function required for providing this service is listed here. An accounting function or a settling function are omitted in the list, because various operations for those functions can be considered, and it is difficult to decide the detailed operations of the functions.

- Application service (distance learning service) provider (1/3)

<table>
<thead>
<tr>
<th>Function</th>
<th>Outline</th>
<th>Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registered user management (User management server)</td>
<td>This function provides the user registration screen (service application screen)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This function authenticates a user and checks the registered user</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• User authentication method</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The simplest system of checking using a password/user ID is considered as a user authentication method. To reduce a workload of the user, it is also considered that information about identification is assigned to a terminal itself and the user is identified based on the information. (However, strictly speaking, this does not indicate authentication of a person). Using an authentication means, such as an IC card, is considered in case the terminal has an I/F. This can specify the user without depending on the terminal.</td>
<td></td>
</tr>
<tr>
<td>Class participant management (Distance learning server)</td>
<td>This function provides the class attendance registration screen (presents attendance conditions), and manages the instruction(class) start time</td>
<td></td>
</tr>
<tr>
<td>Session connection request/acceptance function [SIP agent function] (MCU)</td>
<td>This function requests connection to and from a class participant.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Basic connection system</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SIP and H.323 are considered as basic connection systems. In this document, the SIP is selected. For details, refer to Section 9-5 of the SA Technical Report.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Session setting procedure</td>
<td></td>
</tr>
<tr>
<td></td>
<td>In this document, a session is established from the MCU to each terminal, but even the session setting from the terminal side will not cause a problem.</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Outline</td>
<td>Issue</td>
</tr>
<tr>
<td>----------</td>
<td>---------</td>
<td>-------</td>
</tr>
</tbody>
</table>
| Stream synthesis/delivery function (MCU) | The stream of videos and audios from the terminal of each participant is synthesized in the form previously defined according to a service and sent to the specified terminal. | • Video synthesis  
A video synthesis technique of a low delay and a light load is required, but there is no method that is well-known and assured.  
• Generation of a sound field  
In mixing of audios from many speakers, audience cannot recognize each speaker. Accordingly, such low delay audio synthesis that produces a sound field is required. There is no method that is well-known and assured.  
• API for constructing a service  
It is desirable to have the API which can be used easily for constructing service logic. There is no API that is well-known and assured.  
• MCU load  
Because an asymmetrical service (display content may also differ every terminal. A flow of a stream may not be symmetrical) is considered, streams of the maximum number of terminals multiplied by two are transferred. Accordingly, the MCU load is high. Another method of performing intercommunication between terminals without MCU is out of consideration because it is difficult to establish a processing system that obtains matching between the terminals when the streams are realized. For the moment, communication among up to ten terminals or communications with limited amount of terminals which sends a stream is considered realistic |
| Stream switching (MCU) | This function receives an instruction from a distance learning server in accordance with the switching of a scene in a service and changes a flow of audio/video streams. | • Stream switching control (MCU)  
No protocol that controls stream switching is well-known and assured. This document proposes the realization over HTTP. |
| File delivery function (Distance learning server) | This function delivers a teaching materials file in accordance with a request from a terminal. | • Copyright protection  
General issues concerning copyright protection also should be considered here. |
<table>
<thead>
<tr>
<th>Function</th>
<th>Outline</th>
<th>Issue</th>
</tr>
</thead>
</table>
| Service state management (Distance learning server) | This function manages a service state, moves to a new state according to a request from each terminal, and instructs stream switching. | - Service state management method  
Because each terminal state needs to be managed, a load is high if the state is managed in a centralized manner by the distance learning server. A method of decentralized management is out of consideration because the transferring of states are too complicated and hard to be realized. For the moment, communication among up to ten terminals or communications with limited amount of terminals which sends a stream is considered realistic |
| Security function | This function performs encryption for preventing wiretapping of a video/audio that is applied in a session  
- Encryption installation method  
There is a great variety of systems realizing an encryption function for preventing wiretapping, depending on using which section and which layer to realize the function. The encryption may be realized between a terminal and the MCU, or between the session management provider and the MCU. Also, the realization can be made on various layers, such as IP layers and application layer (encryption of a video/audio content itself). | |

---

<table>
<thead>
<tr>
<th>Function</th>
<th>Outline</th>
<th>Issue</th>
</tr>
</thead>
</table>
| Service menu offer | This function offers a service menu to a user. | - Compatibility of the protocol related to the menu display  
Because each browser provider extends HTML, JavaScript, CSS, and DHTML independently, they have no compatibility in a fine part. |
### Session management provider

<table>
<thead>
<tr>
<th>Function</th>
<th>Outline</th>
<th>Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Connection (session) management (Session management server)</strong></td>
<td>This function sets up/manages a session between the MCU and each terminal and between the terminals in accordance with an instruction from the distance learning server.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Basic connection function</td>
<td>The issue is on the definition of basic connection function in a communication service. In this service, the issue is investigated through the SIP. Refer to Section 9-5 of the SA Technical Report.</td>
</tr>
<tr>
<td><strong>Connection destination address specification (Directory service)</strong></td>
<td>This function specifies a user network address registered in the distance learning server.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Directory implementation format</td>
<td>No directory format is specified in this document.</td>
</tr>
<tr>
<td><strong>Security function</strong></td>
<td>This function performs encryption for preventing wiretapping of a video/audio that is applied in a session</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Encryption installation method</td>
<td>There is a great variety of systems realizing an encryption function for preventing wiretapping, depending on using which section and which layer to realize the function. The encryption may be realized between a terminal and the MCU, or between the session management provider and the MCU.</td>
</tr>
</tbody>
</table>
### Content delivery provider

<table>
<thead>
<tr>
<th>Function</th>
<th>Outline</th>
<th>Issue</th>
</tr>
</thead>
</table>
| **Content delivery function**  | This function delivers both videos and audios of multiple users  
The function delivers the optimum teaching materials content to multiple users.  
The function manages a multicast address (when the number of trainees is large).                                                                                                                                                                                                                                           | • **Multicast use**  
Multicast may be used when the number of trainees is exceeding the number of terminals supported by MCU.                                                                                                                                                                                                                                 |
| **Bi-directional video communication pass setting** | This function sets a communication path for bi-directional video communication.                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                                                                                 |
| **Cache/mirror management function** | This function previously stores a teaching materials content for smoothly delivering teaching materials.                                                                                                                                                                                                                                                                                          | • **API for previous registration**  
No API for registering teaching materials in a cache/mirror previously and intentionally is well-known and assured.                                                                                                                                                                                                                             |
| **Service network management function** | This function monitors/controls the network operation and network state of a delivery network.                                                                                                                                                                                                                                                                                               |                                                                                                                                                                                                                                                                                                                                 |
| **Band control and management function** | This function secures a band required for sending a video/audio to a delivery network.                                                                                                                                                                                                                                                                                                      | • **Bandwidth assurance protocol**  
There is a system that secures a band using the RSVP. However, to realize it on the practical HIKARI Service platform, there is an issue in the aspect of facilities and operation (evaluation of the possibility when the absolute value of the band is insufficient and processing and a billing system then). |

Copyright (C) 2001-2002 HIKARI Service Architecture Consortium. All rights reserved.
### Terminal (1/2)

<table>
<thead>
<tr>
<th>Function</th>
<th>Outline</th>
<th>Issue</th>
</tr>
</thead>
</table>
| Content playing (Stream decoding) | This function receives the encoded video/audio stream and decodes it using a decoding application. The function properly displays the decoded video streams on the screen using a Web browser and a dedicated application. The function outputs the decoded audio streams. | • Decoding delay  
The MPEG-4 formats for realization of low delay are investigated. For details, refer to Section 9-2 of the SA Technical Report. There is no established low delay decoding system using these formats.  
• Synchronization between a video and an audio  
The synchronization between a video and an audio is necessary. The extent of the synchronization depends on a service. In many TV telephone-like services (combination of a conversation of human beings and an expressions of their faces), there is no significant incongruity up to 200 msec. In most cases, priority is given to a low delay of an audio. |
|                                  | • Audio/video format                                                   |                                                                                                 |
|                                  | - Video: MPEG-4, Simple Profile Level 3                                |                                                                                                 |
|                                  |   (When a band of 384 k or more is required, Core Profile Level 3 Subset, up to 2 M) |                                                                                                 |
|                                  | - Audio: PCM, ADPCM, MPEG-1 Audio Layer 1/2, MP3                      |                                                                                                 |
|                                  | - Multiplex: RFC 3016                                                  |                                                                                                 |
|                                  | • Future standardization                                              |                                                                                                 |
|                                  | The MPEG-4 protocols suitable for low delay/high image quality are established and may be spread and used widely in future. For details, refer to Section 9-2 of the SA Technical Report. |                                                                                                 |
| Content encoding                 | This function captures and digitizes a video of a video input means, such as a camera. The function captures and digitizes an audio from an audio input means. The function encodes the captured video/audio. | • Capture/encoding delay  
The MPEG-4 formats for realization of a low delay is investigated. For details, refer to the SA Technical Report. No low delay decoding system is established using their formats. If the formats are converted by the MCU, they need not be the same formats as the decoding formats.  
• Flicker noise  
A flicker appears on the screen when illumination is performed by a fluorescent lamp. It is necessary to use flicker less camera or flicker depression software. |
| Content sending                  | This function sends the coded video/audio.                            |                                                                                                 |
| Connection/disconnection to network | This function executes connection and disconnection to a network. This service is premised on the always-on connection of a fixed address. | • System of connection/disconnection to network  
When many terminals perform the always-on connection at a fixed address, the address may be insufficient for IPv4. |
<table>
<thead>
<tr>
<th>Function</th>
<th>Outline</th>
<th>Issue</th>
</tr>
</thead>
</table>
| Security                        | This function performs encryption for preventing wiretapping of a video/audio. | • Encryption for preventing wiretapping  
There is a great variety of systems realizing an encryption function, depending on using which section and which layer to realize the function. The encryption may be realized between a terminal and the session management provider or between a terminal and the MCU. |
| Terminal security                | This function prevents unauthorized invasion into terminal equipment. | • Firewall function  
A firewall function is necessary for preventing unauthorized invasion into terminal equipment. However, a video/audio stream needs to reach a terminal. Several systems are proposed and implemented as a gateway function (Proxy function) in the application levels, such as RTSP, SIP, and HTTP. |
| Session connection request/acceptance function [SIP UA function] (MCU) | This function sets up a session from the MCU and a session to another terminal. | • Basic connection system  
SIP and H.323 are considered as basic connection systems. In this document, the SIP is selected. For details, refer to Section 9-5 of the SA Technical Report. |
| Buffering function              | This function absorbs the jitter and delay of video signals.            | • Setting size of buffer  
High quality (guarantee of a fixed reproduction speed and high image quality) and a reduced delay have a tradeoff relationship. The priority will be determined depending on an application. In most cases, priority is given to a low delay. |
| Terminal event sending function  | This function sends an event, such as pressing a button and a key on the terminal, to the instruction management server. | • No protocol for sending and receiving an event is established. This service assumes an HTTP-based protocol. |
| Screen switching function        | This function changes the video display position on the screen, the display position of a button, and teaching materials, by user operation or according to an event from the instruction management server. They are performed basically from the terminal of an instructor. The instructor requires more screens and buttons than a trainee. It is assumed that the instructor uses a high function terminal than the trainee. | • No protocol for sending and receiving an event is established. |
4.2 Interface in Bi-directional Video Communication Service

The specific interface requirements of a bi-directional video communication service are listed in accordance with the following flow:

1. Registration on a distance learning service
2. Posting of teaching materials
3. Class announcement
4. Class attendance registration
5. Opening the class
6. Instruction function (trainee’s intermediate participation/early leaving/speech)
7. Video display switching
8. Inter-trainee conversation (group learning)
9. Closing the class

4.2.1 Service registration function

4.2.1.1 Outline of function

This function allows a service executor (instructor) to register instruction time/the number of trainees in the distance learning server in the distance learning service. This is realized on the Web-based menu that the distance learning service function provider prepares. The distance learning executor sets a class condition from the menu screen and receives an ID and a password for utilization.

