IPTV RTSP Interface Specification

IPTV RTSP Interface Specification 0.3.doc

Version 0.3

February 12, 2007

Cisco Systems, Inc

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
0.1	Cisco Systems, Inc.	Initial revision	02/02/2007
0.2	Cisco Systems, Inc.	Added use case for a session changing content.	02/05/2007
0.3	Cisco Systems, Inc.	Added some clarifying text to various paragraphs	02/12/2007

Contents

1 Introduction

1.1 Overview

This document defines the Cisco Systems, Inc. IPTV RTSP (Real Time Streaming Protocol) interface supported by the Cisco Content Delivery System (CDS)

This protocol is based on the RFC 2326 “Real Time Streaming Protocol” specifications, with some limited modifications and extensions. This document will provide details on the supported RTSP messages and Cisco specific extensions.

This document will be used to develop the by VOD STB client vendors to implement the client side of the protocol.

Refer to RTSP RFC 2326 for all items not defined in this document.

1.2 Acronyms Defined

Acronym	Definition
CDS	Content Delivery System
DVB	Digital Video Broadcast
RTSP	Real Time Streaming Protocol
SDP	Session Description Protocol
STB	Set top Box
TBD	To Be Defined
VOD	Video On Demand

Table 1

1.3 Related Documents

[1] - RFC 2616 “Hypertext Transfer Protocol -- HTTP/1.1”
[2] - RFC 2326 “Real Time Streaming Protocol (RTSP)”.
[3] - RFC 2327 “SDP: Session Description Protocol”.

2 Communications

2.1 Overview

This section provides a high level overview of the communications between the STB and CDS RTSP server.

2.2 Requests

The client first connects to the CDS RTSP server, and sends a DESCRIBE request specifying the desired content. The DESCRIBE response returns in SDP format the duration of the content. If the client does not require the content duration, this request is not required.

The client sends a SETUP request which includes the URL of the first content to be streamed. The SETUP returns, in the Session header, the session ID for all future requests to the CDS RTSP server.

The PLAY and PAUSE requests are sent to the CDS RTSP server. Typically, only one content URL is requested by the client to be streamed, but the current specification allows more than one content element to be streamed in a single session. In the current instantiation, if the URL in a subsequent PLAY request is different than the current URL, the stream will be immediately transitioned to the new content and not queued.

The TEARDOWN request is sent to the CDS RTSP server.

The GET_PARAMETER is sent to the CDS RTSP server for getting the current stream position and also as a heartbeat.

The ANNOUNCE method can be sent by the CDS RTSP server.

2.3 Connections

The client maintains one connection only to the CDS RTSP server. The host and port number of the CDS RTSP server is specified in the RTSP URL.

The connection is a TCP connection. The connection is persisted for the duration of the VOD session.

2.4 Disconnections and Reconnections

The server will close the connection and terminate the RTSP session if no request is sent to the RTSP server through the connection for a specific time period (5 minutes by default). Note that the client is expected to do a GET_PARAMETER periodically to adjust the current stream position. A different timeout value can be specified by the RTSP server in response to the SETUP request.

If for some reason, the client gets disconnected from the RTSP server, the client will try to reconnect and will issue a GET_PARAMETER for the current session to check if the session is still alive. If not, the RTSP server will return a Session Not Found error (454).

When the client wants to close the session, it first sends a TEARDOWN request to the RTSP server, and after receiving the response, closes the connection.

If the client cannot tune to the stream, the client must issue a TEARDOWN to the RTSP server.

2.5 Server Behavior

When reaching the end of stream (or beginning of stream in rewind), the RTSP server will issue an ANNOUNCE message to the client and enter a stopped state.

3 RTSP Methods

3.1 Methods Supported

The following methods are supported:

- SETUP (Client->RTSP server)
- TEARDOWN (Client->RTSP serve)
- PLAY (Client->RTSP Server)
- PAUSE (Client->RTSP Server)
- GET_PARAMETER (Client->RTSP Server)
- DESCRIBE (Client->RTSP Server)
- ANNOUNCE (RTSP Server->Client)
- OPTIONS (Client->RTSP Server

The following methods are not supported:

- SET_PARAMETER
- REDIRECT

Refer to RTSP RFC 2326 for all items not defined in this document.

