TTC標準
Standard

# JJ-40.20

## Content Transfer System
## Using SOAP/HTTP in NGN Environments

Version 1.1

Established on April 24, 2012

THE TELECOMMUNICATION TECHNOLOGY COMMITTEE

Telecommunication
Technology
Committee

Contents

<Reference information>

## 1. Relationship with International Recommendations

This standard has no particular relation to international recommendations.

## 2. History of Revision

| Revision | Date | Description |
|---|---|---|
| Version 1.0 | February 23, 2012 | Initial publication |
| Version 1.1 | April 24, 2012 | Error correction |
| | | |
| | | |

## 3. Industrial Property Rights

Information regarding submission of "The Policy for the Handling of Industrial Property Rights" associated with this standard is available on TTC’s Web page.

## 4. Contact

Media Coding Working Group

## 1 Introduction

This standard was developed to accomplish content sending/reception between telecommunication systems using SOAP/HTTP after SIP-based call establishment in NGN environments under JT-Q3402.

## 2 References

The literature that is referenced in this standard is as follows.

[MIME] "IANA MIME Media Types http://www.iana.org/assignments/media-type/", IANA

[Q3402] "NGN UNI Signalling Profile (Protocol Set 1)", TTC Standard JT-Q3402, Ver. 1, The Telecommunication Technologies Committee, May 2009

[RFC1867] "Form-based File Upload in HTML", RFC 1867, IETF, November 1995

[RFC2046] "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, IETF, November 1996

[RFC2388] "Returning Values from Forms: multipart/form-data", RFC 2388, IETF, August 1998

[RFC2616] "Hypertext Transfer Protocol", RFC 2616, IETF, June 1999

[RFC2854] "The "text/html" Media Type", RFC 2854, IETF, June 2000

[RFC3902] "The "application/soap+xml" media type", RFC 3902, IETF, September 2004

[RFC4145] "TCP-Based Media Transport in the Session Description Protocol (SDP)", TTC Standard JF-IETF-RFC4145, The Telecommunication Technologies Committee, March 2007

[RFC5234] "Augmented BNF for Syntax Specifications: ABNF", RFC 5234, IETF, January 2008

[SOAP] "SOAP Version1.2 Part 1: Messaging Framework (Second Edition)", W3C, April 2007
"SOAP Version1.2 Part 2: Adjuncts (Second Edition)", W3C, April 2007

[XML] "Extensible Markup Language(XML) 1.0 (Fifth Edition)", W3C, November 2008

[XML Namespaces] "Namespaces in XML 1.0 (Third Edition)", W3C, December 2009

[XML Schema] "XML Schema Part 0: Primer Second Edition", W3C, October 2004
"XML Schema Part 1: Structures Second Edition", W3C, October 2004
"XML Schema Part 2: Datatypes Second Edition", W3C, October 2004

## 3 Terminology

Content : Digital data of pictures, images, documents, etc.

Content transfer : Sending and receiving of content

Out-band : Signal channel for call settling using telephone numbers

In-band : Channel for exchanging media information between pieces of equipment after out-band call establishment

Outgoing call : Starting request for call establishment using telephone numbers

Incoming call : Receiving request for call establishment

Outgoing call equipment
: Equipment for requesting call establishment using communication protocols as specified in this standard

Incoming call equipment
: Equipment for receiving call establishment request using communication protocols as specified in this standard

Equipment : Outgoing call equipment and incoming call equipment

Sending function : Function for sending content using communication protocols as specified in this standard

Receiving function : Function for receiving content using communication protocols as specified in this standard

Client function : Function for sending request for content transfer

Server function : Function for receiving request for content transfer

NGN : Next Generation Network that uses Internet protocol techniques and realizes multimedia services with a combination of telephones, data communications and streaming broadcasts

UNI : Specification of interface that connects a carrier's telecommunication facility to an end user's facility (User-Network Interface)

BN form : Backus-Naur Form, or metalanguage that is used to define a context-free grammar. It is used to define protocol syntax specification language such as ASN.1 and XML syntax.

HTTP : Protocol that is used to send and receive data between a server and client

XML : Markup language used to describe the meaning and structure of a document and data

SOAP : Protocol based on XML and HTTP and used to read data and services in other computers

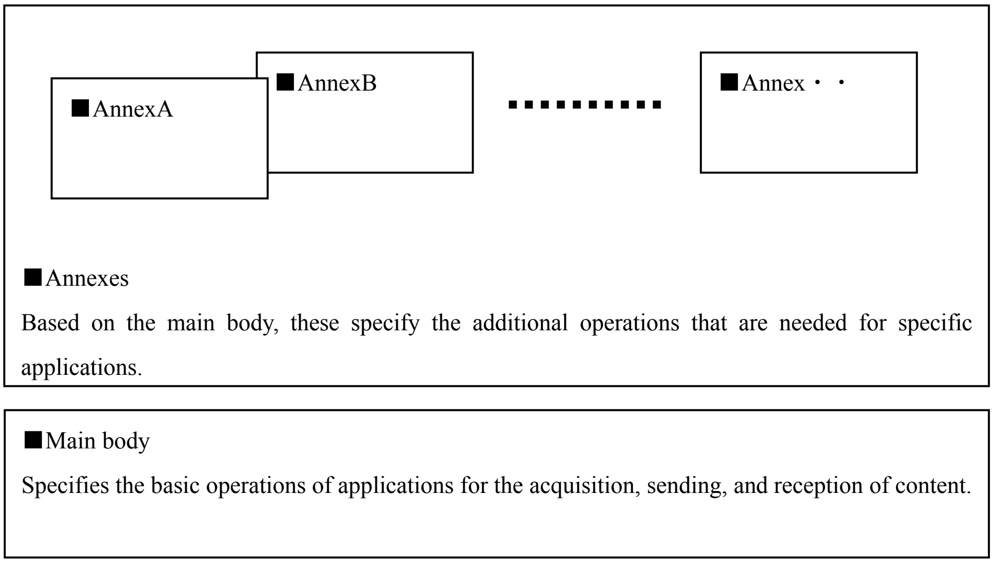
SOAP HTTP Binding : Use of an HTTP request and its HTTP response for the transfer of a SOAP message

Quasi-normal state : A state that can be generated by some combinations of operations in spite of their being performed as specified

Abnormal state : A state that is generated after an abnormal action that is outside the scope of this standard

## 4 Structure of This Standard

In its main body, this standard specifies the basic communications protocols for the acquisition, sending, and reception of content files. Annexes specify protocols for the transmission of additional information, based on the main body. The additional information is used to deal with the content files in sending equipment and receiving equipment by extension of the protocols specified by the main body.



■Annexes

Based on the main body, these specify the additional operations that are needed for specific applications.

■Main body

Specifies the basic operations of applications for the acquisition, sending, and reception of content.

Figure 4-1/JJ-40.20: Structure of this standard

### 4.1 Revision of This Standard

If the content of the main body and its annexes are revised, this standard shall be revised. If the revision changes the protocols, the version of the protocol in Section 9.1 shall be revised.

### 4.2 Notes

#### 4.2.1 Description of Syntax

XML documents that are contained in this standard shall be described in accordance with the BN form.

For the details of the descriptions of different syntaxes, refer to the references in Chapter 2.

#### 4.2.2 Namespace of XML Element

The namespace of an XML element that is defined by this protocol shall be as follows:

| | | |
|---|---|---|
| Main body | : | http://www.ttc.or.jp/mmsys/ct |

## 5 System Model

To explain sending processing in the content transfer system, a model of typical sending equipment is shown in Figure 5-1 with a focus on the data flow.

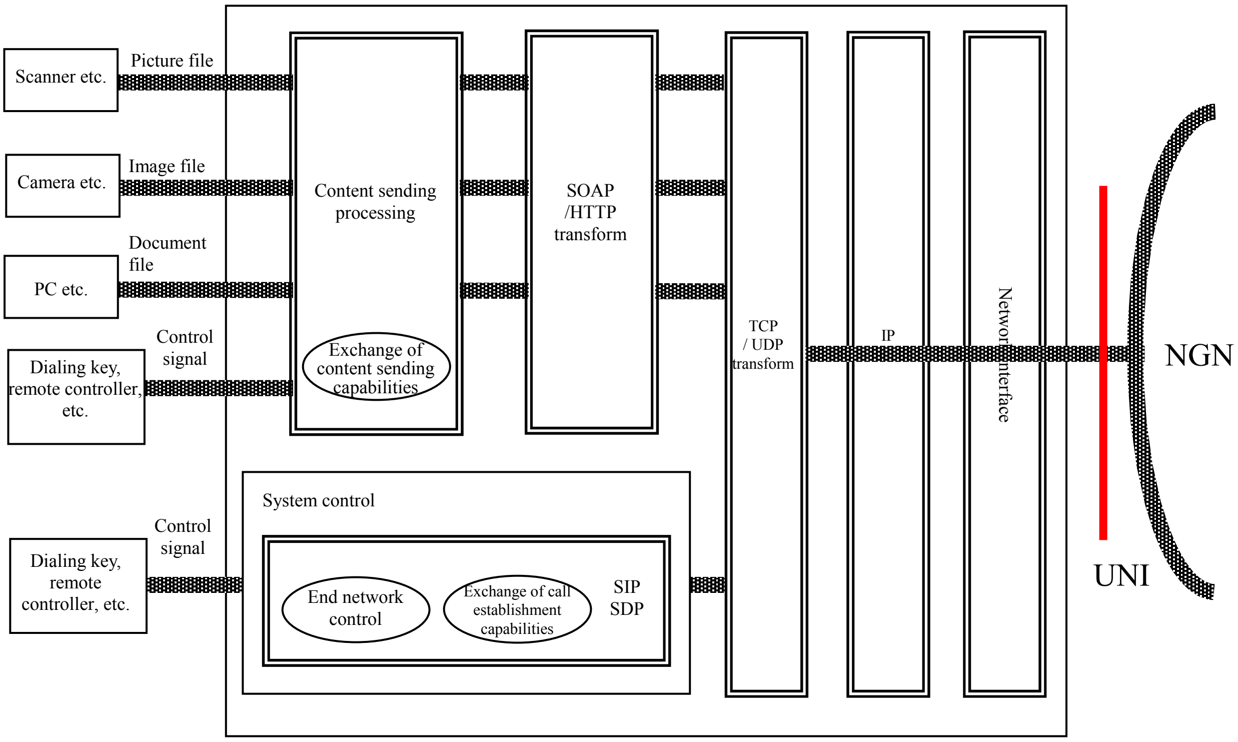
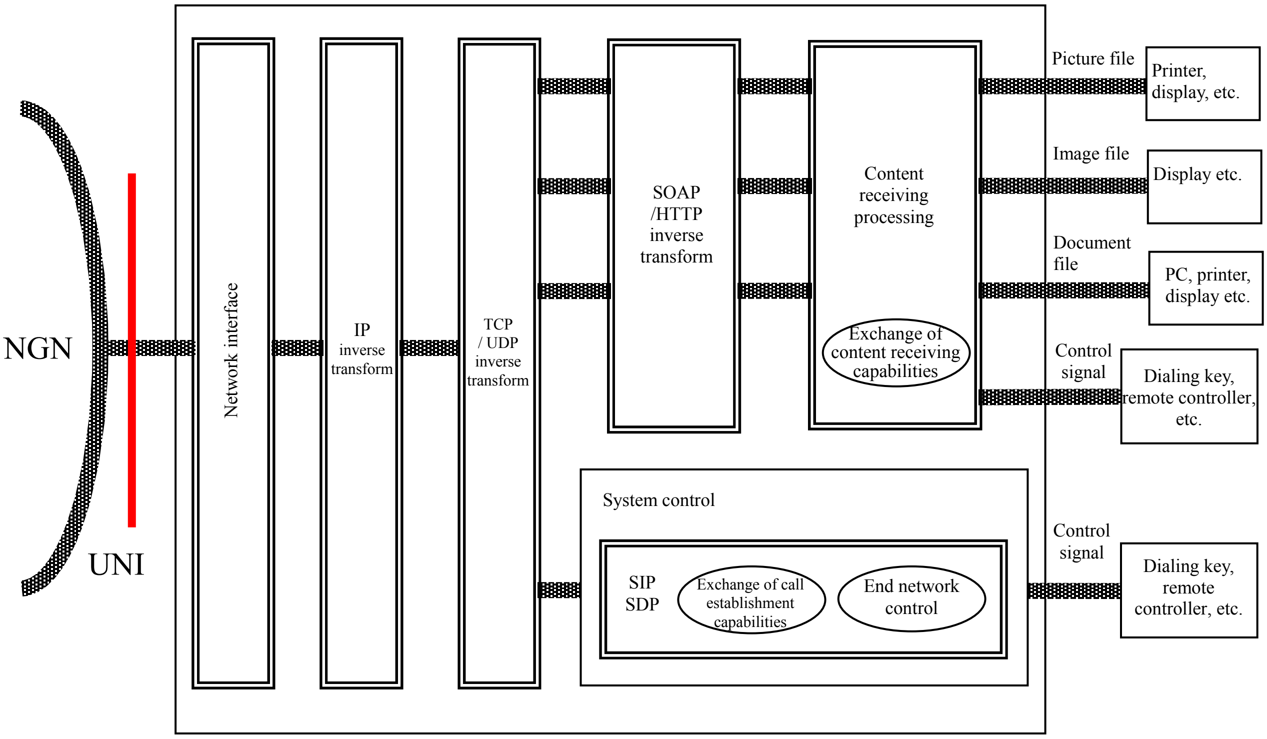


Figure 5-1/JJ-40.20: Sending equipment model with a focus on data flow

To explain receiving processing in the content transfer system, a model of typical receiving equipment is shown in Figure 5-2 with a focus on the data flow.



NGN
UNI
Network interface
IP inverse transform
TCP / UDP inverse transform
SOAP /HTTP inverse transform
Content receiving processing
Exchange of content receiving capabilities
System control
SIP SDP
Exchange of call establishment capabilities
End network control
Picture file
Printer, display, etc.
Image file
Display etc.
Document file
PC, printer, display etc.
Control signal
Dialing key, remote controller, etc.
Control signal
Dialing key, remote controller, etc.

Figure 5-2/JJ-40.20: Receiving equipment model with a focus on data flow

# 6 Protocol Configuration

A protocol configuration shall have a hierarchical structure that conforms to an OSI reference model.

After SIP-based call establishment under JT-Q3402, content transfer shall be in accordance with the in-band protocols using SOAP/HTTP that are specified by this standard. Due to the fact that these protocols are specified as in-band protocols, content transfer can also be realized by these protocols after connecting to remote equipment by some other means than NGN.

Table 6-1/JJ-40.20: Protocol configuration

| Layer | | Protocols used | |
|---|---|---|---|
| | | Out-band | In-band |
| 7 | Application | SIP, SDP | SOAP<br>HTTP |
| 6 | Presentation | | |
| 5 | Session | | |
| 4 | Transport | UDP | TCP |
| 3 | Network | IPv4/IPv6 | |
| 2 | Data link | Ethernet etc. | |
| 1 | Physical | | |

<Reason for use of SOAP/HTTP>

For the following reasons, the SOAP/HTTP protocols shall be used for content transfer by the in-band method.

(1) HTTP is most widely used in communication protocols over a TCP connection.

(2) SOAP is used as the standard for remote procedure calls (RPC) as WEB services become widely available.

(3) Many development environments have been established. This enables the short-term development of the relevant protocols.

(4) Any function can be freely defined in SOAP. This facilitates the extension of the protocols.

## 6.1 SIP and SDP

The content transfer that is specified by these protocols shall be conducted in the SIP session.

It shall conform to TTC Standard JT-Q3402.

## 6.2 TCP

To transfer HTTP packets, TCP shall be used as the transport layer protocol as laid down in JF-IETF-RFC4145.

## 6.3 HTTP

This protocol shall be used for content transfer. It shall conform to RFC2616 (HTTP1.1).