4.2.1.2 Service registration function sequence diagram

The sequence of this function is shown in Fig. 4.2.1.
4.2.1.3 Service sequence and applicable protocol set

The protocol set used in this function is shown in Table 4.2.1.

Table 4.2.1 Protocol Set Applied to Service Registration Function

<table>
<thead>
<tr>
<th>Processing</th>
<th>Interface function</th>
<th>Data item</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial menu display</td>
<td>(1) Initial menu request</td>
<td>Initial menu URL</td>
<td>Upper layer: HTTP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>User identification information</td>
<td>Lower layer: TCP</td>
</tr>
<tr>
<td></td>
<td>(2) Initial menu display</td>
<td>Initial menu data</td>
<td></td>
</tr>
<tr>
<td>Service registration screen</td>
<td>(3) Service registration</td>
<td>Service registration screen URL</td>
<td></td>
</tr>
<tr>
<td>request</td>
<td>request</td>
<td>User identification information</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(4) Service registration</td>
<td>Service registration screen (HTML)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>screen display</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service registration condition</td>
<td>(5) Service registration</td>
<td>Utilization condition input screen URL</td>
<td></td>
</tr>
<tr>
<td>input</td>
<td>request</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(6) Utilization condition</td>
<td>Utilization condition input screen (HTML)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>input screen display</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID, password notification</td>
<td>(7) Utilization condition</td>
<td>Utilization condition</td>
<td>Upper layer: HTTPS</td>
</tr>
<tr>
<td></td>
<td>input</td>
<td></td>
<td>Lower layer: TCP</td>
</tr>
<tr>
<td></td>
<td>(8) ID, password</td>
<td>ID, password display screen (HTML)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>notification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Registration completion</td>
<td>(9) Registration acknowledgement</td>
<td>Acknowledgement notification</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(10) Registration completion</td>
<td>Completion screen (HTML)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>notification</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.2.1.4 Applicable condition

This service registration allows the executor who uses the distance learning function to register for using the function. Basically, a utilization condition ought to be requested. The sequence is not fixed to the above one and there are various ways. Further, another information necessary for the application can also be provided.

This service provides the above function by an HTML browser. Further, the HTTPS should be used in the place where secrecy should be kept.

(1) User authentication

An ID and a password are used here as the simplest user authentication when a service is used. However, if there is another authentication means, it can be used.

(2) Billing information

It can be fully assumed that some billing is performed along with a service registration. However, the transfer of the billing information is not shown explicitly in the above sequence.

(3) HTML content

It is desirable that the version of HTML used here should be displayed by various browsers (version 3.2 or equivalent).

(4) Portal function

The first initial menu in the sequence diagram can be menus which the service portal provides or ones which the screen of “distance learning service” provides that is reached as a result of tracing a link from the service portal.
4.2.2 Posting teaching materials function

4.2.2.1 Outline of function

The executor (instructor) of the distance learning service can register teaching materials used in a class after he or she registers service utilization. The teaching materials are registered in the distance learning service provider (teaching materials server). The executor confirms an applicant from the ID and password obtained in the course of a service registration, at registration of teaching materials. The teaching materials are handled as files and the contents are not examined whether they are text, images, or videos.

4.2.2.2 Posting teaching materials function sequence diagram

The sequence diagram of this function is shown in Fig. 4.2.2.

![Figure 4.2.2 Posting Teaching Materials Function Sequence](Copyright (C) 2001-2002 HIKARI Service Architecture Consortium. All rights reserved.)
4.2.2.3 Service sequence and applicable protocol set

The protocol set used in this function is shown in Fig. 4.2.2.

Table 4.2.2 Protocol Set Applied to Posting Teaching Materials Function

<table>
<thead>
<tr>
<th>Processing</th>
<th>Interface function</th>
<th>Data item</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teaching materials registration screen request</td>
<td>(1) Teaching materials registration screen request</td>
<td>ID, password acknowledgement screen URL User identification information</td>
<td></td>
</tr>
<tr>
<td>Teaching materials registration screen display</td>
<td>(2) ID, password acknowledgement screen display</td>
<td>ID, password acknowledgement screen (HTML)</td>
<td>Upper layer: HTTPS Lower layer: TCP</td>
</tr>
<tr>
<td>Teaching materials registration screen display</td>
<td>(3) ID, password input</td>
<td>ID, password</td>
<td></td>
</tr>
<tr>
<td>Teaching materials registration screen display</td>
<td>(4) Teaching materials registration screen display</td>
<td>Teaching materials registration screen (HTML)</td>
<td></td>
</tr>
<tr>
<td>Teaching materials registration</td>
<td>(5) Teaching materials registration</td>
<td>Teaching materials file (file transfer)</td>
<td></td>
</tr>
<tr>
<td>Teaching materials registration completion notification</td>
<td>(6) Teaching materials registration completion notification</td>
<td>Teaching materials registration completion screen (HTML)</td>
<td></td>
</tr>
</tbody>
</table>

4.2.2.4 Applicable condition

Various patterns are considered in posting teaching materials in the teaching materials server and are not fixed to the above sequence. For example, there may also be a registration method that depends on the data type of the teaching materials. When the teaching materials exist in any place other than the terminal of an executor, they may also be transferred from there. Here, to register the teaching materials, the browser-based menu should be displayed.

Further, the teaching materials may also be delivered directly from the terminal of an executor to the terminal of a trainee in the class. In this case, the teaching materials are not registered. Further, only when the teaching materials are used in the class, the teaching materials are registered.

(1) User authentication
   An ID and a password are used here as the simplest user authentication when a service is used. However, if there is another authentication means, it can be used.

(2) Billing information
   It can be fully assumed that some billing is performed along with posting the teaching materials. However, the transfer of the billing information is not shown explicitly in the above sequence.

(3) HTML content
   It is desirable that the version of HTML used here should be displayed by various browsers (version 3.2 or equivalent).

(4) Portal function
   The first initial menu in the sequence diagram can be menus which the service portal provides or ones which the screen of “distance learning service” provides that is reached as a result of tracing a link from the service portal.
4.2.3 Class announcement function

4.2.3.1 Outline of function

The class announcement function registers the screen for announcing a class and previously stores teaching materials in the content delivery provider preparing for delivery.

4.2.3.2 Class announcement function sequence diagram

The sequence diagram of this function is shown in Fig. 4.2.3.

Fig. 4.2.3 Class Announcement Function Sequence

Repeat items 5 to 8 as the need arises.
4.2.3.3 Service sequence and applicable protocol set

The protocol set used in this function is shown in Table 4.2.3.

<table>
<thead>
<tr>
<th>Processing</th>
<th>Interface function</th>
<th>Data item</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class announcement screen registration</td>
<td>(1) Class announcement request</td>
<td>Class data</td>
<td>Upper layer: HTTP</td>
</tr>
<tr>
<td></td>
<td>(2) Class announcement request response</td>
<td>Response</td>
<td>Lower layer: TCP</td>
</tr>
<tr>
<td></td>
<td>(3) Class announcement screen data sending</td>
<td>Class announcement data</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(4) Class announcement screen registration completion</td>
<td>Acknowledgement response</td>
<td></td>
</tr>
<tr>
<td>Teaching materials delivery registration</td>
<td>(5) Teaching materials delivery registration request</td>
<td>Teaching materials data</td>
<td>Upper layer: HTTPS</td>
</tr>
<tr>
<td></td>
<td>(6) Teaching materials delivery registration request response</td>
<td>Acknowledgement response</td>
<td>Lower layer: TCP</td>
</tr>
<tr>
<td></td>
<td>(7) Teaching materials transfer</td>
<td>Teaching materials file (transfer)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(8) Instructional transfer response</td>
<td>Acknowledgement response</td>
<td></td>
</tr>
</tbody>
</table>

4.2.3.4 Applicable condition

This service assumes that any class announcement processing will be performed on the HTTP base and the procedure should basically be performed on the HTTP base. When secrecy is required, the handling of the above sequence ought to be performed using the HTTPS.

(1) Portal function
In this service, class announcement is performed using the portal function and the screen registration is performed from the distance learning service provider. However, also concerning these items, various patterns are considered. The distance learning service provider itself may greatly provide the class announcement screen. In this case, the distance learning executor may create the announcement screen. There can also be various forms of the recruitment screen itself.

Even if the class announcement is registered in the service portal, the registration sequence is not fixed and various protocols are considered.

(2) Previous delivery of teaching materials
In this document, it is assumed that teaching materials is registered previously in the delivery function, but this operation is not necessarily required. The teaching materials may be delivered directly from the distance learning service provider. If the teaching materials are registered in the delivery function, there is no fixed registration procedure and the procedure may greatly differ from the above procedure.

(3) Teaching materials delivery protocol
In this document, teaching materials should be delivered using the HTTP/HTTPS, but the FTP may also be used.
4.2.4 Class attendance registration function

4.2.4.1 Outline of function

This function allows a trainee to register for attending a class. The registration is performed from the Web-based menu screen that the distance learning service provider prepares. A user registers by entering class attendance conditions.

4.2.4.2 Class attendance registration sequence diagram

The sequence of this function is shown in Fig. 4.2.4.

---

Fig. 4.2.4 Class Attendance Registration Function Sequence
4.2.4.3 Service sequence and applicable protocol set

The protocol set used in this function is shown in Table 4.2.4.

<table>
<thead>
<tr>
<th>Processing</th>
<th>Interface function</th>
<th>Data item</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class announcement screen</td>
<td>(1) Class announcement screen request</td>
<td>Class announcement screen URL</td>
<td>Upper layer: HTTP</td>
</tr>
<tr>
<td></td>
<td>(2) Class announcement screen display</td>
<td>Class announcement screen data (HTML)</td>
<td>Lower layer: TCP</td>
</tr>
<tr>
<td>Class attendance registration screen</td>
<td>(3) Class attendance registration screen request</td>
<td>Class attendance registration screen URL</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>User identification data</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(4) Class attendance registration screen display</td>
<td>Class attendance registration screen data (HTML)</td>
<td></td>
</tr>
<tr>
<td>Class attendance registration</td>
<td>(5) Class attendance registration</td>
<td>Trainee registration data</td>
<td>Upper layer: HTTPS</td>
</tr>
<tr>
<td></td>
<td>(6) Class attendance registration</td>
<td>Acknowledgement screen data (HTML)</td>
<td>Lower layer: TCP</td>
</tr>
</tbody>
</table>

4.2.4.4 Applicable condition

Because exceedingly various types are considered in the data required for the distance learning application, the application processing may be performed by splitting the screen into multiple screens. There may be such format in which one class is selected from multiple classes. In this case, the format will not necessarily conform to the above sequence. This service assumes that such distance learning registration is made in the browser base. Trainee data ought to be transferred using the HTTPS to keep secrecy.

(1) Portal function

In this service, a trainee traces a link from the class announcement screen of the service portal and registers from the distance learning application screen that the distance learning service provider provides. However, the service portal need not to provide the class announcement screen.

(2) Billing information

It can be fully assumed that some billing is performed along with posting the teaching materials. However, the transfer of the billing information is not shown explicitly in the above sequence.

(3) HTML content

It is desirable that the version of HTML used here should be able to be displayed by various browsers (version 3.2 or equivalent).