3.1.1 Status Codes

The Status-Code returned in the RTSP response must be in the range 100 to 599 (as per RFC). All known status codes are described in this document. If the server returns another value, the client must treat the error according to the range value:

- 1xx: Informational - Request received, continuing process
- 2xx: Success - The action was successfully received, understood, and accepted
- 3xx: Redirection - Further action must be taken in order to complete the request
- 4xx: Client Error - The request contains bad syntax or cannot be fulfilled
- 5xx: Server Error - The server failed to fulfill an apparently valid request

3.2 Method Headers

Client and servers must ignore any unrecognized headers.

3.2.1 Case-Sensitivity

From the RTSP client, all headers are case-sensitive. This means that, in all RTSP requests and responses, the client must use the headers exactly as specified in this document.

The client may treat headers coming from the RTSP server as case-insensitive.

3.2.2 CSeq

From the client to the server, the CSeq header value is a numerical string (up to 9 digits) representing an unsigned number. (1 to 999999999). The value wraps around at 999999999 to 1. The value is incremented for each method sent to the server.

```
CSeq = "CSeq" “:” SP 1*9DIGIT
```

From the server to the client, the CSeq header will be a numerical string of up to 20 digits.

```
CSeq = "CSeq" “:” SP 1*20DIGIT
```

3.2.3 RTSP URI

The RTSP URI will specify the content to be played for the generated session. The URI will contain the RTSP server IP address, port, and the content identifier. Example:

```
rtsp_URI = rtsp://10.20.30.40:554/titanic.mpg
```

The RTSP URI syntax is as follows:

```
rtsp_URI = ("rtsp" ":" IP “:” PORT “/” CONTENT) | "*"
IP = *<TEXT, excluding SP, CR, LF>
PORT = 1*5DIGIT
CONTENT = 1*254<TEXT, excluding “/”, SP, CR, LF>
```

3.2.4 RTSP Extensions

The following RTSP headers are defined as syntax extensions:

- x-mayNotify

  This is part of the SETUP request that indicates the RTSP server may send asynchronous ANNOUNCE messages to the client.
- x-playNow

  This is part of the PLAY request to indicate to the RTSP server to perform this play operation now, do not queue the request.
- x-notice

  This is part of the ANNOUNCE request from the RTSP server to the client. It contains the reason for the announce message.

3.3 SETUP

The SETUP method is used to initiates a RTSP session for a specified URI, targeting a specified destination. The response from the server contains the session ID for the RTSP session. The response also contains some parameters needed for tuning to the proper stream.

Limitations of the RTSP implementation:

- o Case-sensitive headers,
- o The SETUP of an already playing session is not supported,
- o The Session value is a numeric string, up to 20 characters long:

Session = "Session" “:” SP 1*20DIGIT [";" "timeout" "=" delta-seconds]

3.3.1 Client Request

The SETUP request contains the following headers:

```
    CSeq: number,
    Transport:
    MP2T/H2221/UDP;unicast;destination=<ip>;client_port=<port>
    x-mayNotify:
```

A single Transport format is supported. The Transport destination attribute contains the IP address for the stream target. The Transport client_port attribute contains the UDP port for the stream target.

The x-mayNotify extension header is used to tell the RTSP server that it is allowed to send ANNOUNCE requests to the client.

Format

```
Setup-Request = "SETUP" SP rtsp_URL SP RTSP-Version CRLF
                CSeq CRLF
                Transport CRLF
                x-mayNotify CRLF
                CRLF
```

CSeq = "CSeq" “:” SP 1*9DIGIT ; 1 to 999999999

Transport = "Transport" “:” SP transport-spec

transport-spec = transport-protocol/profile/lower-transport parameters

transport-protocol = "MP2T"

profile = "H2221"

lower-transport = "UDP"

parameters = ";" "unicast" ";" "destination" "=" destination “;” “client_port” “=” client_port destination = 1*15TEXT ; dotted-decimal format

client_port = 1*5DIGIT ; 1 to 65535

x-mayNotify = "x-mayNotify:"

3.3.2 Server Response

The server response may return a success status code (200).