A SOAP message shall be placed in the HTTP message body and then transferred to a destination. (SOAP HTTP Binding)

## 6.4 SOAP

This protocol shall conform to SOAP1.2 [SOAP].

## 6.5 SOAP HTTP Binding

In accordance with SOAP Version1.2 Part 2 [SOAP], an HTTP request and its HTTP response shall be used to transfer a SOAP message. It shall conform to the SOAP Request-Response Message Exchange patterns.

# 7 Protocol Scope

## 7.1 Scope

After SIP-based call establishment by the out-band method under JT-Q3402, this standard specifies three protocols for the in-band method using SOAP/HTTP, in order to realize content transfer between pieces of equipment.

### 7.1.1 In-band Protocol Specification 1

This is a protocol specification for a connection mode in which outgoing call equipment calls a content server containing multiple items of content and the equipment acquires that content based on a list of content received from the server.

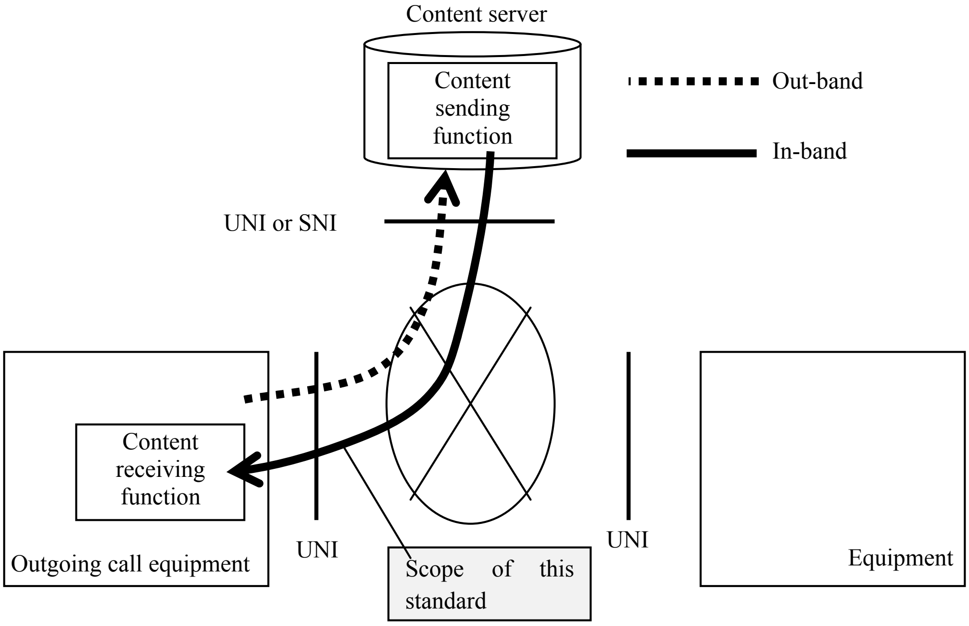


Figure 7-1/JJ-40.20: Scope of in-band protocol specification 1

### 7.1.2 In-band Protocol Specification 2

This is a protocol specification for a connection mode in which outgoing call equipment calls incoming call equipment and sends content to the incoming call equipment.

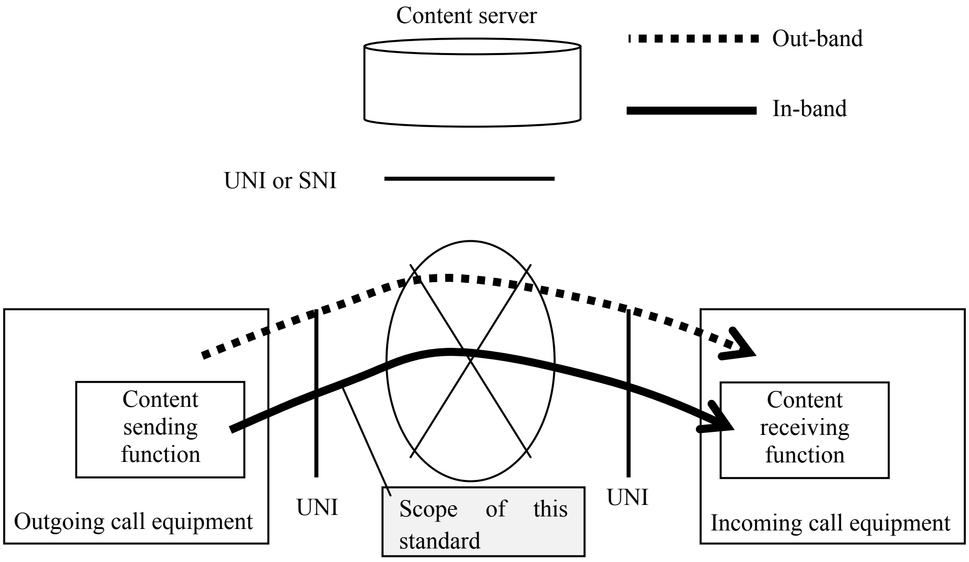


Figure 7-2/JJ-40.20: Scope of in-band protocol specification 2

### 7.1.3 In-band Protocol Specification 3

This is a protocol specification for a connection mode in which outgoing call equipment calls incoming call equipment and the incoming call equipment sends content to the outgoing call equipment.

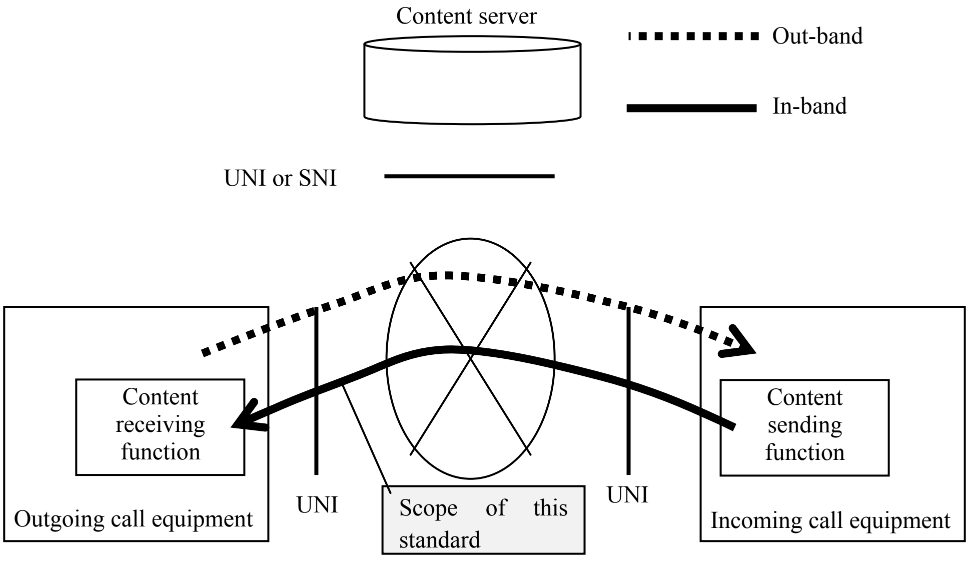


Figure 7-3/JJ-40.20: Scope of in-band protocol specification 3

## 7.2 Out-band Connection Modes

For SIP/SDP call establishment by the out-band method, the two modes shown in Table 7-1 are defined.

Table 7-1/JJ-40.20: Description of out-band connection modes

| Connection modes | | Description |
|---|---|---|
| 1 | Single call | Call that establishes SIP session to perform content transfer only |
| 2 | Additional call | Call that adds SIP session to existing voice, image, and data communication calls to perform content transfer |

## 7.3 In-band Connection Modes

For content transfer by the in-band method, the three connection modes shown in Table 7-2 are defined. The in-band protocol specifications listed in Table 7-3 apply to these modes.

Table 7-2/JJ-40.20: Description of in-band connection modes

| Connection modes | | Description |
|---|---|---|
| 1 | Content acquisition on outgoing call side | Outgoing call equipment acquires content that is owned by incoming call equipment. |
| 2 | Content sending from outgoing call side | Outgoing call equipment sends its content to incoming call equipment. |
| 3 | Content sending from incoming call side | Incoming call equipment sends its content to outgoing call equipment. |

Table 7-3/JJ-40.20: In-band connection modes and applied protocol specifications

| Connection modes | | Applied protocol specifications |
|---|---|---|
| 1 | Content acquisition on outgoing call side | Section 7.1.1 "In-band Protocol Specification 1" |
| 2 | Content sending from outgoing call side | Section 7.1.2 "In-band Protocol Specification 2" |
| 3 | Content sending from incoming call side | Section 7.1.3 "In-band Protocol Specification 3" |

## 7.4 Communication Phases for Content Transfer

Content transfer uses the out-band and in-band methods. The out-band method is used to perform a call establishment phase in which a call is established by SIP/SDP and a call release phase in which a call is released. The in-band method is used to perform a phase whereby TCP connection is established after call establishment, a capability notification and confirmation phase in which the content sending side inquires of the content receiving side about its capability, a content list acquisition phase in which a list of obtainable content is understood, a job creation phase in which a job for content file transfer is created, a content sending phase in which a content file is transferred, and a phase in which the TCP connection is disconnected.

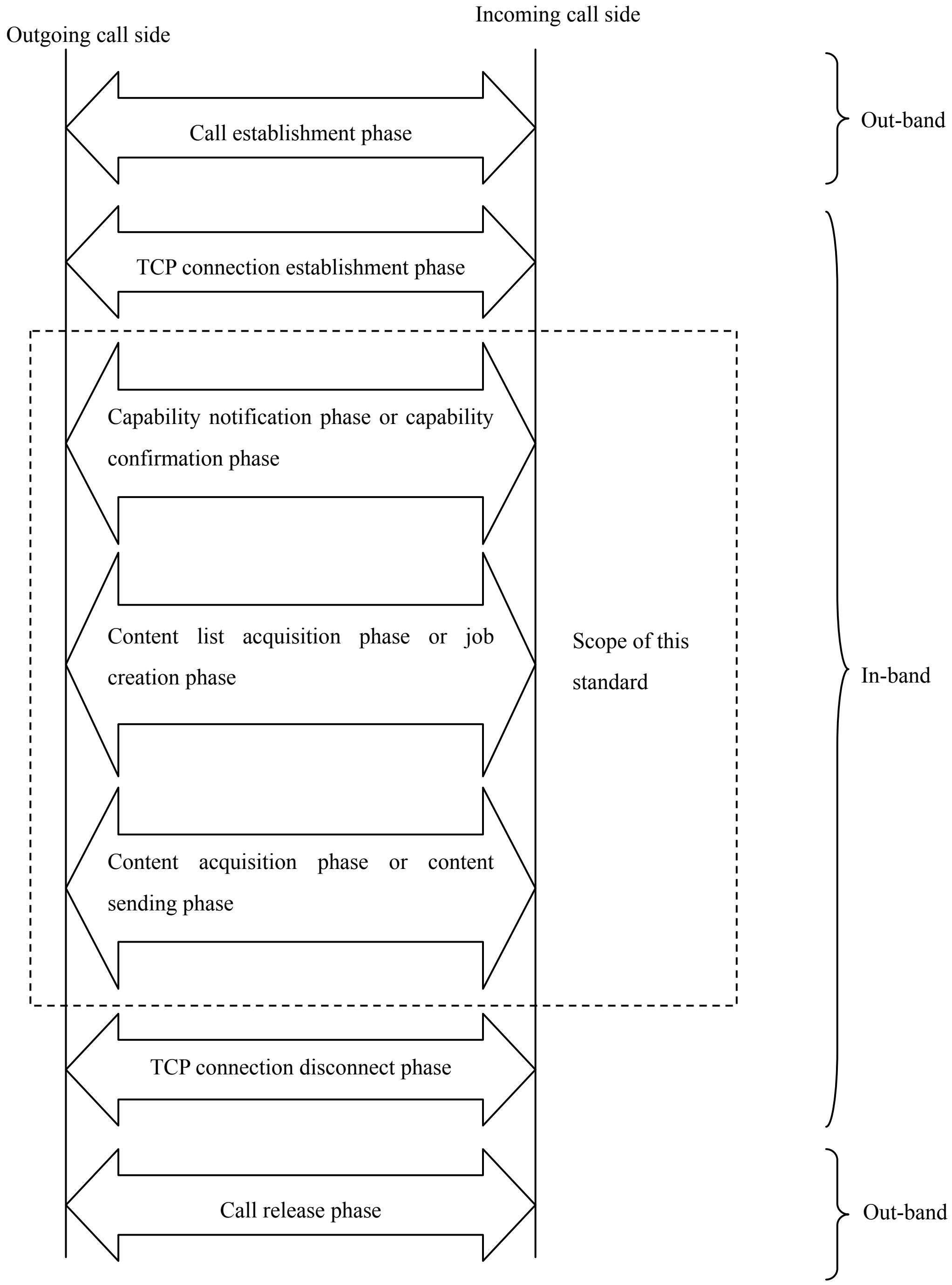

Incoming call side
Outgoing call side
Call establishment phase
Out-band
TCP connection establishment phase
Capability notification phase or capability confirmation phase
Content list acquisition phase or job creation phase
Scope of this standard
In-band
Content acquisition phase or content sending phase
TCP connection disconnect phase
Call release phase
Out-band

Figure 7-4/JJ-40.20: Communication phases for content transfer

### 7.4.1 Call Establishment Phase

This phase is intended to establish a call before content transfer by the in-band method.

### 7.4.2 TCP Connection Establishment Phase

This phase is intended to establish a TCP connection before content transfer. In this protocol, the content sending side shall start a TCP connection with the content receiving side. The sending side shall open the TCP connection to a listening port posted through SIP/SDP by the receiving side. The sending side (not the receiving side) shall disconnect the TCP connection established in this phase. The method of establishing the connection shall conform to TTC Standard JT-Q3402.

### 7.4.3 Capability Notification Phase

This phase is for the receiving side to notify the sending side of its capability before the content list acquisition phase in connection mode 1 of Section 7.3. Specifically, the receiving side informs the other side of the content format that it can receive (MIME-TYPE).

### 7.4.4 Capability Confirmation Phase

This phase is for the sending side to identify the capability of the receiving side before the job creation phase in connection modes 2 and 3 of Section 7.3. Specifically, the sending side can confirm the format (MIME-TYPE) of the content that can be received by the receiving side.

### 7.4.5 Content List Acquisition Phase

This phase is for the receiving side to acquire a list of content that is owned by the sending side before the content acquisition phase in connection mode 1 of Section 7.3.

### 7.4.6 Job Creation Phase

This phase is for the sending side to issue a job creation request to the receiving side and for the receiving side to answer whether to receive the request before the content sending phase in connection modes 2 and 3 of Section 7.3. The job creation request includes information of the content (file name, capacity, MIME-TYPE, etc.) that is to be sent.

### 7.4.7 Content Acquisition Phase

This phase is for the receiving side to select content from a list of content that is acquired from the sending side in the content list acquisition phase, and issue a content acquisition request using the GET method of the HTTP protocol to acquire the content, in connection mode 1 of Section 7.3.

### 7.4.8 Content Sending Phase

This phase is for the sending side to send the receiving side an HTTP message body in which a file to be transferred is contained, using the POST method of the HTTP protocol, in connection modes 2 and 3 of Section 7.3. One file shall be contained in one HTTP message body.

### 7.4.9 TCP Connection Disconnect Phase

This phase is intended to disconnect a TCP connection after the completion of the content acquisition phase and the content sending phase. In this protocol, the TCP connection shall be disconnected from the equipment that opened the TCP connection.

### 7.4.10 Call Release Phase

This phase is intended to release a call after the completion of a content transfer by the in-band method.

# 8 Connection Modes

This chapter details the connection modes in "7.2 Out-band Connection Modes" and "7.3 In-band Connection Modes".