Further, because it cannot be expected that the terminal of a trainee has a high function, it is desirable that the distance learning registration screen uses a low resolution and assumes poor input equipment (without a keyboard).
4.2.5 Opening class function

4.2.5.1 Outline of function

This function sets a session of video and audio streams between the distance learning service function, the distance learning execution function, and the trainee function. The inside of the Opening class function is subdivided into two phases of “video and audio quality negotiation processing” and “session setting processing”. Each processing is performed between the distance learning service function and terminal (distance learning execution function) and between the distance learning service function and terminal (trainee function).
4.2.5.2 Opening class function sequence diagram

The sequence of this function is shown in Figs. 4.2.5 (a) and 4.2.5 (b).

Note: Repeat steps (1) to (10) for the number of trainees.

Fig. 4.2.5 (a) Opening Class Function Sequence
(Distance learning service function – Distance learning execution function)

Fig. 4.2.5 (b) Opening Class Function Sequence
(Distance learning service function – Trainee function)
4.2.5.3 Service sequence and applicable protocol set

The protocol used in this function is shown in Table 4.2.5.

<table>
<thead>
<tr>
<th>Processing</th>
<th>Interface function</th>
<th>Data item</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video and audio</td>
<td>(1) INVITE</td>
<td>Command code (INVITE), connection destination address, connection source</td>
<td>Upper layer: SP</td>
</tr>
<tr>
<td>quality negotiation</td>
<td></td>
<td>address, video quality condition, audio quality condition, ID, and password</td>
<td>Lower layer: UDP</td>
</tr>
<tr>
<td></td>
<td>(2) INVITE</td>
<td>Command code (INVITE), connection destination address, connection source</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>address, video quality condition, and audio quality control</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(3) Progress</td>
<td>Destination address, status (Progress)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(4) Progress</td>
<td>Destination address, status (Progress)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(5) Provisional</td>
<td>Destination address, status (PRACK)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reliable Responses</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acknowledgement</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(6) Precondition</td>
<td>Destination address, status (COMET)</td>
<td>Upper layer: SIP</td>
</tr>
<tr>
<td></td>
<td>met</td>
<td></td>
<td>Lower layer: UDP</td>
</tr>
<tr>
<td></td>
<td>(7) Progress</td>
<td>Destination address, status (Progress)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>:COMET</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(8) Progress</td>
<td>Destination address, status (Progress)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>:INVITE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(9) Acknowledgement</td>
<td>Destination address, status (ACK)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(10) Visual</td>
<td>Video and audio</td>
<td>Upper layer: RTP</td>
</tr>
<tr>
<td></td>
<td>Communication</td>
<td></td>
<td>Lower layer: UDP</td>
</tr>
</tbody>
</table>

4.2.5.4 Applicable condition

(1) Opening class timing

There are many alternatives of setting the timing for opening a class such as an arrival of instruction start time and a request from a distance learning executor (instructor). In this sequence, the arrival of the instruction start time is used for timing as one of examples.

For using the request from the distance learning executor as the timing, the transfer of information using a Web server must be added between the distance learning execution function and the distance learning service function.

(2) Stream session initiation

There are two methods to realize the session initiation. One is to initiate from the distance learning executor or a trainee and another is to initiate from the distance learning service function. In this sequence, the method of initiation from the distance learning service function is shown as the example. In the method of initiation from the terminal, there is a difference that the source and destination (orientation of an arrow mark) of a protocol becomes reverse as a connection request is issued from the terminal side. But there is no difference in the protocol to be used and the content of information.

As for the number of sessions to be initiated, one way is to select a session from multiple sessions and display it in a terminal and another way is to synthesize multiple sessions from each terminal into one stream and display it in a terminal. In the latter way, synthesizing in the terminal needs a high function terminal. Accordingly, in this sequence, a system enables the use of a low function terminal, that is, a method of obtaining one session, which is synthesized in the MCU of the distance learning service function and is sent to the terminal, is shown as an example.
(3) Acquisition timing of address
The address registered when a distance learning executor makes registration for the use of a distance learning service or the address registered when a trainee registers class attendance registration is used as a terminal address. The address of a terminal is converted to an IP address using ENUM and DNS. If the trainee must temporarily attend distance learning with the address different from the registered one, the session management function enables the registration of temporal address by the SIP register function or the distance learning service function enables it by application screen of Web server.

(4) Selection of a video format and an audio format
The selected video format and audio format are as shown below.
- Video : MPEG-4
- Audio : MPEG Audio Layer1 or Layer2
- Multiplex : IETF RFC 3012
As the MPEG-4 profile, SimpleProfile@L3 is promising. For the applicability of another profile, refer to Section 9-2 of the SA Technical Report.
When a format that requires a broadband is selected, bandwidth reservation may be required in the sequence.

(5) Order of session initiation
The session between the distance learning service function and the trainee function and the session between the distance learning service function and the distance learning execution function may be initiated regardless of the sequence.

(6) Start of the distance learning application program
In this sequence, the application program used in distance learning is automatically invoked when a terminal is powered on. It may be manually invoked by a command.
4.2.6 Trainee’s intermediate participation/early leaving/speech function

4.2.6.1 Outline of function

This function initiates a session of video and audio streams between the distance learning service function and the trainee function when a participation application was made from a trainee to the class that was already started (intermediate participation), releases the session of the video and audio streams established between the distance learning service function and the trainee function when the trainee desired leaving in the course of class (early leaving), and sends the video of the trainee to another trainee when the trainee makes a speech in the course of class (speech).

(1) Intermediate participation

The function consists of “intermediate participation application”, “video and audio quality negotiation”, and “session initiation”. The “video and audio quality negotiation” and “session initiation” are equal to those in the Opening class function between the distance learning service function and the trainee function.

(2) Early leaving

The function consists of “early leaving application” and “session releasing”. The “session releasing” is equal to that in the closing class function between the distance learning service function and the trainee function.

(3) Speech

This function consists of “speech application” and “speaker’s video sending”. The “speaker’s video sending” is equal to that in the video switching function.

4.2.6.2 Trainee’s intermediate participation/intermediate secession/speech function sequence diagram

The sequence of this function is shown in Figs. 4.2.6 (a), 4.2.6 (b), and 4.2.6 (c).

In the intermediate participation, the sequence of changing class attendance place is shown. There are various systems to change the class attendance place. And the adoption of other systems is not limited.
### Secession Application

Christopher

<table>
<thead>
<tr>
<th>Content Delivery Function</th>
<th>Instruction information request</th>
<th>HTTPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instruction information display</td>
<td>HTTPS</td>
<td>&quot;Secession&quot;</td>
</tr>
<tr>
<td>Approval inquiry</td>
<td>HTTPS</td>
<td>&quot;OK&quot;</td>
</tr>
<tr>
<td>Completion notification</td>
<td>HTTPS</td>
<td></td>
</tr>
</tbody>
</table>

**Fig. 4.2.6 (b) Trainee’s Early Leaving Function Sequence**

### Speech Application

Christopher

<table>
<thead>
<tr>
<th>Content Delivery Function</th>
<th>Instruction information request</th>
<th>HTTPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instruction information display</td>
<td>HTTPS</td>
<td>&quot;Speech&quot;</td>
</tr>
<tr>
<td>Approval inquiry</td>
<td>HTTPS</td>
<td>&quot;OK&quot;</td>
</tr>
<tr>
<td>Completion notification</td>
<td>HTTPS</td>
<td></td>
</tr>
</tbody>
</table>

**Fig. 4.2.6 (c) Trainee’s Speech Function Sequence**
## 4.2.6.3 Service sequence and applicable protocol set

The protocol set used in this function is shown in Table 4.2.6.

### Table 4.2.6 Protocol Applied to Intermediate Participation/Early Secession/ Speech Functions for Trainee

<table>
<thead>
<tr>
<th>Processing</th>
<th>Interface function</th>
<th>Data item</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(1) Instruction information request</td>
<td>Instruction information request URL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(2) Instruction information display</td>
<td>Instruction information screen (HTML)</td>
<td>Upper layer: HTTPS Lower layer: TCP</td>
</tr>
<tr>
<td>Intermediate</td>
<td>(3) “Participation”</td>
<td>Selected menu (Participation)</td>
<td></td>
</tr>
<tr>
<td>participation</td>
<td>(4) Condition inquiry</td>
<td>Condition input screen (HTML)</td>
<td></td>
</tr>
<tr>
<td>application</td>
<td>(5) Condition notification</td>
<td>Input content (user ID, password, and address)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(6) Completion notification</td>
<td>Completion screen (HTML)</td>
<td></td>
</tr>
<tr>
<td>Early secession</td>
<td>(1) Instruction information request</td>
<td>Instruction information request URL</td>
<td></td>
</tr>
<tr>
<td>application</td>
<td>(2) Instruction information display</td>
<td>Instruction information screen (HTML)</td>
<td>Upper layer: HTTPS Lower layer: TCP</td>
</tr>
<tr>
<td></td>
<td>(3) “Secession”</td>
<td>Selected menu (secession)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(4) Approval inquiry</td>
<td>Notification information (information about early secession and the seceder) Selection condition (HTML)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(5) “OK”</td>
<td>Selected menu (OK)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(6) Completion notification</td>
<td>Completion screen (HTML)</td>
<td></td>
</tr>
<tr>
<td>Speech</td>
<td>(1) Instruction information request</td>
<td>Instruction information request URL</td>
<td></td>
</tr>
<tr>
<td>application</td>
<td>(2) Instruction information display</td>
<td>Instruction information screen (HTML)</td>
<td>Upper layer: HTTPS Lower layer: TCP</td>
</tr>
<tr>
<td></td>
<td>(3) “Speech”</td>
<td>Selected menu (speech)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(4) Approval inquiry</td>
<td>Notification information (information about speech and speaker) Selection condition (HTML)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(5) “OK”</td>
<td>Selected menu (OK)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(6) Completion notification</td>
<td>Completion screen (HTML)</td>
<td></td>
</tr>
</tbody>
</table>

## 4.2.6.4 Applicable condition

1. **Speech intention display method**
   As a means of notifying an instructor (distance learning executor) of the trainee’s request to speak, there are two methods. One uses a camera and a microphone of trainee’s terminal (for example, raising hand) and another uses the operation of a distance learning application (client function). In this sequence, the latter one is shown as an example. In the former method, the request is notified as the video and audio from the trainee. And all trainees must be displayed on the screen of instructor (distance learning executor)’s terminal.

2. **Speaker’s video sending method**
   In a MCU, the video of a trainee who makes a speech is synthesized with other streams based on the instruction from an instructor (distance learning executor). Then a MCU sends the synthesized stream to other trainees.
4.2.7 Video display switching function

4.2.7.1 Outline of function

This function switches a video sent to a trainee’s display based on the instruction from an instructor (distance learning executor). The function consists of “video display switching instruction” and “switching condition specification”.

4.2.7.2 Video display switching function sequence diagram

The sequence of this function is shown in Fig. 4.2.7.

Fig. 4.2.7 Video Display Switching Function Sequence
### 4.2.7.3 Service sequence and applicable protocol set

The protocol set used in this function is shown in Table 4.2.7.