The server response, in case of success, includes the following headers:

```
CSeq: number,
Session: number [; timeout=value] (timeout is optional),
Range: value,
Transport:
MPT2/H2221/UDP;unicast;client=value;control_address=value;destina
tion=value;bandwidth=value
```

The Session represents the session ID for all requests to the RTSP server.

The Session value can include a timeout value (as per RFC). The default is 5 minutes. The client must send to the RTSP server a heartbeat (or equivalent) at a rate around 80% of the timeout value.

The range is the NPT of the stream. In the SETUP response the start time will be 0.00 and the end time will be EOS.

The client is a dotted-decimal IP address of the client STB.

The control_address is a dotted-decimal IP address of the controlling device. In the case of IPTV, this value is the IP of the client STB.

The destination is a dotted-decimal IP address and UDP port of the destination for the stream.

The bandwidth is a numeric value representing the bitrate of the selected content in bits-per-second.

Format

```
Setup-Response = RTSP-Version SP Status-Code SP Reason-Phrase CRLF
                 CSeq CRLF
                 Session CRLF
                 Range CRLF
                 Transport CRLF
                 CRLF
```

Reason-Phrase = *<TEXT, excluding CR, LF>

CSeq = "CSeq" “:” SP 1*9DIGIT ; 1 to 999999999

Session = "Session" “:” SP 1*20(DIGIT) [";" "timeout" "=" delta-seconds]

delta-seconds = 1*5DIGIT

Range = "Range" “:” SP "npt= " npt-range

npt-range = npt-time "-"

npt-time = "now" | npt-sec

npt-sec = 1*DIGIT ["." 1*3DIGIT]

Transport = "Transport" “:” SP transport-spec

transport-spec = transport-protocol/profile/lower-transport parameters

transport-protocol = "MP2T"

profile = "H2221"

lower-transport = "UDP"

parameters = “;” “unicast” “;” “client” “=” client “;” “control_address” “=” control_address “;” “destination” “=” destination “;” “bandwidth” “=” bandwidth

client = 1*15CHAR ; dotted-decimal format

control_address = 1*15CHAR ; dotted-decimal format

destination = destination_ip “:” destination_port

destination_ip = 1*15CHAR ; dotted-decimal format

destination_port = 1*5DIGIT ; 1 to 65535

bandwidth = 1*8DIGIT ; 1 to 25000000

Error Cases

For the following errors, the server must return the following:
Invalid ‘Content_ID’: 404 Not Found,

Other error codes, as defined in RFC, may also be returned to denote other kinds of failure.

In case of error, which header is returned, and their value is not specified.

3.4 TEARDOWN

The TEARDOWN request is issued to delete the RTSP session and release allocated resources.

3.4.1 Client Request

The TEARDOWN request contains the following headers:
CSeq: number,
Session: number

Format

```
Teardown-Request = "TEARDOWN" SP RTSP-URI SP RTSP-Version CRLF
                   CSeq CRLF
                   Session CRLF
                   CRLF
```

CSeq = "CSeq" “:” SP 1*9DIGIT ; 1 to 999999999

Session = "Session" “:” SP 1*20DIGIT

3.4.2 Server Response

The server response includes the following headers:
CSeq: number,
Session: number.

Format

```
Teardown-Response = RTSP-Version SP Status-Code SP Reason-Phrase CRLF
                    CSeq CRLF
                    Session CRLF
                    CRLF
```

Reason-Phrase = *<TEXT, excluding CR, LF>

CSeq = "CSeq" “:” SP 1*9DIGIT ; 1 to 999999999

Session = “Session” “:” SP 1*20DIGIT

3.5 PLAY

The PLAY method is used to start playing the content. The Play method is also used for fast-forward and rewind.