## 8.1 Out-band Connection Modes

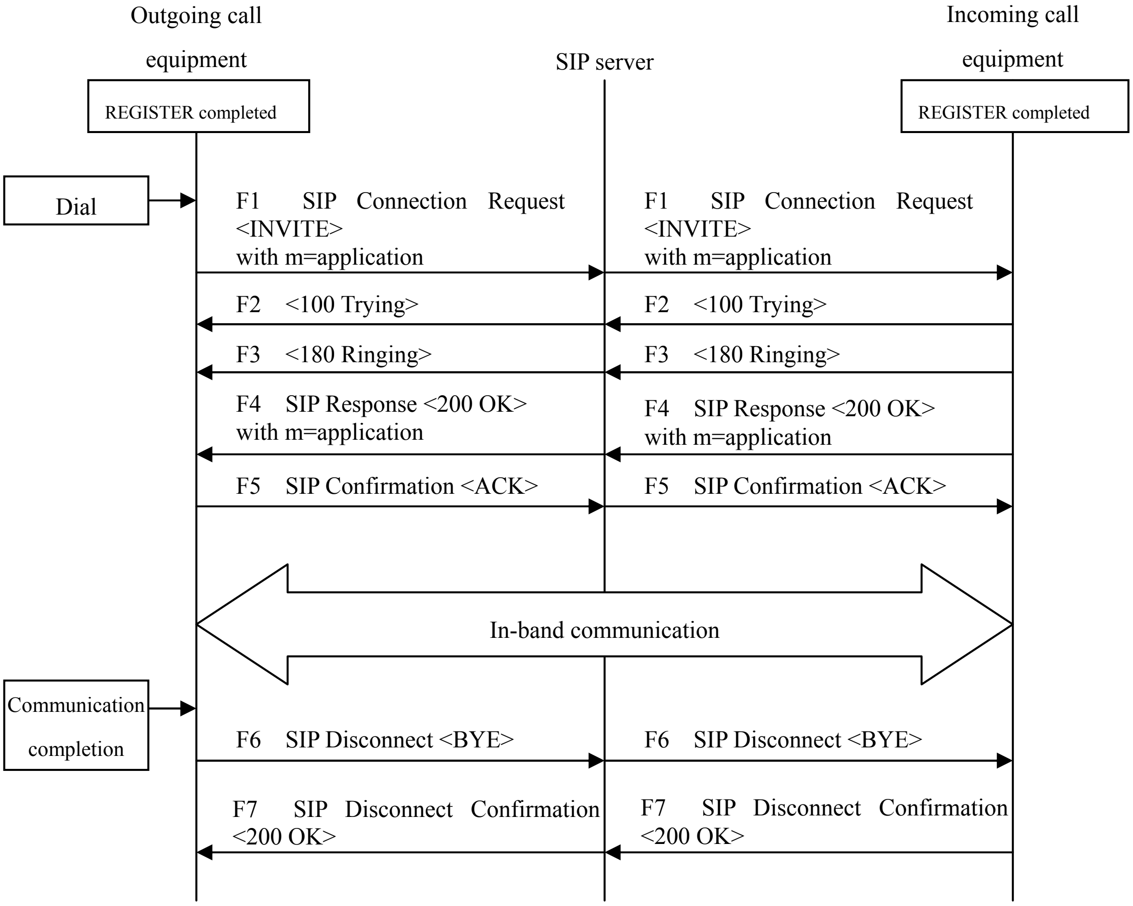
This section describes the connections modes listed in Table 7-1.

SIP call establishment and call release shall conform to JT-Q3402 and parameters shall conform to JF-IETF-RFC4145.

Examples are shown in Sections 8.1.1 and 8.1.2.

### 8.1.1 Connection Mode Case 1 (Single Call)

This is a call connection mode case in which a TCP session is established by SIP to perform content transfer only.

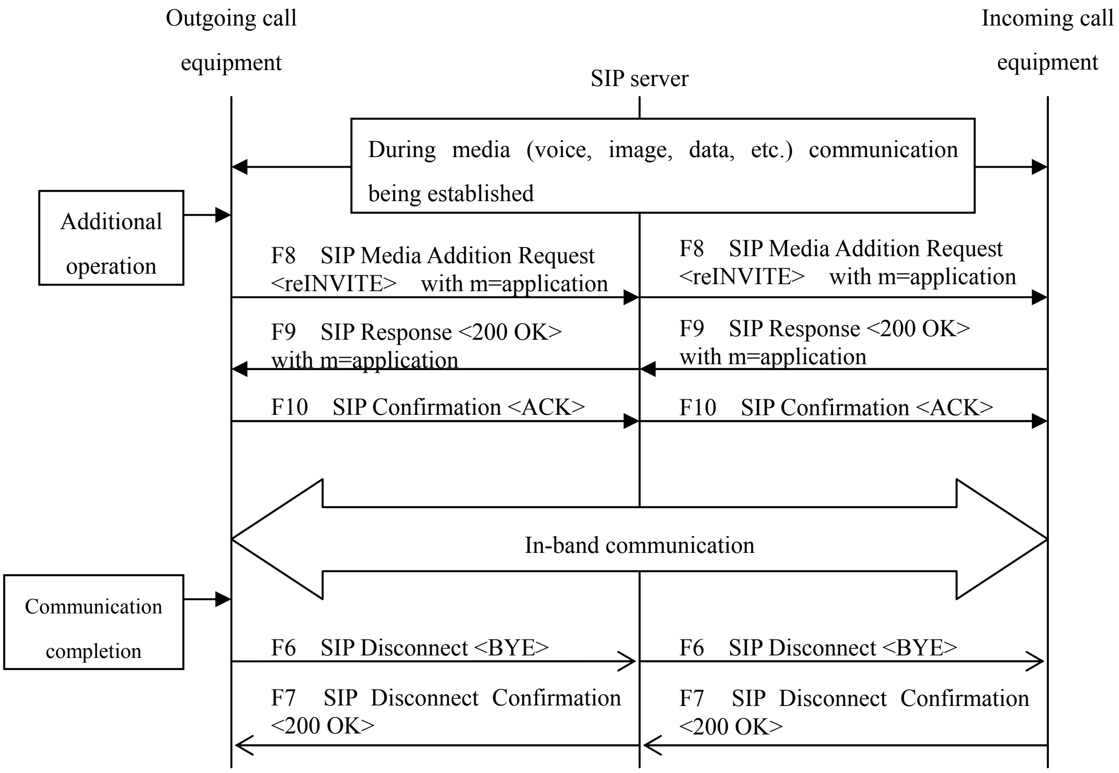

Outgoing call equipment
SIP server
Incoming call equipment
REGISTER completed
REGISTER completed
Dial
F1 SIP Connection Request <INVITE> with m=application
F1 SIP Connection Request <INVITE> with m=application
F2 <100 Trying>
F2 <100 Trying>
F3 <180 Ringing>
F3 <180 Ringing>
F4 SIP Response <200 OK> with m=application
F4 SIP Response <200 OK> with m=application
F5 SIP Confirmation <ACK>
F5 SIP Confirmation <ACK>
In-band communication
Communication completion
F6 SIP Disconnect <BYE>
F6 SIP Disconnect <BYE>
F7 SIP Disconnect Confirmation <200 OK>
F7 SIP Disconnect Confirmation <200 OK>

* Described for the case of disconnecting from the outgoing call equipment as an example

Figure 8-1/JJ-40.20: Case of single call-based communication establishment and disconnection between outgoing call equipment and incoming call equipment

### 8.1.2 Connection Mode Case 2 (Additional Call)

This is a call connection mode case in which a TCP session is added by SIP to perform content transfer during media (voice, image, data, etc.) communications.


Outgoing call equipment
SIP server
Incoming call equipment
During media (voice, image, data, etc.) communication being established
Additional operation
F8 SIP Media Addition Request <reINVITE> with m=application
F8 SIP Media Addition Request <reINVITE> with m=application
F9 SIP Response <200 OK> with m=application
F9 SIP Response <200 OK> with m=application
F10 SIP Confirmation <ACK>
F10 SIP Confirmation <ACK>
In-band communication
Communication completion
F6 SIP Disconnect <BYE>
F6 SIP Disconnect <BYE>
F7 SIP Disconnect Confirmation <200 OK>
F7 SIP Disconnect Confirmation <200 OK>

*Described for the case of simultaneously disconnecting voice, image, and data from the outgoing call equipment as an example

Figure 8-2/JJ-40.20: Case of additional call-based communication establishment and disconnection between outgoing call equipment and incoming call equipment

## 8.2 In-band Connection Modes

This section details the connection modes listed in Table 7-2.

### 8.2.1 Connection Mode 1 (Content Acquisition from Outgoing Call Equipment)

In this connection mode, multiple items of content exist on the server equipment (incoming call side), while the client equipment (outgoing call side) selects and acquires multiple items of content on the server side. (For details on the designation of client and server operations, see Section 9.3.)

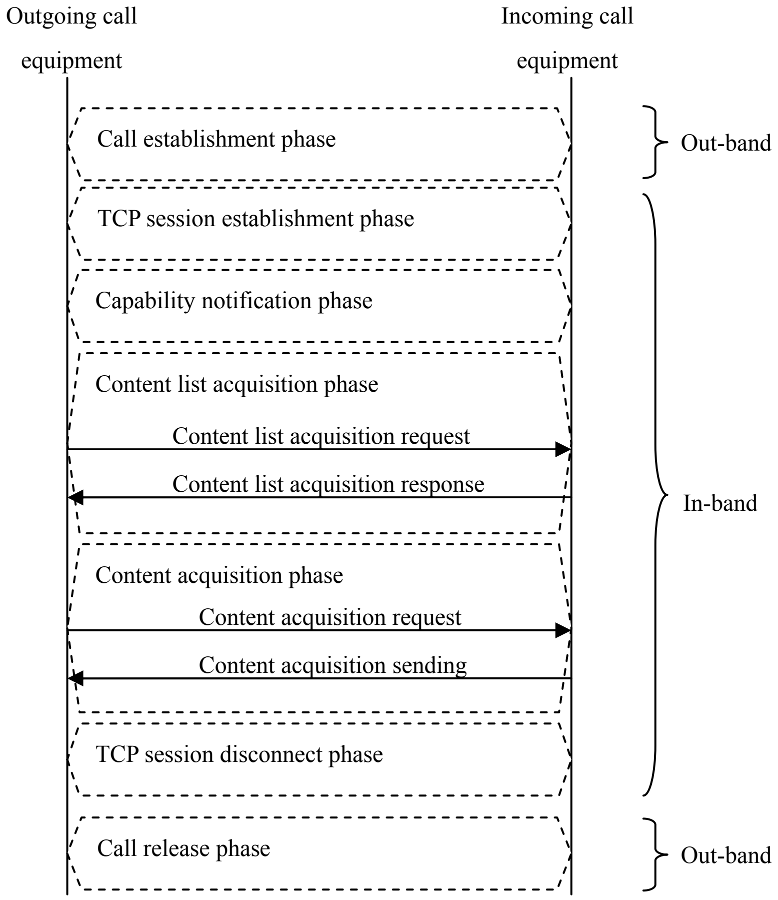

Outgoing call equipment
Incoming call equipment
Call establishment phase
Out-band
TCP session establishment phase
Capability notification phase
Content list acquisition phase
Content list acquisition request
Content list acquisition response
In-band
Content acquisition phase
Content acquisition request
Content acquisition sending
TCP session disconnect phase
Call release phase
Out-band

Figure 8-3/JJ-40.20: Connection mode 1

・Capability notification phase

After call establishment, the outgoing call equipment informs the incoming call equipment of its capabilities to receive content by type.

・Content list acquisition phase

In order to acquire a list of multiple items of content (content list) in the incoming call equipment, the outgoing call equipment requests the incoming call equipment to send the content list, and thus acquires the list.

・Content acquisition phase

In order to acquire the content selected from the content list, the outgoing call equipment issues a content request by the HTTP GET method and then acquires the content. When acquiring multiple items of content, the equipment shall repeat the content request by the HTTP GET method as many times as the number of items of content to be acquired.

### 8.2.2 Connection Mode 2 (Content Sending from Outgoing Call Equipment)

In this connection mode, the outgoing call equipment and the incoming call equipment shall be operated as a client and as a server, respectively, so that the former sends content to the latter. (For details on the designation of client and server operations, see Section 9.3.)

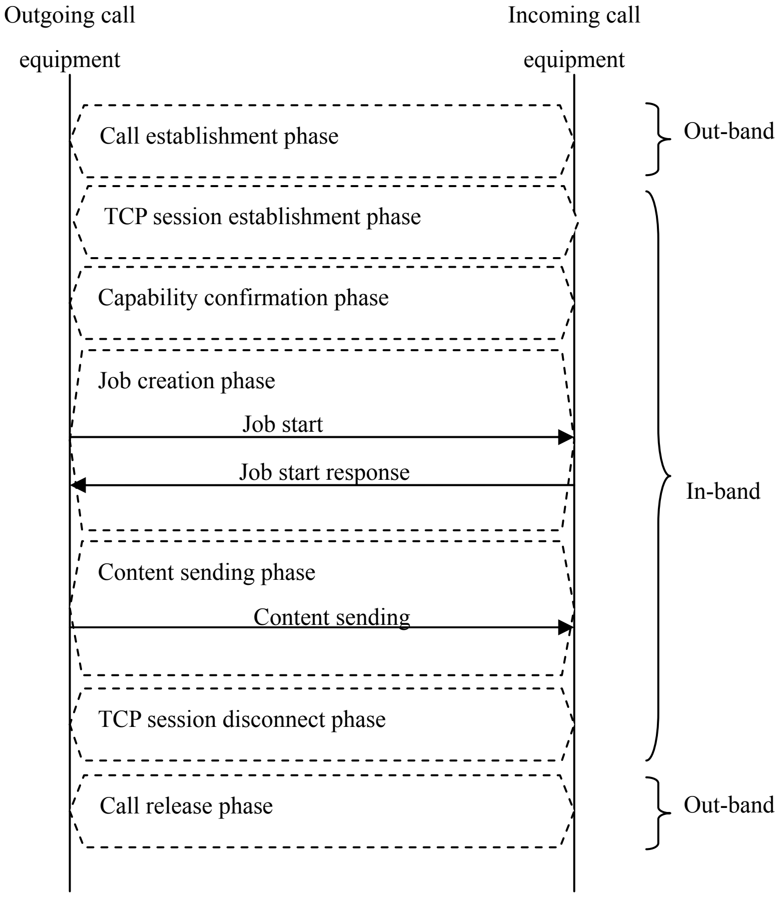


Figure 8-4/JJ-40.20: Connection mode 2

・Capability confirmation phase

After call establishment, the outgoing call equipment confirms the capabilities of its counterpart by issuing a request to identify the content types that can be received by the incoming call equipment.

・Job creation phase

The outgoing call equipment informs the incoming call equipment of the receiving conditions, such as the sizes and formats of the content to be sent, and then confirms that its counterpart is ready to receive the content.

・Content sending phase

The outgoing call equipment sends content to the incoming call equipment by the HTTP POST method. When sending multiple items of content, the outgoing call equipment shall repeat the process from the job creation phase to the content sending phase as many times as the number of items of content to be sent.

### 8.2.3 Connection Mode 3 (Content Sending from Incoming Call Equipment)

In this connection mode, the incoming call equipment and the outgoing call equipment shall be operated as a client and as a server, respectively, so that the former sends content to the latter. (For details on the designation of client and server operations, see Section 9.3.)