#### Table 4.2.7 Protocol Set Applied to Video Display Switching Function

<table>
<thead>
<tr>
<th>Processing</th>
<th>Interface function</th>
<th>Data item</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video display switching instruction</td>
<td>(1) Instruction information request</td>
<td>Instruction information request URL</td>
<td>Upper layer: HTTPS Lower layer: TCP</td>
</tr>
<tr>
<td></td>
<td>(2) Instruction information display</td>
<td>Instruction information screen (HTML)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(3) “Video display switching”</td>
<td>Selected menu (Video switching)</td>
<td></td>
</tr>
<tr>
<td>Switching condition specification</td>
<td>(4) Condition inquiry</td>
<td>Condition input screen (HTML)</td>
<td>Upper layer: HTTPS Lower layer: TCP</td>
</tr>
<tr>
<td></td>
<td>(5) “Trainee”</td>
<td>Selected menu (Trainee)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(6) Condition inquiry</td>
<td>Condition input screen (HTML)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(7) Condition notification</td>
<td>Condition information (“Christopher”, “displayed to all trainees”)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(8) Completion notification</td>
<td>Completion screen (HTML)</td>
<td></td>
</tr>
</tbody>
</table>

### 4.2.7.4 Applicable condition

(1) Video switching method

There are several methods of switching scene, such as switching from the scene of an instructor to the scene of a blackboard. One of them is switching an object (for photography) of one camera, and another is selecting a camera from multiple cameras. In this sequence, the latter method is shown as an example. Further, the video switching adopts a method of changing a video to be synthesized by the MCU, but the adoption of other methods will not be prevented.
4.2.8 Inter-trainee conversation function (group learning function)

4.2.8.1 Outline of function
The function divides trainees into multiple groups and enables conversation between the trainees in the group. The function consists of "group learning support" and "group learning condition specification".

4.2.8.2 Inter-trainee conversation function sequence diagram
The sequence of this function is shown in Fig. 4.2.8.

![Fig. 4.2.8 Inter-trainee Conversation Function Sequence]

4.2.8.3 Service sequence and applicable protocol
The protocol set used in this function is shown in Table 4.2.8.

<table>
<thead>
<tr>
<th>Processing</th>
<th>Interface function</th>
<th>Data item</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group learning support</td>
<td>(1) Instruction information request</td>
<td>Instruction information request URL</td>
<td>Upper layer: HTTPS</td>
</tr>
<tr>
<td></td>
<td>(2) Instruction information display</td>
<td>Instruction information screen (HTML)</td>
<td>Lower layer: TCP</td>
</tr>
<tr>
<td></td>
<td>(3) &quot;Group&quot;</td>
<td>Selected menu (Group)</td>
<td></td>
</tr>
<tr>
<td>Group learning condition specification</td>
<td>(4) Condition inquiry</td>
<td>Condition input screen (HTML)</td>
<td>Upper layer: HTTPS</td>
</tr>
<tr>
<td></td>
<td>(5) Condition notification</td>
<td>Notification information (Information about trainee)</td>
<td>Lower layer: TCP</td>
</tr>
<tr>
<td></td>
<td>(6) Completion notification</td>
<td>Completion screen (HTML)</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.2.8 Protocol Set Applied to Inter-trainee Conversation Function
4.2.8.4 Applicable condition

(1) Grouping timing
   An instructor (distance learning executor) instructs grouping in the course of a class.

(2) Inter-trainee conversation implementation system
   There are various systems to realize this function that limits the conversation between trainees within the range of group members. In this sequence, the method of switching a video to be synthesized by the MCU is shown as an example.
4.2.9 Closing class function

4.2.9.1 Outline of function

This function releases sessions of video and audio streams established among the distance learning service function, the distance learning execution function, and the trainee function. The function consists of “closing class instruction” and “session releasing”.

4.2.9.2 Instruction end function sequence diagram

The sequence of this function is shown in Fig. 4.2.9.

Fig. 4.2.9 Closing Class Function Sequence
4.2.9.3 Service sequence and applicable protocol set

The protocol set used in this function is shown in Table 4.2.9.

<table>
<thead>
<tr>
<th>Processing</th>
<th>Interface function</th>
<th>Data item</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Closing class instruction</td>
<td>(1) Instruction information request</td>
<td>Instruction information request URL</td>
<td>Upper layer: HTTPS Lower layer: TCP</td>
</tr>
<tr>
<td></td>
<td>(2) Instruction information display</td>
<td>Instruction information screen (HTML)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(3) “Instruction end”</td>
<td>Selected menu (Closing class)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(4) Completion display</td>
<td>Completion screen (HTML)</td>
<td></td>
</tr>
<tr>
<td>Session releasing</td>
<td>(5) Resource release request</td>
<td>Command (BYE) and destination address</td>
<td>Upper layer: SIP Lower layer: UDP</td>
</tr>
<tr>
<td></td>
<td>(6) Resource release response</td>
<td>Destination address and status (ACK)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(7) Resource release request</td>
<td>Command (BYE) and destination address</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(8) Resource release response</td>
<td>Destination address and status (ACK)</td>
<td></td>
</tr>
</tbody>
</table>

4.2.9.4 Applicable condition

(1) Closing class timing

An arrival of the closing time or a request from a distance learning executor (instructor) may terminate the class. In this sequence, a request from an instructor terminates the class. If the arrival of the closing time terminates the class, the distance learning service function should display announcement of closing before releasing sessions.

(2) Session releasing

There are two methods to realize the session releasing. One is to release from a distance learning executor or a trainee and another is to release from the distance learning service function. In this sequence, the method of releasing from the distance learning service function is shown as an example. In the method of releasing from the terminal, there is a difference that the source and destination (orientation of an arrow mark) of a protocol becomes reverse as a releasing request is issued from the terminal, but there is no difference in the protocol to be used and the content of information.

The session between the distance learning service function and the trainee function and the session between the distance learning service function and the distance learning execution function can be released regardless of the sequence.
Appendix

RTSP Basic Specifications
A.1 Introduction ......................................................................................................................... A.1
  A.1.1 Positioning of This Appendix ......................................................................................... A.1
  A.1.2 Relationship with RFC ................................................................................................. A.1
  A.1.3 Explanation of Phrase .................................................................................................. A.1
A.2 Basic Protocol of This Appendix ......................................................................................... A.3
  A.2.1 Purpose ........................................................................................................................ A.3
  A.2.2 Basic Connection .......................................................................................................... A.3
    A.2.2.1 Prevention of timeout ............................................................................................. A.3
  A.2.3 Media ............................................................................................................................ A.3
  A.2.4 System .......................................................................................................................... A.3
    A.2.4.1 Server/client common requirements ......................................................................... A.4
    A.2.4.2 Server requirements ............................................................................................... A.4
    A.2.4.3 Client requirements ................................................................................................. A.4
  A.2.5 Service Mode ............................................................................................................... A.4
  A.2.6 Basic Function ............................................................................................................. A.4
    A.2.6.1 Flow of basic sequence .......................................................................................... A.4
    A.2.6.2 State transition ....................................................................................................... A.5
A.3 Time-shift ............................................................................................................................ A.6
  A.3.1 Change in Jumpable Range by Live Relay Continuation .............................................. A.6
  A.3.2 Change in Jumpable Range by Limit of Server Storage Function ................................ A.6
    A.3.2.1 Remarks on restart of pause .................................................................................. A.7
A.4 Function Implementation ..................................................................................................... A.8
  A.4.1 Playback Start .............................................................................................................. A.8
  A.4.2 End (End of content) .................................................................................................. A.8
  A.4.3 End (Termination by a user) ....................................................................................... A.9
  A.4.4 Pause and Restart ....................................................................................................... A.9
  A.4.5 Jump Playback ............................................................................................................ A.10
  A.4.6 Fast forward Playback/Rewind Playback .................................................................. A.12
  A.4.7 Method Acknowledgement (Server performance Acknowledgement) ...................... A.12
A.5 RTSP Message .................................................................................................................... A.13
  A.5.1 List of Methods ....................................................................................................... A.13
  A.5.2 List of Headers ........................................................................................................ A.13
    A.5.3 Header Common Items ............................................................................................. A.15
      A.5.3.1 Handling of each line ......................................................................................... A.14
      A.5.3.2 Require header and Unsupported header ........................................................... A.14
    A.5.4 Response Status Code ............................................................................................. A.15
A.6 Detailed RTSP Message Format ......................................................................................... A.16
  A.6.1 OPTIONS .................................................................................................................. A.16
    A.6.1.1 Request message format ...................................................................................... A.16
    A.6.1.2 Response message format ................................................................................ A.16
  A.6.2 DESCRIBE ................................................................................................................ A.16
    A.6.2.1 Request message format ...................................................................................... A.16
    A.6.2.2 Response message format ................................................................................ A.16
  A.6.3 SETUP ..................................................................................................................... A.17
    A.6.3.1 Request message format ...................................................................................... A.17
    A.6.3.2 Response message format ................................................................................ A.17
  A.6.4 PLAY ......................................................................................................................... A.18
    A.6.4.1 Request message format ...................................................................................... A.18
    A.6.4.2 Response message format ................................................................................ A.18
  A.6.5 PAUSE ..................................................................................................................... A.20
    A.6.5.1 Request message format ...................................................................................... A.20
    A.6.5.2 Response message format ................................................................................ A.20
  A.6.6 TEARDOWN ............................................................................................................. A.21
    A.6.6.1 Request message format ...................................................................................... A.21
A.6.6.2 Response message format ................................................................. A.21
A.6.7 ANNOUNCE ...................................................................................... A.22
A.6.7.1 Request format ................................................................................ A.22
A.6.7.2 Response message format ............................................................... A.22

A.7 SDP Description .................................................................................... A.23
A.7.1 Presentation Information ...................................................................... A.23
A.7.2 Control Information ............................................................................. A.23
  A.7.2.1 Content provicable time ............................................................... A.23
  A.7.2.2 Content provicable range ........................................................... A.23
  A.7.2.3 Enabled time of Time-shift .......................................................... A.24
A.7.3 Decoding Information ......................................................................... A.24
  A.7.3.1 Coding (codec) type ................................................................. A.24
  A.7.3.2 Transfer rate ............................................................................ A.24
A.7.4 Sample SDP Messages ....................................................................... A.25
  A.7.4.1 VoD sample .............................................................................. A.25
  A.7.4.2 Sample of Live with time-shift ................................................... A.25

A.8 Sample Sequences ................................................................................ A.26
A.8.1 Playback Start ................................................................................... A.26
A.8.2 Jump Playback ................................................................................... A.27
A.8.3 Fast forward 1 (When fast forward is available) ................................ A.28
A.8.4 Fast forward 2 (When fast forward is unavailable) ............................. A.29
A.8.5 End 1 (Content end) .......................................................................... A.29
A.8.6 End 2 (End by user) .......................................................................... A.30
A.1 Introduction

A.1.1 Positioning of This Appendix

This appendix specifies the basic interface specification of the stream delivery control between a delivery server and a terminal in the content delivery. The System Interface Requirements on HIKARI Service Network Architecture is positioned as the uppermost base document. This appendix specifies the basic specification nearer to implementation concerning the stream delivery control (refer to Chapter 3 of the System Interface Requirements on HIKARI Service Network Architecture) in the content delivery function. This appendix assumes utilization of a broadband network that is a feature of the HIKARI service platform, but it does not limit an available bandwidth and a protocol in particular and specifies the protocol that can be applied to various operating environment.

In the System Interface Requirements on HIKARI Service Network Architecture, the RTSP that is an RFC as a protocol that realizes the stream delivery control and special operation control in the video delivery service. This is because media data of various formats is handled and an open specification is aimed at. Concerning the RTSP, there is an issue (refer to Section 9-4 of the SA Technical Report) that various interpretations can exist only if they conform to the range specified in the RFC 2326 and the interconnectivity between various vendors is extremely difficult. Accordingly, in this appendix, a protocol on the RTSP message format description is prepared to ensure the interconnectivity between the vendors. Specifically, implementation methods of each function in the stream delivery control, RTSP message common item/detailed contents, SDP descriptions, and sequence samples are specified.

Based on the implementation realized using this appendix as a guideline, we expect that many video delivery servers and terminals of multi-vendors can use the video delivery service mutually and a HIKARI service platform that is easy to use by users will be realized. Besides, this document is more understandable if it is read referring to the RFC 2326.

A.1.2 Relationship with RFC

The protocol specified in this appendix has a relationship with high-level compatibility of the RTSP specified in the RFC 2326 and the messages and headers that are essential in the RFC 2326 must be specified implicitly in the protocol of this appendix. When a description is omitted in this appendix, it conforms to the description of the RFC 2326. SDP in this specification has a relationship with high-level compatibility of the SDP specified in the RFC 2327. When a description of SDP is omitted in this appendix, it conforms to the description of the RFC 2327.

Note that, in this appendix, a video stream data format is not specified.

A.1.3 Explanation of Phrase

In the message format described in this appendix, the description enclosed by a square has the following meanings.

- **Boldface** or equally wide **Courier**: Fixed character used as is.
- **Italic**: Modified to a value actually used.

Besides, the character enclosed by parentheses means that it can be omitted. The description of the part not enclosed by a square conforms to Backus-Naur form (BNF) described in the RFC 2326 and RFC 2616.

Further, the phrase described in this appendix has the following meanings.

**Live**

This service provides the video distribution from a live hall, such as a concert hall and a stadium via the HIKARI service platform in real time. The time-shift function, such as a pause and fast forward/rewinding, can be realized by viewing a video stored in a server and a client terminal.

**RFC (Request For Comments)**

The RFC is a standard recommendation document recognized as a publicly opened specification draft by the ISO (International Standardization Organization) and issued by Internet Architecture Bureau (IAB). The RTSP described in this appendix is specified in the RFC 2326 and the SDP is specified in the RFC 2327. Unless specified otherwise in this appendix, the RFC indicates the RFC 2326.
VoD (Video on Demand)

Contents and the length of a content can not be modified while the service is being provided. This service provides the content stored in the server from the desirable time of a user.

Web server

This Web server installs HTTP/1.0 or later.

Web client

This Web client is terminal software that acquires HTML data from the Web server using HTTP/1.0 or later version.

Persistent Connection

The persistent connection is a command of sending and receiving mode that uses a TCP. It indicates a method in which multiple types of command sending and receiving are performed in the single TCP connection without disconnecting the command sending and receiving. It is also called permanent connection. It is the antonym for Transient Connection.

Client

Terminal software that interchanges the control information between the media server and the terminal using the RTSP is simply described as a "client" in this appendix unless otherwise specified. Usually, this client is installed on the same terminal in which a Web browser is installed and connected to a media server based on the information acquired from the Web browser.

Non-aggregate operation

When a content consists of multiple media, such as an audio and a video, a control URL is provided every media and the media is controlled individually. It is the antonym of aggregate operation.

Server

The media delivery server that installs the RTSP specified in this appendix is simply described as a "server" unless specified otherwise.

Services

The VoD and Live services are targeted. Usually, because both the VoD and Live services are provided by a single server, the services cannot be specified from the control URL. To specify it, the operation of the client needs to be changed based on the supplementary information described in every content.

Aggregate operation

When a content consists of multiple media, such as an audio and a video, all media is controlled by performing control for a control URL.

Time-shift

In Live, special playback, such as pause, jump playback, rewinding, and fast forward, are performed by receiving a live video and storing it simultaneously in a storage unit for the server and the client. In this appendix, we assume that time-shift is performed by the usage of storage in the server.

Transient Connection

The transient connection is a command of sending and receiving mode that uses a TCP. It indicates a method in which the TCP connection is performed by repeating connection/disconnection for every command. It is the antonym for Persistent Connection.

"Mandatory" (sending side)

Concerning the sending side, "mandatory" indicates that data must be added and sent to a sequence for a method and the method for a header respectively. Further, concerning the receiving side, it indicates that "mandatory" must be installed. When the method and header are received, the receiving side needs to interpret the meaning.
A.2 Basic Protocol of This Appendix

A.2.1 Purpose

The protocol specified in this appendix is used for the stream delivery control of a video content from a server to a client. A first purpose of this appendix is to provide the interconnectivity between different vendors, and this appendix specifies the basic protocol of the stream delivery control.

A.2.2 Basic Connection

The protocol specified in this appendix is performed in the one-to-one correspondence between the sending and receiving sides of a video. IP must be used in the communication of a network layer and TCP must be used in the communication of a transport layer. The RTSP specified here targets only the persistent connection. That is, while a video content is being received, the operation must be done without disconnecting the TCP session. When the TCP session is disconnected, the server should stop the stream delivery.

Further, when the server sets a timeout and there is no request from the client for a fixed period of time, the connection with the client must be closed and the stream must be stopped.

A.2.2.1 Prevention of timeout

The client should prevent the disconnection from the server by sending a method or a heart-beat signal before a timeout occurs. Fig. A.2.2.1 shows an example in which a heart-beat signal is sent once in every 30 seconds. The server may also stop the sending of a stream if an effective method or heart-beat signal is not sent from the client in the specified time even when the stream is sent. The heart-beat signal must be the simplest text CR+LF(0x0d,0x0a) and this text should be sent to the server within the fixed time. The server may not to return a response for this text. The client can check whether the session is still connected by receiving TCP Ack.

A.2.3 Media

The media handled in this appendix is only the content of that can be handled by the aggregate operation and we does not handle the content that requires non-aggregate operation. That is, each media, such as a video and an audio, handles only the content multiplexed in a stream. In such content, all media can be controlled by performing control for a control URL without controlling each media individually.

Further, MPEG-2 TS is assumed as the main media. However, the protocol specified in this appendix does not limit the type of the media and may be applied to another media.

A.2.4 System

The system specified in this appendix assumes the configuration consisting of the server and client shown in Fig. A.2.4.2. The operation in the configuration in which the gateway that relays data between the server and the client is not specified.

Copyright (C) 2001-2002 HIKARI Service Architecture Consortium. All rights reserved.
A.2.4.1 Server/client common requirements

An IP and a TCP must be installed.

A.2.4.2 Server requirements

The server in this appendix must satisfy the following conditions.

- A connection request must be received from the client at a specific TCP port (usually, number 554 specified in the RFC is used).
- Multiple continuous requests of the same session must be accepted.

A.2.4.3 Client requirements

The client in this appendix must satisfy the following condition.

- A connection request must be sent to a specific TCP port of the server (usually, number 554 specified in the RFC is used).

A.2.5 Service Mode

The video distribution services handled in this appendix are VoD and Live. However, in Live the mode in which time-shift operation is enabled in accordance with an instruction from the client is also handled by simultaneously recording data in the storage of the server and sending it to the client.

A.2.6 Basic Function

A.2.6.1 Flow of basic sequence

The comparison between the necessary functions and the sequence in VoD, Live (time-shift enabled), and Live (time-shift disabled) is shown in Table A.2.6.1. It describes four cases of playback start, under playback, end, and others as a flow from the playback start time to viewing end time. Note that special playback, such as a pause and fast forward, do not exist in the Live that disables time-shift.

<table>
<thead>
<tr>
<th>Basic function</th>
<th>VoD Live (time-shift enabled)</th>
<th>Live (time-shift disabled)</th>
<th>Basic sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Playback start</strong></td>
<td>Acquisition of content info.</td>
<td>Acquisition of content info.</td>
<td>Playback start (Section A.4.1)</td>
</tr>
<tr>
<td></td>
<td>Preparation of content</td>
<td>Preparation of content</td>
<td>DESCRIBE+SETUP+PLAY</td>
</tr>
<tr>
<td></td>
<td>Playback start</td>
<td>Playback start</td>
<td></td>
</tr>
<tr>
<td><strong>Under playback</strong></td>
<td>Special playback</td>
<td></td>
<td>Pause/playback (Section A.4.4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PAUSE+PLAY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Jump playback (Section A.4.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PAUSE+PLAY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Fast forward playback/rewinding</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>playback (Section A.4.6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PAUSE+PLAY</td>
</tr>
<tr>
<td><strong>End</strong></td>
<td>End</td>
<td>End</td>
<td>End of content (Section A.4.2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ANNOUNCE+TEARDOWN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>End by user (Section A.4.3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PAUSE+TEARDOWN</td>
</tr>
<tr>
<td><strong>Others</strong></td>
<td>Acknowledgement of server</td>
<td>Acknowledgement of server</td>
<td>Acknowledgement of method (Section</td>
</tr>
<tr>
<td></td>
<td>performance</td>
<td>performance</td>
<td>A.4.7)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OPTIONS</td>
</tr>
</tbody>
</table>

Copyright (C) 2001-2002 HIKARI Service Architecture Consortium. All rights reserved.
A.2.6.2 State transition

The state transition specified in Appendix A of the RFC is illustrated below corresponding to the method described in the preceding section.

(1) Client state

Fig. A.2.6.1 Client State Transition

(2) Server state

Fig. A.2.6.2 Server State Transition
A.3 Time-shift

In Live, there is a system that enables special playback, such as pause and rewinding playback by simultaneously storing and distributing data by the server. The special playback performed using the server storage function in this manner is called time-shift. Live with time-shift is identical to VoD as the function, but there are several points to which attention must be paid on the implementation.

A.3.1 Change in Jumpable Range by Live Relay Continuation

Because the length of a content is predetermined for VoD, the time range for special operation is fixed. For example, when the content of which the content length is for ten minutes is viewed, the jumpable range is fixed as 0 to 600 seconds. On the other hand, for Live that can be time-shifted, the client that was accessed after \( t_1 \) seconds from the distribution start enabling a special operation between 0 and \( t_1 \) seconds in the beginning, but enables the special operation between 0 and \( (t_1 + \Delta t) \) seconds after \( \Delta t \) seconds. The range for special operation is increased with time goes by. Accordingly, the client should always know the time range in which the special operation is enabled. For this purpose, the client must calculate the time range for special operation by storing \( t_1 \) and counting \( \Delta t \).

\[ \text{Fig. A.3.1.1 Change of Maximum Value of time range for special operation} \]

A.3.2 Change in Jumpable Range by Limit of Server Storage Function

In Live, content distribution may be continued without determining the distribution end time. On the contrary, to perform time-shift function, data must be stored in the server. However, because the server storage capacity is limited, time-shift is disabled if delivery is continuing for a long time. In the server, to prevent this, time-shift is always enabled for a fixed time by using a fixed storage area repetitively. If the maximum storage time determined previously by the server is \( t_r \), special playback is enabled from the higher value of either \( t_2 - t_r \) or 0 to \( t_2 \) in the client. For \( t_r = 0 \) in particular, the Live in which time-shift is disabled is used.

To install this function, the client must recalculate the special operable range by recording \( t_r \) in addition to the function described in the preceding section. \( t_r \) must be provided to the client using the recordtime attribute of the SDP (for details, refer to Section A.7.2.3).
A.3.2.1 Remarks on restart of pause

When the maximum storage time \( t_r \) is set in the server, if a pause state lasts for a long time and the server providable range is exceeded, and if the video content is restarted from the same time position as just before pause, an error may occur. Accordingly, when the video content is restarted, whether the server providable range is exceeded restart position or not, must be monitored.
A.4 Function Implementation

This section shows the sequence for implementing each function using a method and describes the minimum necessary headers at that time.

A.4.1 Playback Start

The playback start sequence is as follows.

1. The client must send DESCRIBE to the server and acquires information about a video content of which playback must be started. In response to this, the server must return the content information using SDP.

2. The client must send SETUP to the server and specifies a transport parameter for receiving video content data.

3. The client must send PLAY to the server.

4. The server that received a PLAY request must start the sending of the video content data in accordance with a value of the Range header.

---

### Fig. A.4.1.1 Playback Start Sequence
A.4.2 End (End of content)

The end sequence allows the client to perform end processing without a user’s instruction when the content sending is terminated from the server. The sequence is as follows.