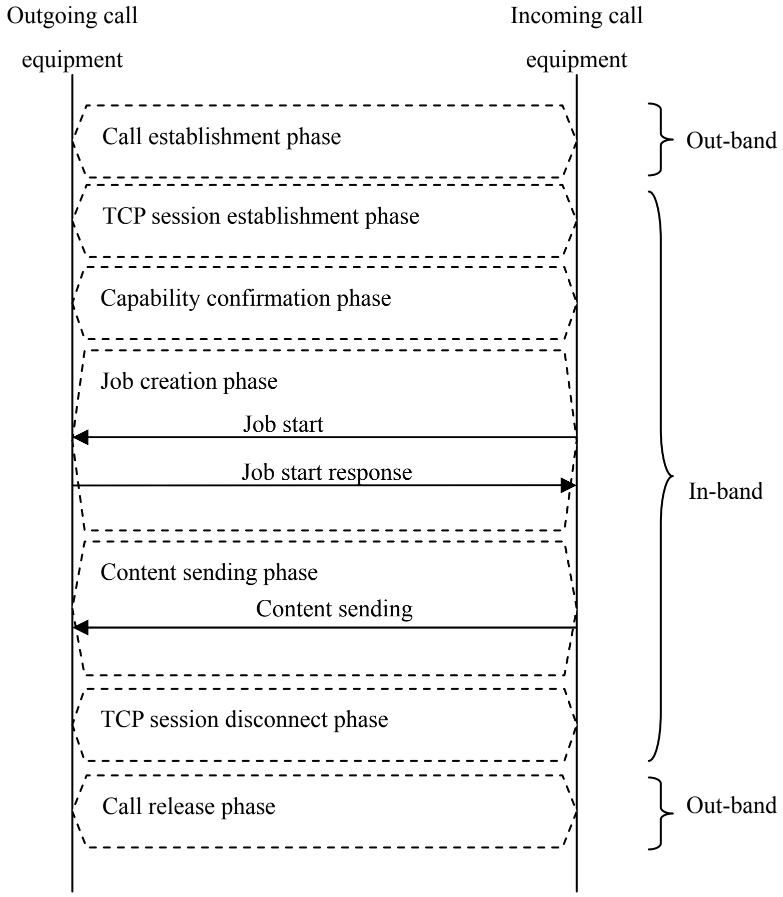


Figure 8-5/JJ-40.20: Connection mode 3

・Capability confirmation phase

After call establishment, the incoming call equipment that sends the content, issues a request to the outgoing call equipment that receives the content, to confirm the content types that can be received by the outgoing call equipment, and thus confirms the capabilities of its counterpart.

・Job creation phase

The incoming call equipment that sends content, informs the outgoing call equipment that receives the content, of the receiving conditions such as the sizes and formats of the content to be sent, and confirms that its counterpart is ready to receive the content.

・Content sending phase

The incoming call equipment that sends content, sends content to the outgoing call equipment that receives the content, by the HTTP POST method. When sending multiple items of content, the incoming call equipment shall repeat the process from the job creation phase to the content sending phase as many times as the number of items of content to be sent.

# 9 Details of Content Transfer Protocol

The in-band content transfer protocol, which is stipulated by this standard, consists of requests from the sending side to the receiving side, and responses from the receiving side to the sending side. These requests and responses use SOAP messages, and HTTP as the transfer protocol.

## 9.1 Version Notation

The version of this protocol shall be described in the tags of "ProtocolName" and "AgentName" for capability notification (InformCapability), capability notification response (InformCapabilityResponse), capability confirmation request (GetCapability), and capability confirmation response (GetCapabilityResponse). The details shall be described for the two types listed in Table 9-1. If the version is revised, backwards compatibility shall be ensured.

For details on the description methods, see Sections 10.2.1, 10.2.2, 10.3.1, and 10.3.2.

Table 9-1/JJ-40.20: Types of version to be described

| Type | Details |
|---|---|
| Protocol type | Represents the name and version of this protocol.<br>If the version is revised, backwards compatibility shall be ensured. |
| Equipment type | Represents the name and version of the equipment (including an application) implementing this protocol. |

Table 9-2/JJ-40.20: Setting details

| Type | Item | Setting details |
|---|---|---|
| Protocol type | Protocol name | Set "ContentsTransfer" |
| | Protocol version | Set "1.0" |
| Equipment type | Equipment name (Application name) | Can be freely set, but must conform to RFC2616. |
| | Equipment version (Application version) | Can be freely set, but must conform to RFC2616. |

Table 9-3/JJ-40.20: Setting details on version number of JJ-40.20 and protocol identification

| Version number of JJ-40.20 | Protocol type | Setting details |
|---|---|---|
| Ver. 1.0 | Protocol name | Set "ContentsTransfer" |
| | Protocol version | Set "1.0" |

## 9.2 How to Handle the Version

### 9.2.1 Protocol Identification

If the version of the protocol type is revised, backwards compatibility shall be ensured.

If the protocol's version contained in the protocol identification of the tag "ProtocolName" differs between capability notification (InformCapability) and capability notification response (InformCapabilityResponse), or between capability confirmation request (GetCapability) and capability confirmation response (GetCapabilityResponse), the protocol of a lower version shall be used.

### 9.2.2 Equipment Identification

For the equipment type, the name of the equipment implementing this protocol can be freely set.

If the equipment type in the tag of "AgentName" is deemed to be the same between capability notification (InformCapability) and capability notification response (InformCapabilityResponse), or between capability confirmation request (GetCapability) and capability confirmation response (GetCapabilityResponse), a SOAP action that can be used between the sending equipment and receiving equipment may be defined privately. Add "x-" or "X-" to the top of a SOAP message that is defined privately. (See Section 10.1.3.)

A child element for a tag of "Private" is provided to a SOAP action that is specified by this protocol. For this child element, a variable (tag) that can be used between the sending equipment and receiving equipment may be defined privately. Add "x-" or "X-" to the top of the name of the variable (tag) that is defined privately. (See Section 10.1.4.)

In the case there being different names for the equipment (application name), the privately defined SOAP action and the variable (tag) that is privately defined for the child element for a tag of "Private" shall be ignored, and the operation specified by this protocol shall be performed.

## 9.3 Operations of Outgoing Call Equipment and Incoming Call Equipment

Outgoing call equipment and incoming call equipment shall operate as a client and a server, respectively, or vice versa. Whether the equipment operates as a client or server shall be determined based on SDP (a line) of SIP.

Table 9-4/JJ-40.20: Operations of outgoing call equipment and incoming call equipment

| Out-band | | | |
|---|---|---|---|
| Outgoing call equipment | | Incoming call equipment | |
| SDP a line | Operation | SDP a line | Operation |
| Set to<br>a=setup:active | Operating as a client | Set to<br>a=setup:passive | Operating as a server |
| Set to<br>a=setup:passive | Operating as a server | Set to<br>a=setup:active | Operating as a client |

## 9.4 Creation of Request Message

### 9.4.1 Setting HTTP Header

The HTTP version to be used shall be HTTP 1.1 because persistent connection is used.

Table 9-5/JJ-40.20: Request’s HTTP header

| Variable | | | Setting details |
|---|---|---|---|
| Request-Line | | | |
| | Method | | ・Set "POST" |
| | Request-URI | | ・Set "/soap_action" |
| | HTTP-Version | | ・Set "HTTP/1.1" |
| Request-Header | | | |
| | Accept | | ・When setting this header, set "application/soap+xml" |
| | Accept-Encoding | | ・Do not set this header. |
| | Expect | | ・When setting this header, set "100-continue" |
| | Host | | ・When the destination’s FQDN is unknown, set the IP address to the host part.<br>・Set the destination’s listening port number for the HTTP server to the port number. |
| | User-Agent | | ・Do not set this header. *1 |
| General-Header | | | |
| | Connection | | ・When subsequently sending an HTTP request, set "Keep-Alive"<br>・Otherwise, set "Close" |
| | Date | | ・Set a time at which to create an HTTP request. |
| Entity-Header | | | |
| | Content-Encoding | | ・Do not set this header. |
| | Content-Type | | ・Set "application/soap+xml" |
| | | charset | ・Set "utf-8" |
| | | action | ・Do not set this parameter. |
| | Content-Length | | ・Set the size of HTTP-Body (byte count). |

*1: See Appendix A.

#### 9.4.1.1 Temporary Response Request

Since this protocol uses HTTP1.1, client equipment can add "Expect: 100-Continue" to the HTTP header of a request. When receiving "Expect: 100-Continue", server equipment shall return "HTTP/1.1 100 Continue". For details on the sequence, see Section 9.7.

Furthermore, the SOAP action shall be performed for the session continue request (Sections 10.2.8 and 10.3.7).

### 9.4.2 Setting SOAP Message

The SOAP version to be used shall be SOAP 1.2.

Unless otherwise noted, the text encoding that can be used in a SOAP message shall be UTF-8.

Table 9-6/JJ-40.20: Request’s SOAP message

| Setting details | | | Remarks |
|---|---|---|---|
| <?xml version="1.0" encoding=" UTF-8" ?> | | | |
| <env:Envelope<br>xmlns:env="http://www.w3.org/2003/05/soap-envelope"<br>xmlns:ct="http://www.ttc.or.jp/mmsys/ct"> | | | |
| | <env:Body> | | |
| | | <request-method-name<br>env:encodingStyle="http://www.w3.org/2003/05/soap-encoding"<br>xmlns:ct="http://www.ttc.or.jp/mmsys/ct"> | Set the name of a request method to request-method-name. |
| | | < (argument)> | |

#### 9.4.2.1 Namespace

This protocol defines the namespace of an XML element. For details on the namespace, see Section 4.2.2.

## 9.5 Creation of Response Message

### 9.5.1 Setting HTTP Header

The HTTP version to be used shall be HTTP 1.1 because persistent connection is used.

Table 9-7/JJ-40.20: Response’s HTTP header

| Variable | | | Setting details |
|---|---|---|---|
| Status-Line | | | |
| | HTTP-Version | | ・Set "HTTP/1.1" |
| | Status-Code | | ・Set "100" |
| | Reason-Phrase | | ・Set "Continue" |
| Request-Header | | | |
| | Server | | ・Do not set this header.*1 |
| General-Header | | | |
| | Connection | | ・Set "Keep-Alive" |
| | Date | | ・Set the time at which to create an HTTP response. |
| Entity-Header | | | |
| | Content-Encoding | | ・Do not set this header. |
| | Content-Type | | ・Set "application/soap+xml" |
| | | charset | ・Set "utf-8" |
| | | action | ・Do not set this parameter. |
| | Content-Length | | ・Set the size of HTTP-Body (byte count). |

*1: See Appendix A.

#### 9.5.1.1 Temporary Response

When "Expect: 100-Continue" is added to the HTTP header of a request from the client equipment, the server equipment shall return a response "HTTP/1.1 100 Continue". For details on the sequence, see Section 9.7.
Furthermore, the SOAP action shall be performed for the session continue request (Sections 10.2.8 and 10.3.7).

### 9.5.2 Setting SOAP Message

The SOAP version to be used shall be SOAP 1.2.

Unless otherwise noted, the text encoding that can be used in a SOAP message shall be UTF-8.

Table 9-8/JJ-40.20: Response’s SOAP message

| Setting details | Remarks |
|---|---|
| <?xml version="1.0" encoding="UTF-8" ?> | |
| <env:Envelope<br>xmlns:env="http://www.w3.org/2003/05/soap-envelope"<br>xmlns:ct="http://www.ttc.or.jp/mmsys/ct"> | |
| <env:Body> | |
| <request-method-name<br>env:encodingStyle="http://www.w3.org/2003/05/soap-encoding"<br>xmlns:ct="http://www.ttc.or.jp/mmsys/ct"> | Set the name of a request method to request-method-name. |
| <(argument)> | |

#### 9.5.2.1 Namespace

This protocol defines the namespace of an XML element. For details on the namespace, see Section 4.2.2.

## 9.6 Request for and Acquisition of SOAP Action

If the equipment type is deemed to be the same as the equipment type in the "AgentName" tag between capability notification (InformCapability) and capability notification response (InformCapabilityResponse), or between capability confirmation request (GetCapability) and capability confirmation response (GetCapabilityResponse), a SOAP action that can be used between the sending equipment and receiving equipment may be defined privately and used. If a SOAP action is added in the future, it must be identified between the sending equipment and receiving equipment. For this purpose, a SOAP action list request shall be issued as a means of understanding the SOAP action of server equipment.

The sending of "/soap_action" by the HTTP GET method shall be the SOAP action list request.

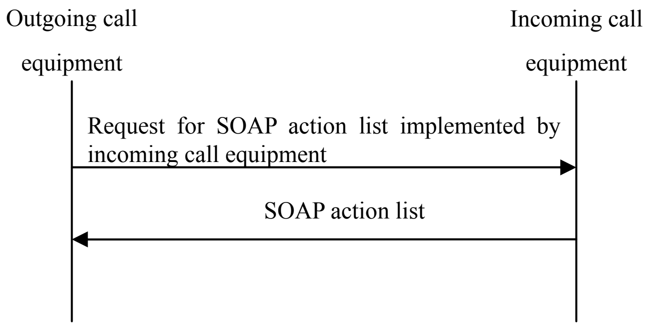

Outgoing call equipment
Incoming call equipment
Request for SOAP action list implemented by incoming call equipment
SOAP action list

Figure 9-1/JJ-40.20: Method for requesting SOAP action list

## 9.7 Sequence of Connection Modes

The in-band content transfer protocol, which uses the SOAP action specified by this standard, has three connection modes. This section describes the sequence of the modes.

In-band connection mode 1: Content acquisition by outgoing call equipment

In-band connection mode 2: Content sending from outgoing call equipment

In-band connection mode 3: Content sending from incoming call equipment

### 9.7.1 Content Acquisition by Outgoing Call Equipment

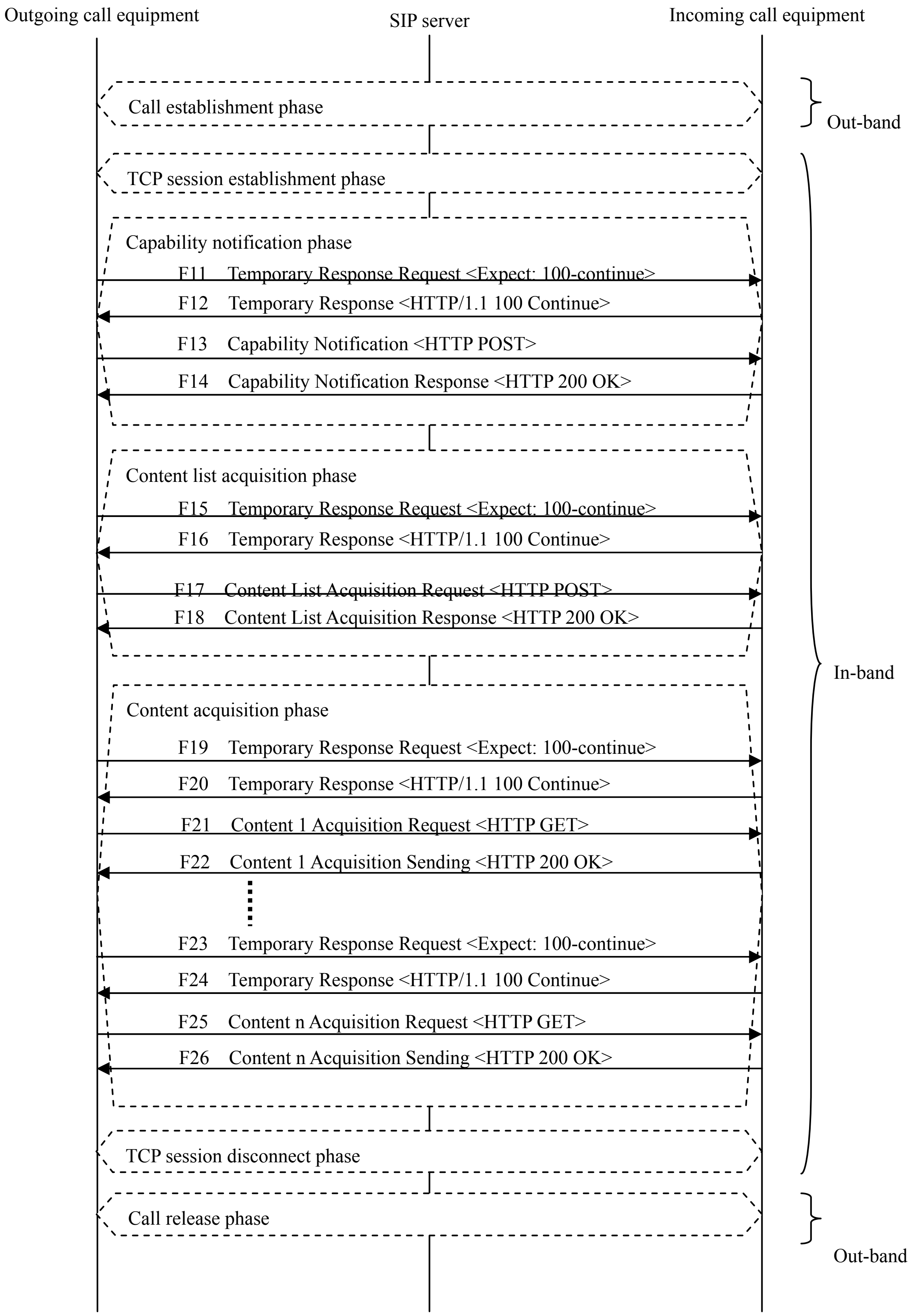

Outgoing call equipment
SIP server
Incoming call equipment
Call establishment phase
Out-band
TCP session establishment phase
Capability notification phase
F11 Temporary Response Request <Expect: 100-continue>
F12 Temporary Response <HTTP/1.1 100 Continue>
F13 Capability Notification <HTTP POST>
F14 Capability Notification Response <HTTP 200 OK>
Content list acquisition phase
F15 Temporary Response Request <Expect: 100-continue>
F16 Temporary Response <HTTP/1.1 100 Continue>
F17 Content List Acquisition Request <HTTP POST>
F18 Content List Acquisition Response <HTTP 200 OK>
In-band
Content acquisition phase
F19 Temporary Response Request <Expect: 100-continue>
F20 Temporary Response <HTTP/1.1 100 Continue>
F21 Content 1 Acquisition Request <HTTP GET>
F22 Content 1 Acquisition Sending <HTTP 200 OK>
F23 Temporary Response Request <Expect: 100-continue>
F24 Temporary Response <HTTP/1.1 100 Continue>
F25 Content n Acquisition Request <HTTP GET>
F26 Content n Acquisition Sending <HTTP 200 OK>
TCP session disconnect phase
Call release phase
Out-band