1. The server must send the end of a stream by sending ANNOUNCE to the client when the data sent to the client is finished.
2. When the client that received an end notification by ANNOUNCE, the client may analyze the end reason and may send TEARDOWN to the server.

The client may perform PLAY operation in the Ready state.

![Diagram](image)

**Fig. A.4.2.1 End Sequence (without a user’s instruction)**

A.4.3 End (Termination by a user)

The end sequence when a request from a user was made in the course of content playback is as follows.

1. The client must send PAUSE to the server in order to request pause.
2. The server must stop sending of a video and must wait for the next message that is sent in the pause state (however, when nothing is sent in the specified time, the same processing after TEARDOWN was received must be performed).
3. The client should send TEARDOWN to the server and requests the stop of a video.

However, the server must stop correctly even if TEARDOWN is received suddenly while data is being sent (Playing state). Further, because the content may be finished at the same time with a PAUSE request from the client and ANNOUNCE may be issued, the server must process PAUSE request even in the Ready state.
A.4.4 Pause and Restart

The pause and restart sequences are as follows.

1. The client must send PAUSE to the server in order to request the pause of a video content.
2. The server must stop sending of a video content and must wait for the next message that is sent in the pause state (however, when nothing is sent within the timeout described in Section A.2.6.1, the same processing after TEARDOWN was received must be performed).
3. When restarting from pause state, the client must send PLAY to the server in order to request restarting of sending a video content that is paused (be sure to add a Range header. The server operation is not guaranteed to the PLAY request that is not added).
4. The server that received a PLAY request must start the data sending of a video content in accordance with a value of the Range header.

Copyright (C) 2001-2002 HIKARI Service Architecture Consortium. All rights reserved.
**A.4.5 Jump Playback**

The jump playback sequence is as follows.

1. The client must send PAUSE to the server in order to request to pause a video content.
2. The server must stop the data sending of a video content and must wait for the next message that is sent in pause state (however, when nothing is sent within the timeout described in Section A.2.6.1, the same processing after TEARDOWN was received must be performed).
3. The server that received PAUSE must stop sending of a content and must enter the Ready state.
4. When restarting from pause state, the client must send PLAY to the server in order to request sending a video content paused (be sure to add a Range header. The server operation is not guaranteed to the PLAY request that is not added).
5. After the Range of PLAY sent from the client is investigated, sending of a stream must be started from a new position if the Range is within the actual range.
A.4.6 Fast forward Playback/Rewind Playback

The fast forward/rewinding playback sequences are as follows.

1. The client must send PAUSE to the server in order to request to pause a video content that is currently received.

2. The server must stop the data sending of a video content and must wait for the next message that is sent in the pause state (however, when nothing is sent within the timeout time, the same processing after TEARDOWN was received must be performed).

3. When restarting from pause state, the client must send PLAY to the server in order to request sending a video content paused (be sure to add a Range header. The server operation is not guaranteed to the PLAY request that is not added).

4. A Scale value of PLAY sent from the client must be investigated and compared with a Scale value that is installed and can be sent. Subsequently, the sending of stream data must be started based on the Scale value that can be sent and a Range value.

Fig. A.6.1 Fast forward Playback/Rewinding Playback Sequence

A.4.7 Method Acknowledgement (Server performance Acknowledgement)

The client must inventory a method (depends on the server) prepared at the server side and must issue the sequence that is suitable for it. The method is inventoried using the OPTIONS method. The method inventory sequence is as follows.

Fig. A.7.1 Method Acknowledgement Sequence
A.5 RTSP Message

This section describes a list of the methods and headers used in the RTSP, and the header assigned to each message in common.

A.5.1 List of Methods

The methods that must be interpreted by the server and the client in this specification are listed respectively in the following table. The items other than “mandatory” items are all options, and need not be interpreted.

<table>
<thead>
<tr>
<th>Method</th>
<th>Server</th>
<th>Client</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESCRIBE</td>
<td>Mandatory</td>
<td></td>
</tr>
<tr>
<td>ANNOUNCE</td>
<td></td>
<td>Mandatory</td>
</tr>
<tr>
<td>GET_PARAMETER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPTIONS</td>
<td>Mandatory</td>
<td></td>
</tr>
<tr>
<td>PAUSE</td>
<td></td>
<td>Mandatory</td>
</tr>
<tr>
<td>PLAY</td>
<td></td>
<td>Mandatory</td>
</tr>
<tr>
<td>RECORD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REDIRECT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SETUP</td>
<td></td>
<td>Mandatory</td>
</tr>
<tr>
<td>SET_PARAMETER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEARDOWN</td>
<td></td>
<td>Mandatory</td>
</tr>
</tbody>
</table>

A.5.2 List of Headers

The headers that must be implemented in the server and the headers added anew in this specification are shown in Table A.5.2.1. Here, regardless of mandatory items in the RFC ([12] Header Field Definitions), only the headers listed here are “mandatory”. The headers those are specified in the RFC and are not described here are options. For example, the User-Agent header and the Server header are options.

The headers described here must be interpreted by the server, but they may not be used in a request from the client. However, because ANNOUNCE may be sent from the server, it must be interpreted by the client. The meaning of each column in A.5.2.1 is as follows.

- **Header**: Name of a header
- **Type**: R → The relevant header is used in the Request message.
  - r → The relevant header is used in the Response message.
  - R → Both the Request and Response messages are used.
- **Method**: Name of a method that uses a header (all → Used in all methods)
## A.5.3 Header Common Items

### A.5.3.1 Handling of each line

The end of each line of the format described in the next section and later must terminate with CR+LF. Further, the boundary with an entity, such as a header and SDP must be indicated by CR+LF. For example, when the server sends a response followed by SDP as the entity, the boundary must be Last header+(CR+LF)+(CR+LF)+SDP itself. The end of an entity depends on the entity and Content-Length.

If there is no entity, a message must terminate with CR+LF. That is, the message must terminate with Last header+(CR+LF) itself. When the server receives such an incomplete message as Last header+(CR+LF), timeout process may be performed after a wait for a fixed time (when (CF+LF) is received anew within the fixed time, the process must be performed as a complete message).

### A.5.3.2 Require header and Unsupported header

The Require header must be used to acknowledge an extended header created by extending this appendix. The Require header may be added to any method. If an acknowledge request is received from the client using the Require header and it proves that any extended header is not implemented for the relevant method in the server, the server must send an error (551 Option not supported) with the unsupported header name as an argument of Unsupported. When the server receives an unknown header that is not specified with Require in a request, the server must simply ignore the relevant header and must continue processing, and the server must not return an error (551 Option not supported). The following example is the response when the extended header added to a request of the client is not installed in the server as x-playAtOnce.

#### Example 1) When Require is specified

<table>
<thead>
<tr>
<th>Client → Server</th>
<th>Server → Client</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLAY rtsp://svr.com/foo/test.mpg RTSP/1.0 CSeq: 302 Session: 47112344 Require: x-playAtOnce x-playAtOnce:</td>
<td>RTSP/1.0 551 Option not supported CSeq: 302 Session: 47112344 Unsupported: x-playAtOnce</td>
</tr>
</tbody>
</table>

#### Example 2) When Require is not specified

<table>
<thead>
<tr>
<th>Client → Server</th>
<th>Server → Client</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLAY rtsp://svr.com/foo/test.mpg RTSP/1.0 CSeq: 302 Session: 47112344 x-playAtOnce:</td>
<td>RTSP/1.0 200 OK CSeq: 302 Session: 47112344</td>
</tr>
</tbody>
</table>
A.5.4 Response Status Code

The response status code conforms to the RFC status code ([7.1.1] Status Code and Reason Phrase). If a request notation format from the client is incorrect, an error (400 Bad Request) must be returned unless specified otherwise.
A.6 Detailed RTSP Message Format

A detailed RTSP message format is described. An actual parameter should be inserted to the part described in Italic of the format. Further, rtsp_URL conforms to the RFC([3.2] RTSP URL).

A.6.1 OPTIONS

OPTIONS acquire a list of the methods installed in the server. The client may make an inquiry in any condition and the server must always interpret the request. The state of the client or server must not change according to this method.

A.6.1.1 Request message format

```
OPTIONS rtsp_URL RTSP/1.0
CSeq: CSeq_Number
```

(1) CSeq
CSeq must conform to the RFC([12.7] CSeq).

A.6.1.2 Response message format

```
RTSP/1.0 Status_Code Reason_Phrase
CSeq: CSeq_Number
Public: (List of the set of methods supported by the server)
VersionSupport: Protocol_Name/Version
```

(1) Public
Public must conform to the RFC([12.28] Public).

(2) VersionSupport
This header is not described in the RFC, but in this specification the version number of this specification should be posted. The format is as follows.

```
Protocol_Name = "HSAC" | extension
Version = major-version "." minor-version
major-version = 1(DIGIT)
minor-version = 1(DIGIT)
```

Besides, the version that conforms to this specification must be
```
protocol-name = HSAC, MajorVersion = 1, MinorVersion = 0.
```
Example)
```
VersionSupport: HSAC/1.0
```

A.6.2 DESCRIBE

DESCRIBE acquires information about a video content specified in a message. The client may make an inquiry in any condition and the server must always interpret the request. The state of the client and the server must not change according to this method.

A.6.2.1 Request message format

```
DESCRIBE rtsp_URL RTSP/1.0
CSeq: CSeq_Number
```

An SDP may also be requested explicitly by adding the Accept header. If the Accept header is not included, the client and server must operate assuming that application/sdp was specified in the Accept header.

Copyright (C) 2001-2002 HIKARI Service Architecture Consortium. All rights reserved.
A.6.2.2 Response message format

<table>
<thead>
<tr>
<th>RTSP/1.0 Status_Code Reason_Phrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSeq: CSeq_Number</td>
</tr>
<tr>
<td>Content-Type: (Type of the message body)</td>
</tr>
<tr>
<td>Content-Length: (Length of the message body)</td>
</tr>
</tbody>
</table>

1. Content-Type
   Content-Type must conform to RFC([12.16] Content-Type). Besides, when DESCRIBE that does not include the Accept header is received, the client and the server must operate assuming that application/sdp is specified in the Accept header. In this case, a response, which includes application/sdp for ContentType, must be returned.

2. Content-Length
   Content-Length must conform to the RFC ([12.14] Content-Length). The SDP is described in Section A.7.

A.6.3 SETUP

SETUP specifies transport parameters used when data is sent and received between the server and the client.

A.6.3.1 Request message format

<table>
<thead>
<tr>
<th>SETUP rtsp_URL RTSP/1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSeq: CSeq_Number</td>
</tr>
<tr>
<td>Transport: Transport_Parameter</td>
</tr>
</tbody>
</table>

1. Transport
   “transport-protocol” format is defined as an extension of the RFC([12.39] Transport) as shown below.

   **Table A.6.3.1 Extension of Transport-protocol**

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>“RTP”</td>
<td>RTP that conforms to the RFC 1889</td>
</tr>
<tr>
<td>“RAW”</td>
<td>Media data itself in which the RTP payload format is not used.</td>
</tr>
</tbody>
</table>

   “profile” format is also defined as an extension of the RFC as shown below.

   **Table A.6.3.2 Extension of Profile**

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>“AVP”</td>
<td>An AVP payload is used in the RTP.</td>
</tr>
<tr>
<td>“MP2T”</td>
<td>Used when media is MPEG-2 TS.</td>
</tr>
</tbody>
</table>

   The Transport header is defined here as shown below.

   ```
   Transport_Parameter= ("RTP/AVP/UDP" | "RAW/MP2T/UDP") parameter
   parameter= unicast-parameter | multicast-parameter
   unicast-parameter = "unicast" ; "destination" = address
   ; "client_port" = port [ "-" port ]
   multicast-parameter = "multicast" ; "destination" = address
   address = host
   port = 1-5(DIGIT)
   ```

   **Example 1)**
   ```
   Transport:
   RAW/MP2T/UDP;unicast;destination=192.168.0.2;client_port=6971
   ```

   **Example 2)**
   ```
   Transport:
   RTP/AVP/UDP;unicast;destination=192.168.0.2;client_port=6970-6971
   ```

Copyright (C) 2001-2002 HIKARI Service Architecture Consortium. All rights reserved.
A.6.3.2 Response message format

<table>
<thead>
<tr>
<th>RTSP/1.0 Status_Code Reason_Phrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSeq: CSeq_Number</td>
</tr>
<tr>
<td>Session: Session_ID</td>
</tr>
<tr>
<td>Transport: Transport_Parameter</td>
</tr>
</tbody>
</table>

(1) Session
Session must conform to the RFC([12.37] Session). When timeout is set, it must be effective in the Ready state and the Playing state. Once the state moves to the Init state, the set value must be reset to the default value, 60 seconds.

Session = "Session" ":" Session_ID [ ; "timeout" "=" delta-seconds ]
Example) Session: 123456789;timeout=90

(2) Transport
The Transport header is defined here as shown below.

Transport_Parameter = ("RTP/AVP/UDP" | "RAW/MP2T/UDP") parameter
parameter = unicast-parameter | multicast-parameter
unicast-parameter = "unicast" ";" "destination" "=" address ";"
uni_param
uni_param = "client_port" "=" port [ "-" port ]
            ;" "server_port" "=" port [ "-" port ] ;" "bitrate" "=" bitrate
multicast-parameter = "multicast" ";" "destination" "=" address ";"
mlt_param
mlt_param = "port" "=" port [ "-" port ] [ ";" "ttl" "=" ttl ]
            ;" "bitrate" "=" bitrate
address = host
port = 1*5(DIGIT)
bitrate = 1*(DIGIT)
ttl = 1*3(DIGIT)

bitrate is a content coding rate. When the server receives a request of unsupported transport, method the server may return an error (461, unsupported transport).

Example 1)
Transport: RAW/MP2T/UDP;unicast;destination=192.168.0.2;client_port=6971;
server_port=1234;bitrate=6000000

Example 2)
Transport: RTP/AVP/UDP;unicast;destination=192.168.0.2;client_port=6970-6971;
server_port=1234;bitrate=4000000

A.6.4 PLAY

PLAY instructs to start playing a video content specified in a message or restarts a playback of the video content that is paused.

A.6.4.1 Request message format

<table>
<thead>
<tr>
<th>PLAY rtsp_URL RTSP/1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSeq: CSeq_Number</td>
</tr>
<tr>
<td>Session: Session_ID</td>
</tr>
<tr>
<td>Range: (npt=)Start_Time-(End_Time)</td>
</tr>
<tr>
<td>Scale: Scale_Value</td>
</tr>
</tbody>
</table>
(1) Range

Range must conform to the RFC([12.29] Range). Only NPT (Normal Play Time) is mandatory, and “npt=” can be omitted only when NPT is used. “now” must be used only in the case of Live and this value may be used in order to specify the latest position (that is, live video) of a content. The server must return an error (457 Invalid range) to the client when “now” is used in VoD. If the specified value is beyond the server provision range, the server must return the error (457 Invalid range) to the client regardless of VoD or Live. Further, in this specification, “current”, “beginning”, and “end” must be implemented. They have the following meaning respectively.
### Table A.6.4.1 Extended Keyword of Range

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>current</strong></td>
<td>Same time when a video content is stopped with PAUSE. “current” is valid only when current is added to PLAY after PAUSE. In other cases, values are not guaranteed.</td>
</tr>
<tr>
<td><strong>beginning</strong></td>
<td>Identical with the initial time of a content the server can provide. For VoD, always replaced to 0 by the server. For the Live that can be time-shifted, this is replaced by 0 or the initial time the server can provide. For the Live that cannot be time-shifted, values are not guaranteed.</td>
</tr>
<tr>
<td><strong>end</strong></td>
<td>Identical with the end time of content the server can provide. For the Live that can be time-shifted, this is replaced by the current NPT and interpreted in the server. For the Live that cannot be time-shifted, values are not guaranteed.</td>
</tr>
</tbody>
</table>

When a header other than “Range: now-” is specified though time-shift is disabled for Live, the server must return an error (457 Invalid range) to the client. The provided service and the applicable range of an extended keyword including now are shown in the following table.

### Table A.6.4.2 Applicable Range of Extended Keyword

<table>
<thead>
<tr>
<th></th>
<th>VoD</th>
<th>Live</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>current</strong></td>
<td>OK: Substituted for a pause position in the server</td>
<td>OK: Substituted for a pause position in the server</td>
</tr>
<tr>
<td></td>
<td>NG: Error 457</td>
<td>NG: Error 457</td>
</tr>
<tr>
<td><strong>beginning</strong></td>
<td>OK: Substituted for 0 in the server</td>
<td>OK: Substituted for the top position that can be provided in the server</td>
</tr>
<tr>
<td></td>
<td>NG: Error 457</td>
<td>NG: Error 457</td>
</tr>
<tr>
<td><strong>end</strong></td>
<td>OK: Substituted for the final position that can be provided in the server</td>
<td>OK: Converted to the latest position in the server</td>
</tr>
<tr>
<td></td>
<td>OK: Converted to the latest position in the server. When Live is finished, error 457 is returned.</td>
<td>OK: Converted to the latest position in the server</td>
</tr>
<tr>
<td><strong>now</strong></td>
<td>NG: Error 457</td>
<td>OK: Converted to the latest position in the server</td>
</tr>
</tbody>
</table>

Example) Range: 0-
Example) Range: now- ; Only for Live

(2) Scale
Scale must conform to the RFC[12.34] Scale. When fast forward playback/rewinding playback is performed, Scale must be added for specifying a playback speed.

When Scale is omitted, the same operation when Scale: 1.0 is added must be performed.

### A.6.4.2 Response message format

```plaintext
RTSP/1.0 Status_Code Reason_Phrase
CSeq: CSeq_Number
Session: Session_ID
Range: Start_Time-(End_Time)
Scale: Scale_Value (only for Fast Forward and Rewind Play Response)
```

(1) Range
Even if Range specifies now and the above keyword, the server must return a request from the client in actual time. When the request from the client is of an NPT format, the response must use the NPT format. Further, even if a request is made with an extended keyword “now” shown in A.6.4.2, the request must be returned converting it to an actual value.
(2) Scale

A value that can be sent actually should be returned. In the process in which the value that can be sent is calculated, the nearest value that is installed in the server should be returned. The client must check the value from the server and must confirm that the speed of the data being sent actually is a requested speed.

Example) When only x1.0 speed, x2.0 speed, and x4.0 speed are available in the server and a request from the client is Scale:2.5, the server should return Scale:2.0 and should send a stream of x2.0 speed. The client must check the response from the server, and should not expect the data of x2.5 speed as requested.

If the obtained result is Scale:1.0, that is, the same result as normal playback, an error (406 Not Acceptable) is returned without returning Scale: 1.0. When the fast forward function is not implemented, an error(400 Bad Request) must be returned.

A.6.5 PAUSE

playbackPause a video content playing specified in a message.

A.6.5.1 Request message format

<table>
<thead>
<tr>
<th>PAUSE rtsp_URL RTSP/1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSeq: CSeq_Number</td>
</tr>
<tr>
<td>Session: Session_ID</td>
</tr>
</tbody>
</table>

A.6.5.2 Response message format

<table>
<thead>
<tr>
<th>RTSP/1.0 Status_Code Reason_Phrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSeq: CSeq_Number</td>
</tr>
<tr>
<td>Session: Session_ID</td>
</tr>
<tr>
<td>Range: Pause_Time</td>
</tr>
</tbody>
</table>

(1) Range

Range returns the time when the stream sending is stopped.

Example) When a stream is sent and stopped after 5.6 seconds.

Range: 5.60

However, the client may receive a header of a “Range: x-y” format. This case must be handled in the same manner as “Range: x”.

A.6.6 TEARDOWN

The service of a video content specified in a message is terminated.

A.6.6.1 Request message format

<table>
<thead>
<tr>
<th>TEARDOWN rtsp_URL RTSP/1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSeq: CSeq_Number</td>
</tr>
<tr>
<td>Session: Session_ID</td>
</tr>
</tbody>
</table>

A.6.6.2 Response message format

<table>
<thead>
<tr>
<th>RTSP/1.0 Status_Code Reason_Phrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSeq: CSeq_Number</td>
</tr>
<tr>
<td>Session: Session_ID</td>
</tr>
</tbody>
</table>
A.6.7 ANNOUNCE

ANNOUNCE is a method only posted from the server to the client and posts the reason of a stream stop to the client.

A.6.7.1 Request format

<table>
<thead>
<tr>
<th>ANNOUNCE</th>
<th>rtsp_URL</th>
<th>RTSP/1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSeq:</td>
<td>CSeq_Number</td>
<td></td>
</tr>
<tr>
<td>Session:</td>
<td>Session_ID</td>
<td></td>
</tr>
<tr>
<td>Notice:</td>
<td>Event_Code Event_Phrase</td>
<td></td>
</tr>
</tbody>
</table>

Because this method is sent asynchronously from the server, the client may receive ANNOUNCE even when a response of another method is being awaited. For example, when PAUSE is sent and a response is expected, ANNOUNCE may be received.

1) CSeq

Because this method is sent asynchronously from the server, CSeq may be the same CSeq of a request from the client. Therefore, CSeq in the message from the server must be started anew from 1 every session.

2) Notice

Notice describes the reason of a stream stop. The format is as follows.

Event_Code = 4DIGT
Event_Phrase = ＊<TEXT, excluding CR, LF, ＊>

Event_Code must be any one of codes shown in the following table.

<table>
<thead>
<tr>
<th>Code</th>
<th>Phrase</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1103</td>
<td>Playout Stalled</td>
<td>Temporarily stopped</td>
</tr>
<tr>
<td>1104</td>
<td>Playout Resumed</td>
<td></td>
</tr>
<tr>
<td>2101</td>
<td>End-of-Stream Reached</td>
<td>Content terminated</td>
</tr>
<tr>
<td>2103</td>
<td>Transition</td>
<td>In transition</td>
</tr>
<tr>
<td>2104</td>
<td>Start-of-Stream Reached</td>
<td>Returned to the initial content</td>
</tr>
<tr>
<td>2306</td>
<td>Continuous Feed Terminated</td>
<td>Live finished</td>
</tr>
<tr>
<td>2401</td>
<td>Ticket Expired</td>
<td>Viewing right expired</td>
</tr>
<tr>
<td>4400</td>
<td>Error Reading Content Data</td>
<td>Data read error</td>
</tr>
<tr>
<td>5200</td>
<td>Server Resource Unavailable</td>
<td>Resource cannot be secured</td>
</tr>
<tr>
<td>5401</td>
<td>Downstream Failure</td>
<td>Stream could not be obtained</td>
</tr>
<tr>
<td>5402</td>
<td>Client Session Terminated</td>
<td>Session terminated</td>
</tr>
<tr>
<td>5403</td>
<td>Server Shutting Down</td>
<td>Server is performing end processing</td>
</tr>
<tr>
<td>5404</td>
<td>Internal Server Error</td>
<td>Server error</td>
</tr>
</tbody>
</table>

A.6.7.2 Response message format

It is desirable that the client who received this method should return a response to the server in the following format. When a load of a processor is remarkably high, such as during stream playback, it may be ignored. Even if the server receives any response from the client, the server operation must not be changed.
A.7 SDP Description

This section explains the method of describing the information that needs to be sent and received between the server and the client in an SDP, concerning each of presentation, playback, and decoding.