Figure 9-2/JJ-40.20: In-band connection mode 1: Sequence diagram for content acquisition by outgoing call equipment

### 9.7.2 Content Sending from Outgoing Call Equipment

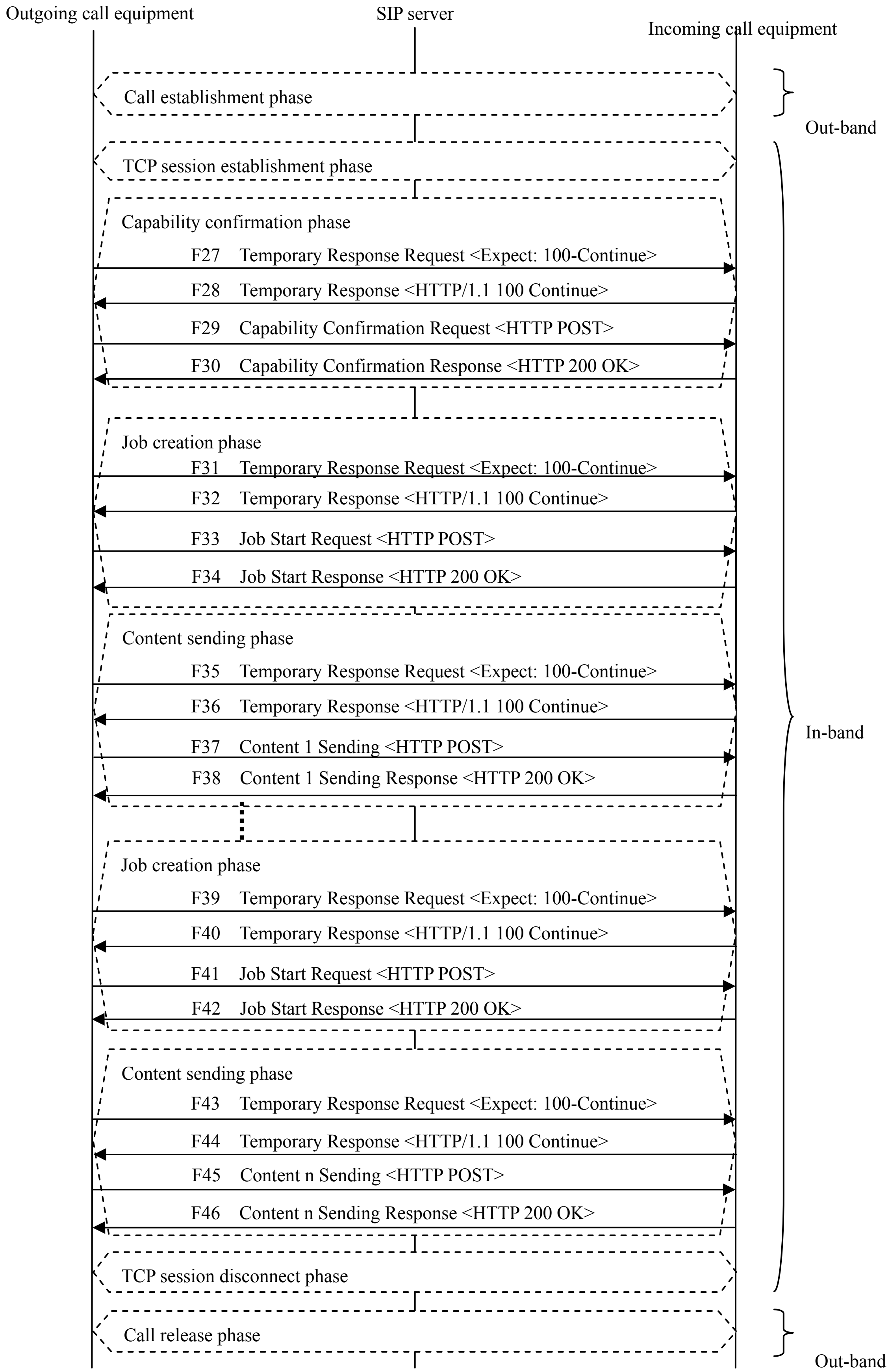
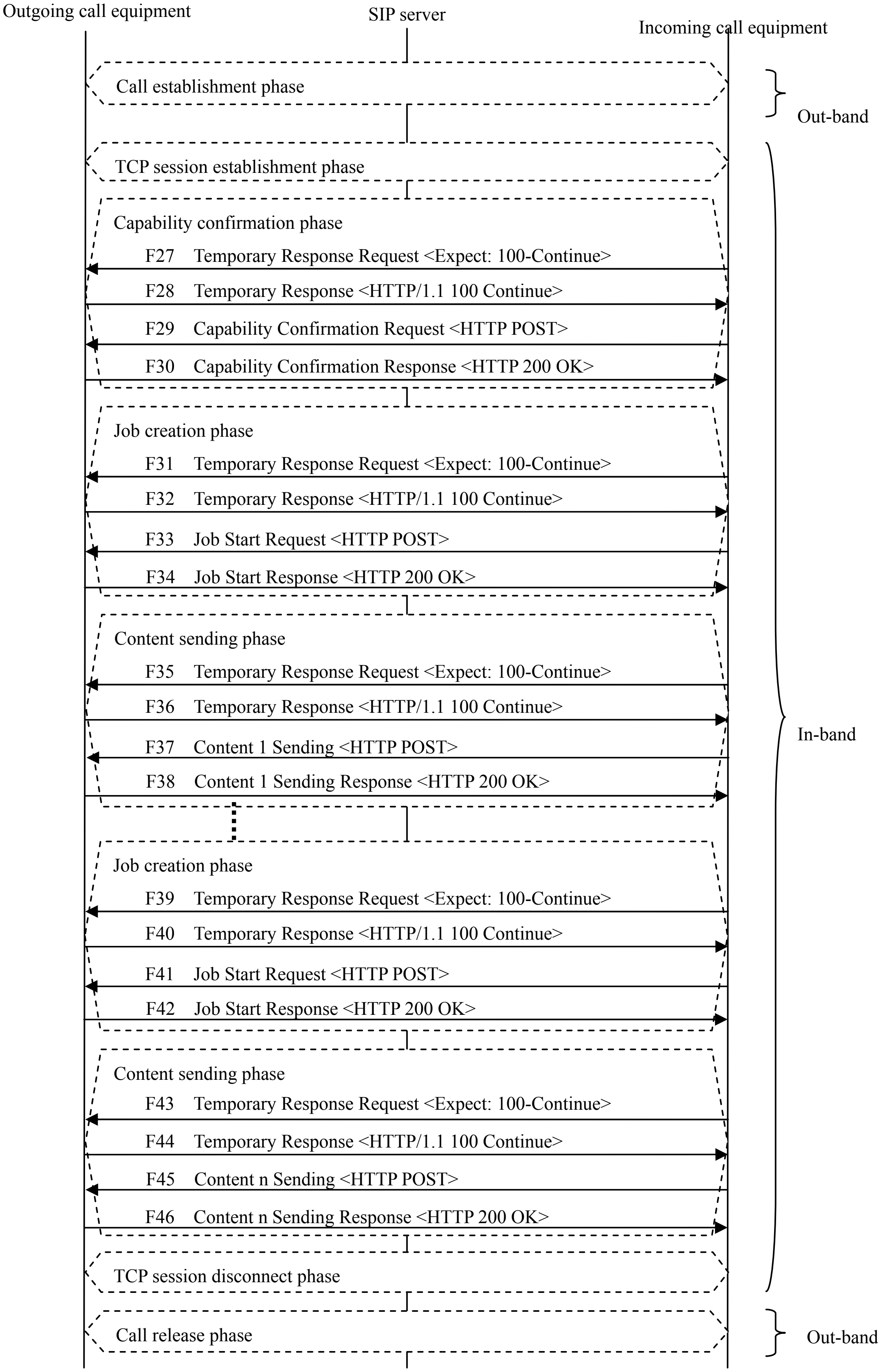

Outgoing call equipment
SIP server
Incoming call equipment
Call establishment phase
Out-band
TCP session establishment phase
Capability confirmation phase
F27 Temporary Response Request <Expect: 100-Continue>
F28 Temporary Response <HTTP/1.1 100 Continue>
F29 Capability Confirmation Request <HTTP POST>
F30 Capability Confirmation Response <HTTP 200 OK>
Job creation phase
F31 Temporary Response Request <Expect: 100-Continue>
F32 Temporary Response <HTTP/1.1 100 Continue>
F33 Job Start Request <HTTP POST>
F34 Job Start Response <HTTP 200 OK>
Content sending phase
F35 Temporary Response Request <Expect: 100-Continue>
F36 Temporary Response <HTTP/1.1 100 Continue>
F37 Content 1 Sending <HTTP POST>
F38 Content 1 Sending Response <HTTP 200 OK>
In-band
Job creation phase
F39 Temporary Response Request <Expect: 100-Continue>
F40 Temporary Response <HTTP/1.1 100 Continue>
F41 Job Start Request <HTTP POST>
F42 Job Start Response <HTTP 200 OK>
Content sending phase
F43 Temporary Response Request <Expect: 100-Continue>
F44 Temporary Response <HTTP/1.1 100 Continue>
F45 Content n Sending <HTTP POST>
F46 Content n Sending Response <HTTP 200 OK>
TCP session disconnect phase
Call release phase
Out-band

Figure 9-3/JJ-40.20: In-band connection mode 2: Sequence diagram for content sending from outgoing call equipment

### 9.7.3 Content Sending from Incoming Call Equipment



Outgoing call equipment
SIP server
Incoming call equipment
Call establishment phase
Out-band
TCP session establishment phase
Capability confirmation phase
F27 Temporary Response Request <Expect: 100-Continue>
F28 Temporary Response <HTTP/1.1 100 Continue>
F29 Capability Confirmation Request <HTTP POST>
F30 Capability Confirmation Response <HTTP 200 OK>
Job creation phase
F31 Temporary Response Request <Expect: 100-Continue>
F32 Temporary Response <HTTP/1.1 100 Continue>
F33 Job Start Request <HTTP POST>
F34 Job Start Response <HTTP 200 OK>
Content sending phase
F35 Temporary Response Request <Expect: 100-Continue>
F36 Temporary Response <HTTP/1.1 100 Continue>
F37 Content 1 Sending <HTTP POST>
F38 Content 1 Sending Response <HTTP 200 OK>
In-band
Job creation phase
F39 Temporary Response Request <Expect: 100-Continue>
F40 Temporary Response <HTTP/1.1 100 Continue>
F41 Job Start Request <HTTP POST>
F42 Job Start Response <HTTP 200 OK>
Content sending phase
F43 Temporary Response Request <Expect: 100-Continue>
F44 Temporary Response <HTTP/1.1 100 Continue>
F45 Content n Sending <HTTP POST>
F46 Content n Sending Response <HTTP 200 OK>
TCP session disconnect phase
Call release phase
Out-band

Figure 9-4/JJ-40.20: In-band connection mode 3: Sequence diagram for content sending from incoming call equipment

# 10 Specification of SOAP Action

## 10.1 Notes

### 10.1.1 Handling of M/O for Variable (Tag)

"M/O" is specified for a variable (tag) of each SOAP message where "M" is an abbreviation for Mandatory and set to a mandatory tag, while "O" is an abbreviation for Option and set to an option tag.

If equipment receives a SOAP action with an option tag, the equipment shall implement the action based on the setting in the option tag when it can interpret the tag. But when it cannot interpret the tag, it shall ignore the action.

### 10.1.2 Handling of Variable (Tag) without Value

Even when the mandatory tag of "M" does not have any value, a tag without value (empty tag) shall be included in a SOAP message.

When the option tag of "O" does not have any value, it is unnecessary to include the tag in the SOAP message.

### 10.1.3 Private Extension of SOAP Action

On the assumption that the equipment type in the tag of "AgentName" is the same between capability notification (InformCapability) and capability notification response (InformCapabilityResponse), or between capability confirmation request (GetCapability) and capability confirmation response (GetCapabilityResponse), a SOAP action that can be used between the sending equipment and receiving equipment may be defined privately. Add "x-" or "X-" to the top of a SOAP message that is defined privately.

### 10.1.4 Private Extension of Tag

In the SOAP action specified by this protocol, a "Private" tag is provided as one of the variables (tags).

On the assumption that the equipment type in the tag of "AgentName" is the same between capability notification (InformCapability) and capability notification response (InformCapabilityResponse), or between capability confirmation request (GetCapability) and capability confirmation response (GetCapabilityResponse), a variable (tag) that can be used between the sending equipment and receiving equipment may be defined privately for a child element of this "Private" tag. Add "x-" or "X-" to the top of the variable (tag) that is defined privately.

## 10.2 Request Message

### 10.2.1 Capability Notification (InformCapability)

For content acquisition, capability notification is used by the client equipment to inform the server equipment of the content types that can be handled by the client equipment. Capability notification is also used for notification and confirmation of the protocol type (protocol name/protocol version) and the equipment type (equipment name (application name)/equipment version (application version)) between client equipment and server equipment.

Tale 10-1/JJ-40.20: Capability notification (InformCapability)

| Variable (Tag) | | M/O | Details |
|---|---|---|---|
| <ct:InformCapability> | | M | － |
| | <ct:ProtocolName> | M | ・Describe the protocol type (protocol name/protocol version), using single-byte alphanumeric characters. |
| | <ct:AgentName> | M | ・Describe the equipment type (equipment name (application name)/equipment version (application version)), using single-byte alphanumeric characters.<br>・Multiple equipment types can be described. |
| | <ct:SupportedFormats> | M | ・Set all supported MIME-Types in comma-separated format. |
| | <Private> | O | ・Shall be used for private extension. |

* "M"= Mandatory, "O" = Option

#### 10.2.1.1 <ct:ProtocolName>

This tag describes the protocol type (protocol name/protocol version) of the client equipment. (See Section 9.2.1.)

■Description method
protocol-name/protocol-version

Example:
protocol-name/protocol-version ⇒ ContentsTransfer/1.0

### 10.2.1.2 <ct:AgentName>

This tag describes the equipment type (equipment name (application name)/equipment version (application version)) of the client equipment. When describing multiple equipment types, describe them in comma-separated format. (See Section 9.2.2.)

■Description method

・One equipment type: equipment-name/equipment-version

・Two equipment types: equipment-name/equipment-version,equipment-name/equipment-version

Example:

One equipment type: ContentsTransferApplication/1.0

Two equipment types: ContentsTransferApplication/1.0,ContentsTransferEquipment/1.0

### 10.2.1.3 <ct:SupportedFormats>

In principle, describe all available capabilities in IANA-registered (see http://www.iana.org/assignments/media-types/) MIME-Type notation and in comma-separated CSV list format. Conform to the following description rules:

(1) A regular expression shall support only one character "*". (Example: "ima*" etc. are not supported)

(2) "*" (asterisk) means "all".

(3) When setting multiple formats, separate them with commas.

(4) Give priority to a format that is described earlier.

(5) "!" (exclamation mark) shall be treated as "negative (NOT)".

(6) Place "!" at the top of MIME-Type.

Example:

| | | |
|---|---|---|
| Support all file formats | ⇒ | */* |
| Not support PNG image format | ⇒ | !image/png |
| Support images, but not moving pictures | ⇒ | image/*,!video/* |

### 10.2.1.4 <Private>

A private tag can be set for a child element. (See Section 10.1.4.)

### 10.2.2 Capability Confirmation Request (GetCapability)

When client equipment sends content to server equipment, a capability confirmation request is used to confirm the content types that can be handled by the server equipment. A capability confirmation request is also used for notification and confirmation of the protocol type (protocol name/protocol version) and the equipment type (equipment name (application name)/equipment version (application version)) between client equipment and server equipment.

Table 10-2/JJ-40.20: Capability confirmation request (GetCapability)

| Variable | | M/O | Details |
|---|---|---|---|
| <ct:GetCapability> | | M | － |
| | <ct:ProtocolName> | M | ・Describe the protocol type (protocol name/protocol version), using single-byte alphanumeric characters. |
| | <ct:AgentName> | M | ・Describe the equipment type (equipment name (application name)/equipment version (application version)), using single-byte alphanumeric characters.<br>・Multiple equipment types can be described. |
| | <Options> | O | ・Will be defined in Annex. |
| | <Private> | O | ・Shall be used for private extension. |

* "M"= Mandatory, "O" = Option

#### 10.2.2.1 <ct:ProtocolName>

Describe the protocol type (protocol name/protocol version) of the client equipment. (See Section 9.2.1.)

■Description method

protocol-name/protocol-version

Example:

protocol-name/protocol-version ⇒ ContentsTransfer/1.0

#### 10.2.2.2 <ct:AgentName>

Describe the equipment type (equipment name (application name)/equipment version (application version)) of the client equipment. When describing multiple equipment types, describe them in comma-separated format. (See Section 9.2.2.)

■Description method

・One equipment type: equipment-name/equipment-version

・Two equipment types: equipment-name/equipment-version,equipment-name/equipment-version

Example:

One equipment type ⇒ ContentsTransferApplication/1.0

Two equipment types ⇒ ContentsTransferApplication/1.0,ContentsTransferEquipment/1.0

#### 10.2.2.3 <Options>

Will be defined in Annex.

#### 10.2.2.4 <Private>

A private tag can be set for a child element. (See Section 10.1.4.)

### 10.2.3 Content List Acquisition Request (GetContentsList)

The client equipment issues a request to the server equipment for acquiring a list of content that is owned by the server equipment.

Table 10-3/JJ-40.20: Content list acquisition request (GetContentsList)

| Variable | | M/O | Details |
|---|---|---|---|
| <ct:GetContentsList> | | M | — |
| | <Private> | O | ・Shall be used for private extension. |

* "M"= Mandatory, "O" = Option

#### 10.2.3.1 <Private>

A private tag can be set for a child element. (See Section 10.1.4.)

### 10.2.4 Job Start Request (CreateJob)

When the client equipment sends content to the server equipment, the former informs the latter of the receiving conditions such as the sizes and formats of the content to be sent.

Table 10-4/JJ-40.20: Job start request (CreateJob)

| Variable | | M/O | Details |
|---|---|---|---|
| <ct:CreateJob> | | M | — |
| | <ct:ContentName> | M | ・Set the name of a file to be sent. |
| | <ct:ContentSize> | M | ・Set the size of the file to be sent, using a byte count (integer of up to 64 bits, represented in single-byte numbers). |
| | <ct:Format> | M | ・Set a MIME-Type of the file to be sent. |
| | <Options> | O | ・Will be defined in Annex. |
| | <Private> | O | ・Shall be used for private extension. |

* "M"= Mandatory, "O" = Option

#### 10.2.4.1 <ct:Format>

In principle, set IANA-registered MIME-Types (see http://www.iana.org/assignments/media-types/).

#### 10.2.4.2 <Options>

Will be defined in Annex.

#### 10.2.4.3 <Private>

A private tag can be set for a child element. (See Section 10.1.4.)

### 10.2.5 Content Acquisition Request

When content is acquired in connection mode 1, where the content has settings in the "Path" and "Name" tags of a content list acquisition response (GetContentsListResponse), a request to acquire the content set in the "Path" and "Name" tags for the content list acquisition response shall be made using the HTTP GET method. Consequently, a new SOAP action shall not be specified to perform a content acquisition request. When multiple items of content are acquired, the content acquisition request by the HTTP GET method shall be repeated as many times as the number of items of content to be acquired. （See Sequence Figure 9-2.）

### 10.2.6 Content Sending

Content sending shall be performed after job start (CreateJob) in connection modes 2 and 3. Client equipment shall send content to the server equipment by the HTTP POST method, in accordance with RFC1867 and RFC2388. Consequently, a new SOAP action shall not be specified to perform content sending. When multiple items of content are sent, the process of the job creation phase and the content sending phase shall be repeated as many times as the number of items of content to be sent. (See Sequence Figure 9-3 and Figure 9-4.)

### 10.2.7 Content Sending End (EndSendContent)

Upon the completion of content sending (Section 10.2.6), content sending end may be performed to provide explicit notification of the completion.

Or, "Close" shall be set to "Connection" in the HTTP header of the request message.

Table 10-5/JJ-40.20: Content sending end (EndSendContent)

| Variable | | M/O | Details |
|---|---|---|---|
| <ct:EndSendContent> | | M | — |
| | <Private> | O | ・Shall be used for private extension. |

* "M"= Mandatory, "O" = Option

#### 10.2.7.1 <Private>

A private tag can be set for a child element. (See Section 10.1.4.)

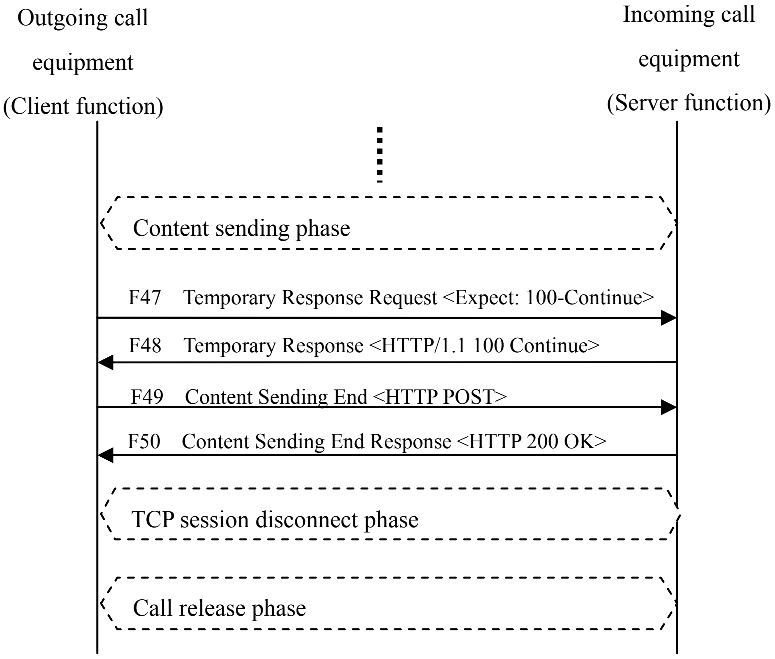

Outgoing call equipment (Client function)
Incoming call equipment (Server function)
Content sending phase
F47 Temporary Response Request <Expect: 100-Continue>
F48 Temporary Response <HTTP/1.1 100 Continue>
F49 Content Sending End <HTTP POST>
F50 Content Sending End Response <HTTP 200 OK>
TCP session disconnect phase
Call release phase

Figure 10-1/JJ-40.20: Example of content sending end sequence

### 10.2.8 Session Continue Request (ContinueSession)

When a user performs content selection, the interval of file sending may widen and the interval of SOAP message communication may widen, and therefore a TCP session timer value in the "SessionTime" tag for capability notification response (InformCapabilityResponse) and capability confirmation response (GetCapabilityResponse) may be exceeded. In this case, this session continue request shall be made in order to continue the in-band TCP session before the TCP session timer value is exceeded. If the client equipment does not explicitly issue a direction to disconnect the HTTP session, the server equipment shall not disconnect the relevant HTTP session to allow for a repetition of a content job start request from the client equipment.

Table 10-6/JJ-40.20: Session continue request (ContinueSession)

| Variable | | M/O | Details |
|---|---|---|---|
| <ct:ContinueSession> | | M | — |
| | <Private> | O | ・Shall be used for private extension. |

* "M"= Mandatory, "O" = Option

#### 10.2.8.1 <Private>

A private tag can be set for a child element. (See Section 10.1.4.)

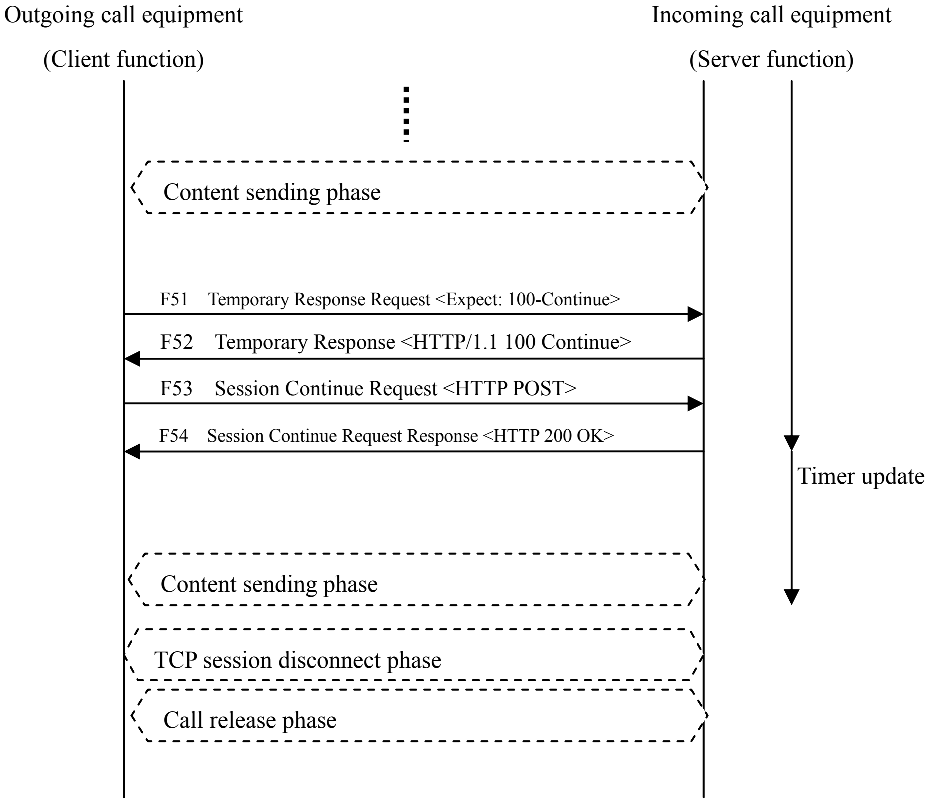

Outgoing call equipment
(Client function)
Incoming call equipment
(Server function)
Content sending phase
F51 Temporary Response Request <Expect: 100-Continue>
F52 Temporary Response <HTTP/1.1 100 Continue>
F53 Session Continue Request <HTTP POST>
F54 Session Continue Request Response <HTTP 200 OK>
Timer update
Content sending phase
TCP session disconnect phase
Call release phase

Figure 10-2/JJ-40.20: Example of session continue sequence

## 10.3 Response Message

### 10.3.1 Capability Notification Response (InformCapabilityResponse)

Upon receiving capability notification (InformCapability) for content acquisition, the server equipment responds to the client equipment with the protocol type (protocol name/protocol version), equipment type (equipment name (application name)/equipment version (application version)), and TCP session timer value of the server equipment.

Table 10-7/JJ-40.20: Capability notification response (InformCapabilityResponse)

| Variable | | M/O | Details |
|---|---|---|---|
| <ct:InformCapabilityResponse> | | M | － |
| | <ct:ProtocolName> | M | ・Describe the protocol type (protocol name/protocol version), using single-byte alphanumeric characters. |
| | <ct:AgentName> | M | ・Describe the equipment type (equipment name (application name)/equipment version (application version)), using single-byte alphanumeric characters.<br>・Multiple equipment types can be described. |
| | <ct:SessionTime> | M | ・Describe TCP session timer value. |
| | <Private> | O | ・Shall be used for private extension. |

* "M"= Mandatory, "O" = Option

#### 10.3.1.1 <ct:ProtocolName>

Describe the protocol type (protocol name/protocol version) of the server equipment. (See Section 9.2.1.)

■Description method

protocol-name/protocol-version

Example:

protocol-name/protocol-version ⇒ ContentsTransfer/1.0

#### 10.3.1.2 <ct:AgentName>

Describe the equipment type (equipment name (application name)/equipment version (application version)) of the server equipment. When describing multiple equipment types, describe them in comma-separated format. (See Section 9.2.2.)

■Description method

・One equipment type: equipment-name/equipment-version

・Two equipment types: equipment-name/equipment-version,equipment-name/equipment-version

Example:

One equipment type ⇒ ContentsTransferApplication/1.0

Two equipment types ⇒ ContentsTransferApplication/1.0,ContentsTransferEquipment/1.0

#### 10.3.1.3 <ct:SessionTime>

After the lapse of the TCP session timer value, the TCP session is disconnected by the server equipment. For instance, when the value is set to "150" , the TCP session is disconnected after 150 seconds. To continue the TCP session, the client equipment shall perform an action to cause the server equipment to continue the session before the lapse of the TCP session timer value (see Sections 10.2.8 and 10.3.7).

The TCP session timer value that is posted by the server equipment under this tag will be refreshed when the value is sent again.

The client equipment controls the TCP session timer based on the latest TCP session timer value that is posted by the server equipment under this tag.

#### 10.3.1.4 <Private>

A private tag can be set for a child element. (See Section 10.1.4.)

### 10.3.2 Capability Confirmation Response (GetCapabilityResponse)

Upon receiving a capability confirmation request (GetCapability) for content sending, the server equipment responds to the client equipment with the protocol type (protocol name/protocol version), equipment type (equipment name (application name)/equipment version (application version)), TCP session timer value, and the content types that can be handled.

Table 10-8/JJ-40.20: Capability confirmation response (GetCapabilityResponse)

| Variable | | M/O | Details |
|---|---|---|---|
| <ct:GetCapabilityResponse> | | M | － |
| | <ct:ProtocolName> | M | ・Describe the protocol type (protocol name/protocol version), using single-byte alphanumeric characters. |
| | <ct:AgentName> | M | ・Describe the equipment type (equipment name (application name)/equipment version (application version)), using single-byte alphanumeric characters.<br>・Multiple equipment types can be described. |
| | <ct:SessionTime> | M | ・Describe TCP session timer value. |
| | <ct:SupportedFormats> | M | ・Set all supported MIME-Types in comma-separated format. |
| | <Options> | O | ・Will be defined in Annex. |
| | <Private> | O | ・Shall be used for private extension. |

* "M"= Mandatory, "O" = Option

#### 10.3.2.1 <ct:ProtocolName>

Describe the protocol type (protocol name/protocol version) of the server equipment. (See Section 9.2.1.)