A.7.1 Presentation Information

The information about an author and a content must be described in accordance with the notation described in the RFC 2327. The format is as follows. For details, refer to the RFC 2327.

```
o=username session_id version network_type address_type address
s=session_name
```

However, when these information exist in the activation meta-file of a video playback player in the client, priority is given to the information.

Example of description)

```
o=HSAC 1234 5678 IN IP4 192.168.0.2
s=new_presentation
```

A.7.2 Control Information

The content providable time, content providable range, and the time-shift enable time are described.

A.7.2.1 Content providable time

The information about the content providable time must be described in accordance with the notation described in the RFC 2327. The format is as follows. For details, refer to the RFC 2327.

```
t=Start_Time End_Time
```

Example of description)

```
t=2873397496 2873404696
```

A.7.2.2 Content providable range

The time range of content is provisioned for usage of a reference in jump playback. The format is as follows.

```
a=range:Start_Time-End_Time
```

Here, specify `Start_Time` and `End_Time` in accordance with the RFC ([3.6] Normal Play Time) as shown below. However, do not describe “now”.

```
npt-time = npt-sec | npt-hhmss
npt-sec = 1*DIGIT [ "." *DIGIT ]
npt-hhmss = npt-hh ":" npt-mm ":" npt-ss [ ":" *DIGIT ]
npt-hh = 1*DIGIT ; any positive number
npt-mm = 1*2DIGIT ; 0-59
npt-ss = 1*2DIGIT ; 0-59
```

For Live, the start time and the end time must be set to 0. That is, they must be described like “a=range:0.0-0.0”. In this appendix, it must be described before media description.

Example of description)

```
a=range:0.0-120.45
```
A.7.2.3 Enabled time of Time-shift

The maximum value of the time range for time-shift must be described as below. For the Live that can be time-shifted, time-shift the value of earliest time position where the client can jump must be shown in units of seconds. When the value is 0, it indicates that time-shift is disabled. Further, when jumping to any time is enabled, it may be indicated using “-1”. In this appendix, it must be described before media description.

<table>
<thead>
<tr>
<th>a=recordtime:Record_Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record_Period = &lt; any positive number &gt; or 0 or -1</td>
</tr>
</tbody>
</table>

Example of description)

Example 1) a= recordtime:3600
Example 2) a= recordtime:0
Example 3) a= recordtime:-1

For example, example 1 shows jumping into the past is enabled for one hour, example 2 shows the time-shift operation is disabled, and example 3 shows that jumping to any time is enabled, respectively.

A.7.3 Decoding Information

A decoding type and a transfer rate are described as the information necessary for decoding.

A.7.3.1 Coding (codec) type

For a system that sends a stream without using an RTP, decoding in the client is realized by transferring a transmission format without transferring a coding type. In such system, the transmission format must be described as follows.

m=application 0 udp Transport_Format

For an MPEG-2 TS stream in particular, the transmission format must be described as follows.

m=application 0 udp mpeg2-ts

When an RTP is used, the transmission format must be described in accordance with the RFC 2327 as follows. For details, refer to the RFC 2327.

m=media port transport fmt_list

Example of description)

m=video 0 ud 31
m=audio 0 udp 0

A.7.3.2 Transfer rate

In this specification, only a stream of a fixed bit rate is handled. The bit rate must be described as the number of bits per second, as shown below. However, when a value that time, the differs in the bitrate in the SETUP response, the client must decode the stream by using a value of the SETUP response.

<table>
<thead>
<tr>
<th>a=bitrate:Bitrate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bitrate = &lt; any positive number &gt;</td>
</tr>
</tbody>
</table>

In this appendix, it must be described before media description.

Example of description)

a=bitrate:6000000

Copyright (C) 2001-2002 HIKARI Service Architecture Consortium. All rights reserved.
A.7.4 Sample SDP Messages

A.7.4.1 VoD sample

o=HSAC
s=sample_presentation1
t=0 0
a=range:0.0-3600.0
a=bitrate:6000000
m=application 0 udp mpeg2-ts

A.7.4.2 Sample of Live with time-shift

o=HSAC
s=sample_presentation2
t=2873397496 2873404696
a=range:0.0-0.0
a=recordtime:3600
a=bitrate:6000000
m=application 0 udp mpeg2-ts
A.8 Sample Sequences

A.8.1 Playback Start

Client → Server
OPTION * RTSP/1.0
CSeq: 1
User-Agent: HIKARI STB 1.0 (Linux 2.2.18)

Server → Client
RTSP/1.0 200 OK
CSeq: 1
Public: OPTIONS, DESCRIBE, SETUP, PLAY, PAUSE, TEARDOWN, ANNOUNCE, GET_PARAMETER
VersionSupport: HSAC/1.0

Client → Server
DESCRIBE rtsp://192.168.1.2:554/sample.mp2 RTSP/1.0
CSeq: 2
Accept: application/sdp
User-Agent: HIKARI STB 1.0 (Linux 2.2.18)

Server → Client
RTSP/1.0 200 OK
Server: HIKARI Video Server 1.0 (Linux Server)
CSeq: 2
VersionSupport: HSAC/1.0
Last-Modified: Fri, 9 Nov 2001 04:14:49 GMT
Content-length: 475
Date: Fri, 9 Nov 2001 06:28:22 GMT
Expires: Fri, 9 Nov 2001 06:28:22 GMT
Content-Type: application/sdp

v=0
o=HIKARI IN IP4 192.168.1.2
s=video server
i=How to use video server
u=http://192.168.1.2/pub/video.html
e=VideoServer@hikari-sac.org
c=IN IP4 192.168.1.2
a=range:npt=0-37.40333
m=audio 0 RTP/AVP 96
m=video 0 RTP/AVP 97
a=orient:multimedia

Client → Server
SETUP rtsp://192.168.1.2:554/sample.mp2 RTSP/1.0
CSeq: 3
Transport: RAW/MP2T/UDP;unicast;destination=192.168.0.2;client_port=6970
User-Agent: HIKARI STB 1.0 (Linux 2.2.18)
Server → Client
RTSP/1.0 200 OK
Server: HIKARI Video Server 1.0 (Linux Server)
CSeq: 3
VersionSupport: HSAC/1.0
Last-Modified: Fri, 9 Nov 2001 06:28:22 GMT
Session: 38581691221856
Expires: Fri, 9 Nov 2001 06:28:22 GMT
Date: Tue, Fri, 9 Nov 2001 06:28:22 GMT
Transport: RAW/MP2T/UDP;unicast;destination=192.168.0.2;client_port=6970;server_port=1234;bitrate=4000000

Client → Server
PLAY rtsps://192.168.1.2:554/sample.mp2 RTSP/1.0
CSeq: 4
Range: 0.000000-37.403333
Session: 38581691221856
User-Agent: HIKARI STB 1.0 (Linux 2.2.18)

Server → Client
RTSP/1.0 200 OK
Server: HIKARI Video Server 1.0 (Linux Server)
CSeq: 4
Range: 0.000000-37.403333
Session: 38581691221856

A.8.2 Jump Playback

Client → Server
PAUSE rtsps://192.168.1.2:554/sample.mp2 RTSP/1.0
CSeq: 100
Session: 38581691221856
User-Agent: HIKARI STB 1.0 (Linux 2.2.18)

Server → Client
RTSP/1.0 200 OK
Server: HIKARI Video Server 1.0 (Linux Server)
CSeq: 100
Range: 11.090000
Session: 38581691221856

Client → Server
PLAY rtsps://192.168.1.2:554/sample.mp2 RTSP/1.0
CSeq: 101
Range: current-
Session: 38581691221856
User-Agent: HIKARI STB 1.0 (Linux 2.2.18)

Server → Client
RTSP/1.0 200 OK
Server: HIKARI Video Server 1.0 (Linux Server)
CSeq: 101
Range: 11.090000-37.403333
Session: 38581691221856
A.8.3 Fast forward 1 (When fast forward is available)

Client → Server
PAUSE rtsp://192.168.1.2:554/sample.mp2 RTSP/1.0
CSeq: 100
Session: 38581691221856
User-Agent: HIKARI STB 1.0 (Linux 2.2.18)

Server → Client
RTSP/1.0 200 OK
Server: HIKARI Video Server 1.0 (Linux Server)
CSeq: 100
Range: 11.090000
Session: 38581691221856

Client → Server
PLAY rtsp://192.168.1.2:554/sample.mp2 RTSP/1.0
CSeq: 101
Range: current-end
Scale: 2.1
Session: 38581691221856
User-Agent: HIKARI STB 1.0 (Linux 2.2.18)

Server → Client
RTSP/1.0 200 OK
Server: HIKARI Video Server 1.0 (Linux Server)
CSeq: 101
Range: 11.090000-37.403333
Scale: 2.0
Session: 38581691221856
A.8.4 Fast forward 2 (When fast forward is unavailable)

Client → Server

PAUSE rtsp://192.168.1.2:554/sample.mp2 RTSP/1.0
CSeq: 100
Session: 38581691221856
User-Agent: HIKARI STB 1.0 (Linux 2.2.18)

Server → Client

RTSP/1.0 200 OK
Server: HIKARI Video Server 1.0 (Linux Server)
CSeq: 100
Range: 11.090000
Session: 38581691221856
User-Agent: HIKARI STB 1.0 (Linux 2.2.18)

Client → Server

PLAY rtsp://192.168.1.2:554/sample.mp2 RTSP/1.0
CSeq: 101
Range: current-end
Scale: 2.1
Session: 38581691221856
User-Agent: HIKARI STB 1.0 (Linux 2.2.18)

Server → Client

RTSP/1.0 400 Bad Request
Server: HIKARI Video Server 1.0 (Linux Server)
CSeq: 101
Session: 38581691221856

A.8.5 End 1 (Content end)

Server → Client

ANNOUNCE rtsp://192.168.1.2:554/sample.mp2 RTSP/1.0
Server: HIKARI Video Server 1.0 (Linux Server)
CSeq: 1
Session: 38581691221856
Notice: 2101 End-of Stream Reached

Client → Server

RTSP/1.0 200 OK
CSeq: 1
Session: 38581691221856
User-Agent: HIKARI STB 1.0 (Linux 2.2.18)

Client → Server

TEARDOWN rtsp://192.168.1.2:554/sample.mp2 RTSP/1.0
CSeq: 100
Session: 38581691221856
User-Agent: HIKARI STB 1.0 (Linux 2.2.18)

Server → Client

RTSP/1.0 200 OK
Server: HIKARI Video Server 1.0 (Linux Server)
CSeq: 100
Session: 38581691221856

Copyright (C) 2001-2002 HIKARI Service Architecture Consortium. All rights reserved.
A.8.6 End 2 (End by user)

Client → Server
PAUSE rtsp://192.168.1.2:554/sample.mp2 RTSP/1.0  
CSeq: 100  
Session: 38581691221856  
User-Agent: HIKARI STB 1.0 (Linux 2.2.18)

Server → Client
RTSP/1.0 200 OK  
Server: HIKARI Video Server 1.0 (Linux Server)  
CSeq: 100  
Range: 11.090000  
Session: 38581691221856

Client → Server
TEARDOWN rtsp://192.168.1.2:554/sample.mp2 RTSP/1.0  
CSeq: 101  
Session: 38581691221856  
User-Agent: HIKARI STB 1.0 (Linux 2.2.18)

Server → Client
RTSP/1.0 200 OK  
Server: HIKARI Video Server 1.0 (Linux Server)  
CSeq: 101  
Session: 38581691221856