■Description method

protocol-name/protocol-version

Example:

protocol-name/protocol-version ⇒ ContentsTransfer/1.0

### 10.3.2.2 <ct:AgentName>

Describe the equipment type (equipment name (application name)/equipment version (application version)) of the server equipment. When describing multiple equipment types, describe them in comma-separated format. (See Section 9.2.2.)

■Description method

- One equipment type: equipment-name/equipment-version
- Two equipment types: equipment-name/equipment-version,equipment-name/equipment-version

Example:

| | | |
|---|---|---|
| One equipment type | ⇒ | ContentsTransferApplication/1.0 |
| Two equipment types | ⇒ | ContentsTransferApplication/1.0,ContentsTransferEquipment/1.0 |

### 10.3.2.3 <ct:SessionTime>

After the lapse of the TCP session timer value, the TCP session is disconnected by the server equipment. For instance, when the value is set to "150" , the TCP session is disconnected after 150 seconds. To continue the TCP session, the client equipment shall perform an action to cause the server equipment to continue the session before the lapse of the TCP session timer value (see Sections 10.2.8 and 10.3.7).

The TCP session timer value that is posted by the server equipment under this tag will be refreshed when the value is sent again.

The client equipment controls the TCP session timer based on the latest TCP session timer value that is notified by the server equipment under this tag.

### 10.3.2.4 <ct:SupportedFormats>

In principle, describe all available capabilities in IANA-registered (see http://www.iana.org/assignments/media-types/) MIME-Type notation and in comma-separated CSV list format. Conform to the following description rules:

(1) A regular expression shall support only one character "*". (Example: "ima*" etc. are not supported)
(2) "*" (asterisk) means "all".
(3) When setting multiple formats, separate them with commas.
(4) Give priority to a format that is described earlier.
(5) "!" (exclamation mark) shall be treated as "negative (NOT)".
(6) Place "!" at the top of MIME-Type.

Example:

| | | |
|---|---|---|
| Support all file formats | ⇒ | */* |
| Not support PNG image format | ⇒ | !image/png |
| Support images, but not moving pictures | ⇒ | image/*,!video/* |

### 10.3.2.5 <Options>

Will be defined in Annex.

#### 10.3.2.6 <Private>

A private tag can be set for a child element. (See Section 10.1.4.)

### 10.3.3 Content List Acquisition Response (GetContentsListResponse)

Upon receiving a content list acquisition request (GetContentsList), the server equipment sends its content list to the client equipment.

Table 10-9/JJ-40.20: Content list acquisition response (GetContentsListResponse)

| Variable | M/O | Details | XML document *2 |
|---|---|---|---|
| <ct:GetContentsListResponse> | M | ― | |
| <ct:ContentsList> | M | ― | |
| <ct:Contents> | M | ― | |
| <ct:Content> | M | ― | Applicable |
| <ct:Path> | M | ・Set a path to the content. | Applicable |
| <ct:Name> | M | ・Set the file name. | Applicable |
| <ct:Size> | M | ・Set the file size, using a byte count (integer of up to 64 bits, represented in single-byte numbers). | Applicable |
| <ct:Format> | M | ・Set the MIME-Type of the file to be sent. | Applicable |
| <ct:Title> | O | ・Set the title of the file. | Applicable |
| <ct:Description> | O | ・Set the text that describes the content. | Applicable |
| <Private> | O | ・Shall be used for private extension. | |

* "M"= Mandatory, "O" = Option

*2: Described as XML document.

#### 10.3.3.1 <ct:ContentsList>

This tag represents a list of content to which a child element is set.

The child element, as in Sections 10.3.3.2 to 10.3.3.6, shall be described within a SOAP action as an XML document.

Note that an XML message described within the SOAP action has characters that must be escaped in accordance with [XML] in Chapter 2.

#### 10.3.3.2 <ct:Contents>

This is a tag to describe listed content as an XML document to which a child element is set.

#### 10.3.3.3 <ct:Content>

This tag represents listed content to which a child element is set.

It shall be possible to set multiple tags as a child element of the "Contents" tag.

#### 10.3.3.4 <ct:Format>

Sets the MIME-Type of a content file.

In principle, set IANA-registered (see http://www.iana.org/assignments/media-types/) MIME-Types.

#### 10.3.3.5 <ct:Title>

Sets the title of a content file. It shall also be possible not to set the title.

When not setting the title, or when a value of the "Title" tag is empty, always set an empty tag to the "Content" tag. (See Section 10.1.2.)

#### 10.3.3.6 <ct:Description>

Sets the text that describes the content. It shall also be possible not to set the text.

When not setting the text, or when a value of the "Title" tag is empty, always set an empty tag to the "Content" tag. (See Section 10.1.2.)

#### 10.3.3.7 <Private>

A private tag can be set for a child element. (See Section 10.1.4.)

### 10.3.4 Job Start Response (CreateJobResponse)

For a job start request (CreateJob), the server equipment informs the client equipment of the JobID and destination path when receiving content.

Table 10-10/JJ-40.20: Job start response (CreateJobResponse)

| Variable | | M/O | Details |
|---|---|---|---|
| <ct:CreateJobResponse> | | M | － |
| | <ct:JobID> | M | ・Set a job identifier (integer of up to 32 bits, represented in single-byte numbers). |
| | <ct:Path> | M | ・Set a destination Request-URI of the HTTP POST method for file sending. |
| | <Options> | O | ・Will be defined in Annex. |
| | <Private> | O | ・Shall be used for private extension. |

* "M"= Mandatory, "O" = Option

#### 10.3.4.1 <ct:JobID>

The "JobID" tag is used when responding to a job start request (CreateJob).

When it is possible to receive the request under the receiving conditions such as sizes and formats of the file posted in the request, use the "JobID" tag with jobID set when responding to the request.

When it is impossible to receive the request under the conditions posted in the request, use the JobID with the following ID (error code) to reject the reception:

-1 = Reception rejected because of improper file name

-2 = Reception rejected because of over capacity

-3 = Reception rejected because of unsupported file format

-4 = Reception preferably rejected because the "Options" and "Private" tags have incomprehensible values
(To ignore the values rather than rejecting the reception, set JobID instead of -4)

#### 10.3.4.2 <Options>

Will be defined in Annex.

#### 10.3.4.3 <Private>

A private tag can be set for a child element. (See Section 10.1.4.)

### 10.3.5 Content Acquisition Sending

Upon receiving an HTTP GET method for content acquisition request in connection mode 1, send the requested content by HTTP1.1 200OK. A SOAP action is not specified.

### 10.3.6 Content Sending End Response (EndSendContentResponse)

This message is returned in response to content sending end (EndSendContent).

For the sequence, see Figure 10-1.

Table 10-11/JJ-40.20: Content sending end response (EndSendContentResponse)

| Variable | | M/O | Details |
|---|---|---|---|
| <ct:EndSendContentResponse> | | M | － |
| | <Private> | O | ・Shall be used for private extension. |

* "M"= Mandatory, "O" = Option

#### 10.3.6.1 <Private>

A private tag can be set for a child element. (See Section 10.1.4.)

### 10.3.7 Session Continue Response (ContinueSessionResponse)

For session continuity (ContinueSession), the server equipment informs the client equipment of a new TCP session timer value that continues the TCP session.

The notification of the TCP session timer value shall be described in the "SessionTime" tag for capability notification response (InformCapabilityResponse) and capability confirmation response (GetCapabilityResponse). (See Sections 10.3.1 and 10.3.2.)

For the sequence, see Figure 10-2.

Table 10-12/JJ-40.20: Session continue response (ContinueSessionResponse)

| Variable | | M/O | Details |
|---|---|---|---|
| <ct:ContinueSessionResponse> | | M | － |
| | <Private> | O | ・Shall be used for private extension. |

* "M"= Mandatory, "O" = Option

#### 10.3.7.1 <Private>

A private tag can be set for a child element. (See Section 10.1.4.)

## 11 Specification of Error Codes

An error generated during the processing of a SOAP message can be posted with SOAP-specified error codes.

The available error codes are listed in Table 11-1.

The details of the errors specified in the SOAP action (Section 10.3.4.1) shall be posted with the SOAP action.

Table 11-1/JJ-40.20: Specification of error codes

| Error code (SOAP specification) *3 | Sub-code (Private specification) | Details | Corresponding SOAP action |
|---|---|---|---|
| VersionMismatch | － | An invalid namespace is found. | All |
| DataEncodingUnknown | － | Encoding of a message cannot be interpreted. | All |

*3: See http://www.w3.org/2003/05/soap-envelope.

# 12 Specification of Quasi-normal and Abnormal State Operations

## 12.1 Specification of Quasi-normal State Operations

Table 12-1/JJ-40.20: Specification of quasi-normal state operations

| Item No. | Status | Countermeasures |
|---|---|---|
| 1 | The protocol version for protocol identification in the tag of "ProtocolName" differs between "capability notification (InformCapability) and capability notification response (InformCapabilityResponse), or between capability confirmation request (GetCapability), and capability confirmation response (GetCapabilityResponse). | ・ Use a lower protocol version.<br>・ Ensure backward compatibility for equipment implementing higher versions. |
| 2 | The equipment name (application name) in the tag of "AgentName" differs between "capability notification (InformCapability) and capability notification response (InformCapabilityResponse), or between capability confirmation request (GetCapability) and capability confirmation response (GetCapabilityResponse). | ・Ignore a privately defined SOAP action, and a variable (tag) defined for a child element of the "Private" tag, and perform the operation specified by this protocol. (See Section 9.2.) |
| 3 | Has no TCP session timer values that should be described in the tag of "SessionTime" for "capability notification response (InformCapabilityResponse) and capability confirmation response (GetCapabilityResponse) " . | ・Operate the TCP session, assuming that the session is not disconnected. |
| 4 | File formats that are supported by content sending equipment cannot be handled by content receiving equipment. | ・Disconnect the TCP session and release the call from outgoing call equipment. |
| 5 | The content of the "Options" and "Private" tags in a SOAP message cannot be understood. | ・Ignore the content.<br>・To reject the content, set an error code to the "JobID" tag for job start response (CreateJobResponse) to post the rejection. (See Section 10.3.4.)<br>・Set a child element of the "Private" tag, so that content sending equipment and content receiving equipment can use it. (See Section 10.1.4.) |

| | | |
|---|---|---|
| 6 | The content receiving equipment designates file formats whose reception is not desired. | ・Set "!" to the tag of "SupportedFormats" for capability notification request (InformCapability) or capability confirmation response (GetCapabilityResponse) to post the reception rejection. (See Sections 10.2.1 and 10.3.2.) |
| 7 | Upon starting content sending, the content receiving equipment wishes to reject the reception for such reasons as the file size being exceeded. | ・Set an error code to the "JobID" tag for job start response (CreateJobResponse) to post the rejection. (See Section 10.3.4.) |
| 8 | During content sending, it is desired to delay the operation because of insufficient memory in the receiving equipment. | ・The receiving equipment does not return TCP's ACK to the sending equipment. |
| 9 | Server equipment receives a request without "Expect: 100-continue" in the header. | ・On the assumption of the reception of the body, return 200 OK after the reception of the body is completed. |
| 10 | Server equipment receives a request with "Expect: 100-continue" in the header and also with the body. | ・After the reception of the body is completed, return 200 OK. |
| 11 | Server equipment receives a request only with "Expect: 100-continue" in the header and returns "HTTP/1.1 100 Continue" , but does not receive the body even after the lapse of a given time. | ・After returning an error response by HTTP status code, release the call. |
| 12 | Client equipment sends "Expect: 100-continue," but does not receive a response from the server equipment. | ・Even if the "HTTP/1.1 100 Continue" response is not returned from the server equipment, start sending the body after the lapse of a given time. |
| 13 | Communication is disconnected without notification of content sending end (EndSendContent). | ・End the content sending, assuming that all of the content has been sent. |
| 14 | F37 Content Sending (HTTP POST) cannot be received because CRLF is inserted before the boundary in front of the content data and after the boundary at the end of the content data, or because CRLF is not inserted. | ・CRFL is treated as an option, as described in RFC2046 5.1.1. Common Syntax. Understand CRLF to receive the content. |
| 15 | When a TCP connection is established, RST packets are returned from the counterpart after sending SYN packets. | ・It is recommended that the same src and dst port numbers be used to resend the SYN packets.<br>・If RST packets are returned even when the SYN packets are resent, release the call. |

## 12.2 Specification of Abnormal State Operations

Table 12-2/JJ-40.20: Specification of abnormal state operations

| Item No. | Status | Countermeasures |
|---|---|---|
| 1 | The "SupportedFormats" tag is set to a capability confirmation request (GetCapability) from the content sending equipment. | ・Ignore the tag.<br>・When a child element of the "Private" tag is used to post the capability, set the element, so that content sending equipment and content receiving equipment can use it.<br>(See Section 10.1.4.) |
| 2 | An upspecified tag is included in a SOAP action which is received by the content receiving side. | ・Ignore the tag.<br>・Return an error response by HTTP status code.<br>・When a tag is set to a child element of the "Private" tag, set the tag, so that content sending equipment and content receiving equipment can use it.<br>(See Section 10.1.4.) |
| 3 | A SOAP request message is sent, but a SOAP response message is not returned. | ・If the response message is not returned even after a lapse of 30 seconds, end the operation, assuming that the other equipment is disconnected for some reason. |
| 4 | An HTTP session is disconnected. | ・Do not perform HTTP reconnection, and release the call, assuming that the sequence has ended.<br>・When performing content sending again, start with an outgoing call. |
| 5 | A TCP session is disconnected during the sending and receiving of content files because of unexpected FIN or RST etc. | ・Do not perform TCP reconnection, and release the call, assuming that the sequence has ended.<br>・When performing content sending again, start with an outgoing call. |

## 13 Examples of Individual Messages

HTTP syntaxes and XML documents in this standard shall be described in accordance with the BN form and in line with the details of references such as referenced RFC.

### 13.1 F1 SIP Connection Request <INVITE＞

```
INVITE ・・・・・・
 (Snip)
v=0
o=- 634249773347155741 634249773347155741 IN IP4 client.kyoto.example.ne.jp
s=-
c=IN IP4 xxx.xxx.xxx.xxx
t=0 0
m=application 9 TCP xxxx
b=AS:2000
a=setup:active
a=connection:new
```

*Point to be noted

・Conform to JT-Q3402 and JF-IETF-RFC4145 for the description of SIP.

・Determine client operation/server operation at "a=setup" . (See Table 9-4.)

### 13.2 F2 SIP Response＜200 OK＞

```
SIP/2.0 200 OK
・・・・・・
(Snip)
v=0
o=- 0 0 IN IP4 client.kyoto.example.ne.jp
s=-
c=IN IP4 xxx.xxx.xxx.xxx
t=0 0
m=application 10000 TCP xxxx
c=IN IP4 xxx.xxx.xxx.xxx
b=AS:2000
a=setup:passive
a=connection:new
```

*Point to be noted

・Conform to JT-Q3402 and JF-IETF-RFC4145 for the description of SIP.

・Determine client operation/server operation at "a=setup" . (See Table 9-4.)

## 13.3 F5 SIP Media Addition Request＜UPDATE＞

```
reINVITE ······
(Snip)
v=0
o=- 634249773347155741 634249773347155741 IN IP4 client.kyoto.example.ne.jp
s=-
c=IN IP4 xxx.xxx.xxx.xxx
t=0 0
m=application 9 TCP xxxx
b=AS:2000
a=setup:active
a=connection:new
```

*Point to be noted

・Conform to JT-Q3402 and JF-IETF-RFC4145 for the description of SIP.

・Determine client operation/server operation at "a=setup" . (See Table 9-4.)

## 13.4 F6 SIP Response＜200 OK＞

```
SIP/2.0 200 OK
······
(Snip)
v=0
o=- 0 0 IN IP4 client.kyoto.example.ne.jp
s=-
c=IN IP4 xxx.xxx.xxx.xxx
t=0 0
m=application 10000 TCP xxxx
c=IN IP4 xxx.xxx.xxx.xxx
b=AS:2000
a=setup:passive
a=connection:new
```

*Point to be noted

・Conform to JT-Q3402 and JF-IETF-RFC4145 for the description of SIP.

・Determine client operation/server operation at "a=setup" . (See Table 9-4.)

## 13.5 F11 Temporary Response Request (Expect: 100-continue)

```
POST /soap_action HTTP/1.1
Content-Length: 318
Content-Type: application/soap+xml; charset="utf-8";
Connection: Keep-Alive
```

```
Expect: 100-continue
Host: xxx.xxx.xxx.xxx:10000
```

### 13.6 F12 Temporary Response (HTTP/1.1 100 Continue)

```
HTTP/1.1 100 Continue
Connection: Keep-Alive
```

### 13.7 F13 Capability Notification (InformCapability)

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ct="http://www.ttc.or.jp/mmsys/ct">
  <env:Body>
    <ct:InformCapability>
      <ct:ProtocolName> ContentsTransfer/1.0</ct:ProtocolName>
      <ct:AgentName>ContentsTransferApplication/1.0,ContentsTransferEquipment/1.0</ct:AgentName>
      <ct:SupportedFormats>image/jpg,image/tiff,!application/*</ct:SupportedFormats>
    </ct:InformCapability>
  </env:Body>
</env:Envelope>
```

### 13.8 F14 Capability Notification Response (InformCapabilityResponse)

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ct="http://www.ttc.or.jp/mmsys/ct">
  <env:Body>
    <ct:InformCapabilityResponse>
      <ct:ProtocolName>ContentsTransfer/1.0</ct:ProtocolName>
      <ct:AgentName>ContentsTransferApplication/1.0,ContentsTransferEquipment/1.0</ct:AgentName>
      <ct:SessionTime>150</ct:SessionTime>
    </ct:InformCapabilityResponse >
  </env:Body>
</env:Envelope>
```

### 13.9 F17 Content List Acquisition Request (GetContentsList)

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ct="http://www.ttc.or.jp/mmsys/ct">
  <env:Body>
```

```
    <ct:GetContentsList >
    </ct:GetContentsList >
  </env:Body>
</env:Envelope>
```

### 13.10 F18 Content List Acquisition Response (GetContentsListResponse)

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ct="http://www.ttc.or.jp/mmsys/ct">
  <env:Body>
    <ct:GetContentsListResponse>
      <ct:ContentsList>
        &lt;?xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?&gt;
        &lt;xmlns:ct=&quot;http://www.ttc.or.jp/mmsys/ct&quot;&gt;
          &lt;ct:Contents&gt;
            &lt;ct:Content&gt;
              &lt;ct:Path&gt;/&lt;/ct:Path&gt;
              &lt;ct:Name&gt;Photo001.jpg&lt;/ct:Name&gt;
              &lt;ct:Size&gt;326863&lt;/ct:Size&gt;
              &lt;ct:Format&gt;image/jpeg&lt;/ct:Format&gt;
              &lt;ct:Title&gt;Travel Photo No1&lt;/ct:Title&gt;
              &lt;ct:Description&gt; picture during a trip 1&lt;/ct:Description&gt;
            &lt;/ct:Content&gt;
            &lt;ct:Content&gt;
              &lt;ct:Path&gt;/photo/&lt;/ct:Path&gt;
              &lt;ct:Name&gt;Photo002.jpg&lt;/ct:Name&gt;
              &lt;ct:Size&gt;180421&lt;/ct:Size&gt;
              &lt;ct:Format&gt;image/jpeg&lt;/ct:Format&gt;
              &lt;ct:Title&gt;Travel Photo No2&lt;/ct:Title&gt;
              &lt;ct:Description&gt; picture during a trip 2&lt;/ct:Description&gt;
            &lt;/ct:Content&gt;
          &lt;/ct:Contents&gt;
      </ct:ContentsList>
    </ct:GetContentsListResponse>
  </env:Body>
</env:Envelope>
```

### 13.11 F21 Content 1 Request <HTTP GET>

```
GET /Photo001.jpg HTTP/1.1
Accept: image/jpeg
```

```
Connection: Keep-Alive
Host: xxx.xxx.xxx.xxx:10000
```

*Point to be noted

<ct:Path> and <ct:Name> for content list acquisition response are used as URL.

## 13.12 F22 Content 1 Sending <HTTP 200 OK>

```
HTTP/1.1 200 OK
Connection: Keep-Alive
Content-Type: image/jpeg
Content-Length: xxx

 (Data)
```

## 13.13 F29 Capability Confirmation Request (GetCapability)

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ct="http://www.ttc.or.jp/mmsys/ct">
  <env:Body>
    <ct:GetCapability>
      <ct:ProtocolName>ContentsTransfer/1.0</ct:ProtocolName>
      <ct:AgentName>ContentsTransferApplication/1.0,ContentsTransferEquipment/1.0</ct:AgentName>
    </ct:GetCapability>
  </env:Body>
</env:Envelope>
```

## 13.14 F30 Capability Confirmation Response (GetCapabilityResponse)

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ct="http://www.ttc.or.jp/mmsys/ct">
  <env:Body>
    <ct:GetCapabilityResponse>
      <ct:ProtocolName>ContentsTransfer/1.0</ct:ProtocolName>
      <ct:AgentName>ContentsTransferApplication/1.0,ContentsTransferEquipment/1.0</ct:AgentName>
      <ct:SessionTime>150</ct:SessionTime>
      <ct:SupportedFormats>image/jpg,image/tiff,!application/*</ct:SupportedFormats>
    </ct:GetCapabilityResponse>
  </env:Body>
</env:Envelope>
```

### 13.15 F33 Job Start Request (CreateJob)

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ct="http://www.ttc.or.jp/mmsys/ct">
  <env:Body>
    <ct:CreateJob>
      <ct:ContentName>Photo002.jpg</ct:ContentName>
      <ct:ContentSize>180421</ct:ContentSize>
      <ct:Format>image/jpeg</ct:Format>
    </ct:CreateJob>
  </env:Body>
</env:Envelope>
```

### 13.16 F34 Job Start Response (CreateJobResponse)

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ct="http://www.ttc.or.jp/mmsys/ct">
  <env:Body>
    <ct:CreateJobResponse>
      <ct:JobID>28</ct:JobID>
      <ct:Path>/photodata</ct:Path>
    </ct:CreateJobResponse>
  </env:Body>
</env:Envelope>
```

### 13.17 F37 Content 1 Sending <HTTP POST>

```
--7da11b1cd70ac2Content-Disposition: form-data; name="xxx"; filename=" Photo002.jpg "Content-Type: image/jpeg
(Data)--7da11b1cd70ac2—
```

### 13.18 F49 Content Sending End (EndSendContent)

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ct="http://www.ttc.or.jp/mmsys/ct">
  <env:Body>
    <ct:EndSendContent>
    </ct:EndSendContent>
  </env:Body>
</env:Envelope>
```

### 13.19 F50 Content Sending End Response (EndSendContentResponse)

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ct="http://www.ttc.or.jp/mmsys/ct">
  <env:Body>
    <ct:EndSendContentResponse >
    </ct:EndSendContentResponse>
  </env:Body>
</env:Envelope>
```

### 13.20 F53 Session Continue Request (ContinueSession)

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ct="http://www.ttc.or.jp/mmsys/ct">
  <env:Body>
    <ct:ContinueSession >
    </ct:ContinueSession>
  </env:Body>
</env:Envelope>
```

### 13.21 F54 Session Continue Response (ContinueSessionResponse)

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ct="http://www.ttc.or.jp/mmsys/ct">
  <env:Body>
    <ct:ContinueSessionResponse >
    </ct:ContinueSessionResponse>
  </env:Body>
</env:Envelope>
```

### 13.22 Error Message

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ct="http://www.ttc.or.jp/mmsys/ct">
  <env:Body>
    <env:Fault>
      <env:Code>
        <env:Value>env: VersionMismatch</env:Value>
      </env:Code>
```

```
     <env:Reason>
        <env:Text xml:lang="en">Version Mismatch</env:Text>
      </env:Reason>
    </env:Fault>
  </env:Body>
</env:Envelope>
```

## 14 Schema of SOAP Message Described in This Main Body

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  targetNamespace="http://www.ttc.or.jp/mmsys/ct" xmlns:ct="http://www.ttc.or.jp/mmsys/ct"
  elementFormDefault="unqualified" attributeFormDefault="unqualified">

  <xs:element name="InformCapability">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="ct:ProtocolName" />
        <xs:element ref="ct:AgentName" />
        <xs:element ref="ct:SupportedFormats" />
        <xs:element name="Private" minOccurs="0" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element type="xs:string" name="ProtocolName" />

  <xs:element type="xs:string" name="AgentName" />

  <xs:element type="xs:string" name="SupportedFormats" />

  <xs:element name="GetCapability">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="ct:ProtocolName" />
        <xs:element ref="ct:AgentName" />
        <xs:element name="Options" minOccurs="0" />
        <xs:element name="Private" minOccurs="0" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="GetContentsList">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Private" minOccurs="0" />
      </xs:sequence>
    </xs:complexType>
```

```
</xs:element>

<xs:element name="CreateJob">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ct:ContentName" />
      <xs:element ref="ct:ContentSize" />
      <xs:element ref="ct:Format" />
      <xs:element name="Options" minOccurs="0">
      </xs:element>
      <xs:element name="Private" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element type="xs:string" name="ContentName" />

<xs:element name="ContentSize">
  <xs:simpleType>
    <xs:restriction base="xs:long">
      <xs:minInclusive value="0" />
      <xs:maxInclusive value="9223372036854775807" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>

<xs:element type="xs:string" name="Format" />

<xs:element name="EndSendContent">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Private" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="ContinueSession">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Private" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
<xs:element name="InformCapabilityResponse">
   <xs:complexType>
      <xs:sequence>
         <xs:element ref="ct:ProtocolName" />
         <xs:element ref="ct:AgentName" />
         <xs:element ref="ct:SessionTime" />
         <xs:element name="Private" minOccurs="0" />
      </xs:sequence>
   </xs:complexType>
</xs:element>

<xs:element type="xs:string" name="SessionTime" />

<xs:element name="GetCapabilityResponse">
   <xs:complexType>
      <xs:sequence>
         <xs:element ref="ct:ProtocolName" />
         <xs:element ref="ct:AgentName" />
         <xs:element ref="ct:SessionTime" />
         <xs:element ref="ct:SupportedFormats" />
         <xs:element name="Options" minOccurs="0" />
         <xs:element name="Private" minOccurs="0" />
      </xs:sequence>
   </xs:complexType>
</xs:element>

<xs:element name="GetContentsListResponse">
   <xs:complexType>
      <xs:sequence>
         <xs:element ref="ct:ContentsList" />
         <xs:element name="Private" minOccurs="0" />
      </xs:sequence>
   </xs:complexType>
</xs:element>

<xs:element name="ContentsList">
   <xs:complexType>
      <xs:sequence maxOccurs="unbounded">
         <xs:element ref="ct:Contents" />
      </xs:sequence>
   </xs:complexType>
</xs:element>
```

```
<xs:element name="Contents">
        <xs:complexType>
            <xs:sequence maxOccurs="unbounded">
                <xs:element ref="ct:Content" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="Content">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="ct:Path" />
                <xs:element ref="ct:Name" />
                <xs:element ref="ct:Size" />
                <xs:element ref="ct:Format" />
                <xs:element ref="ct:Title" minOccurs="0" />
                <xs:element ref="ct:Description" minOccurs="0" />
                <xs:element name="Options" minOccurs="0" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element type="xs:string" name="Path" />

    <xs:element type="xs:string" name="Name" />

    <xs:element name="Size">
        <xs:simpleType>
            <xs:restriction base="xs:long">
                <xs:minInclusive value="0" />
                <xs:maxInclusive value="9223372036854775807" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>

    <xs:element type="xs:string" name="Title" />

    <xs:element type="xs:string" name="Description" />

    <xs:element name="CreateJobResponse">
        <xs:complexType>
            <xs:sequence>
        <xs:element ref="ct:JobID" />
        <xs:element ref="ct:Path" />
        <xs:element name="Options" minOccurs="0" />
        <xs:element name="Private" minOccurs="0" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="JobID">
    <xs:simpleType>
      <xs:restriction base="xs:int">
        <xs:minInclusive value="-4" />
        <xs:maxInclusive value="2147483647" />
      </xs:restriction>
    </xs:simpleType>
  </xs:element>

  <xs:element name="EndSendContentResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Private" minOccurs="0" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="ContinueSessionResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Private" minOccurs="0" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

# 15 Conclusion

This standard describes the content transfer system using SOAP/HTTP in an NGN environment, focusing on the basic specifications by which sending equipment sends content to receiving equipment. Future extension of the standard will be described in Annex.

## Annex

To be added in the future.

# Appendix A  Interconnection with Equipment As Per Technical Report TR-1038 Published Before Establishment of JJ-40.20

The technical report TR-1038 published before the establishment of JJ-40.20 described the protocol types (protocol name and protocol version) and equipment types (equipment name and equipment version) that are included in the tags of "ProtocolName" and "AgentName" for SOAP actions "capability notification (InformCapability), capability notification response (InformCapabilityResponse), capability confirmation request (GetCapability), and capability confirmation response (GetCapabilityResponse) " , in "User-Agent" and "Server" of the HTTP header.

When interconnection with equipment conforming to TR-1038 is foreseen, it is necessary to implement the equipment conforming to this standard JJ-40.20 so that it is possible to understand the protocol and equipment types that are described in "User-Agent" and "Server" of the HTTP header, because the protocol and equipment types are not included in the SOAP action.

When the protocol and terminal types are described in both the SOAP action and HTTP header, priority shall be given to the protocol and terminal types that are included in the SOAP action.

The method for describing "User-Agent" and "Server" of the HTTP header, which is included in technical report TR-1038, is as follows:

■Method for describing User-Agent

Describe a protocol type (protocol name/protocol version) and equipment type (equipment name (application name)/equipment version (application version)) in "User-Agent" of the HTTP header that is requested by client equipment, as a comment in accordance with RFC2616.

Describe multiple equipment types in comma-separated format.

User-Agent:<SP><protocol-name>/<protocol-version><SP>(<equipment-name >;<SP><equipment-version>;)

Table a-1/TR-1038: Setting details of User-Agent

| Item | Setting details |
|---|---|
| <SP> | "%x20" (Single-byte space) |
| <protocol-name> | Set the item according to Table 9-3. |
| <protocol-version> | Set the item according to Table 9-3. |
| <equipment-name> | Set the item using single-byte alphanumeric characters. |
| <equipment-version> | Set the item using single-byte alphanumeric characters. |

■Method for describing Server

Describe a protocol type (protocol name/protocol version), equipment type (equipment name (application name)/equipment version (application version)), and TCP session timer value (see Section 10.3.1.3) in "Server" of the HTTP header that is responded to by server equipment, as a comment in accordance with RFC2616. Describe multiple equipment types in comma-separated format.

Server: <SP><protocol-name>/<protocol-version><SP>(<equipment-name >;<SP><equipment-version>;)<SP>x:<TCP-session-timer-value (second) >;

Table a-2/TR-1038: Setting details of Server

| Item | Setting details |
|---|---|
| <SP> | "%x20" (Single-byte space) |
| <protocol-name> | Set the item according to Table 9-3. |
| <protocol-version> | Set the item according to Table 9-3. |
| <equipment-name> | Set the item using single-byte alphanumeric characters. |
| <equipment-version> | Set the item using single-byte alphanumeric characters. |
| <TCP-session-timer-value (second)> | Set the item using single-byte numbers. |