Limitations of the RTSP implementation:

- Case-sensitive headers,

- only NPT time (Normal Play Time) (no SMPTE time-code),
- no 'time' parameter (play starts immediately),
- no Speed header,
- Scale has a limited number of digits, only whole numbers supported
- Range header is only used for seeking to a specific location (not to play back to back),
- Range only specifies the starting point (ending point is always end of content),
- If Range is not specified, then current position is assumed,
- Play commands are not queued (by the server). Therefore the x-playNow header is mandatory.

3.5.1 Client Request

The PLAY request contains the following headers:

CSeq: number,
Session: number,
Range: value,
Scale: value,
x-playNow:.

The Range header is optional. Only the NPT format is supported. Only starting point is specified: npt-range = npt-time "-". For trick modes (fast-forward and rewind), the Range is not specified (always use current position).

The Scale header is optional. The value "1" means normal play. The Scale value is a floating point number (positive or negative). If the Scale header is not specified, a value of 1 is assumed.

The CDS can support trickmode speeds from 4x to 127x.

If the server cannot do the exact scaling specified, it will use the best-fit scale it can do, and return that value to the client. The best-fit scale is defined as the closest value to the specified scale.

The scales that are enabled for each content URL are known by the CDS. For some categories, such as data-only content (no video or audio) no trick mode operations are defined. In those cases the only scale allowed is 1x.

A scale of 0 is invalid and the server will return 400 Bad Request.

Format

Play-Request = "PLAY" SP RTSP-URI SP RTSP-Version CRLF

```
          CSeq CRLF
          Session CRLF
          [Range CRLF]
          [Scale CRLF]
          x-playNow CRLF
          CRLF
```

CSeq = "CSeq" “:” SP 1*9DIGIT ; 1 to 999999999

Session = "Session" “:” SP 1*20DIGIT

Range = "Range" “:” SP "npt= " npt-range

Scale = "Scale" “:” SP ["-"] 1*3DIGIT ["." *1DIGIT]

x-playNow = "x-playNow" “:”

npt-range = npt-time "-"

npt-time = "now" | npt-sec

npt-sec = 1*DIGIT ["." 1*3DIGIT]

3.5.2 Server Response

The server response includes the following headers:

CSeq: number,
Session: number,
Range: (NPT) value,
Scale: value.

The Scale value returned is the actual value by the RTSP server.

Format

```
Play-Response = RTSP-Version SP Status-Code SP Reason-Phrase CRLF
                    CSeq CRLF
                    Session CRLF
                    Range CRLF
                    Scale CRLF
                    CRLF
```

Reason-Phrase = *<TEXT, excluding CR, LF>

CSeq = "CSeq" “:” SP 1*9DIGIT ; 1 to 999999999

Session = "Session" “:” SP 1*20DIGIT

Range = "Range" “:” SP "npt= " npt-range

```
Scale = "Scale" “:” SP [ "-" ] 1*3DIGIT [ "." *3DIGIT ]

npt-range = npt-time "-" [ npt-time ]

npt-time = now | npt-sec

npt-sec = 1*DIGIT [ "." 1*3DIGIT ]
```

The server may also return the end time of the session stream (second npt-time in npt-range).

3.6 PAUSE

The PAUSE method is used to immediately pause the session stream.

Limitations of the RTSP implementation:

- o Case-sensitive headers,
- o The server returns the current position in a Range header.

3.6.1 Client Request

The PAUSE request contains the following headers:

CSeq: number,
Session: number,
Range: value.

The Range header is optional. Only the NPT format is supported. Only starting point is specified: npt-range = npt-time "-".

Format

```
Pause-Request = "PAUSE" SP RTSP-URI SP RTSP-Version CRLF
                CSeq CRLF
                Session CRLF
                [Range]
                CRLF

CSeq = "CSeq" “:” SP 1*9DIGIT ; 1 to 999999999

Session = "Session" “:” SP 1*20DIGIT

Range = "Range" “:” SP "npt= " npt-range

npt-range = npt-time "-"

npt-time = "now" | npt-sec

npt-sec = 1*DIGIT [ "." 1*3DIGIT]
```

3.6.2 Server Response

The server response includes the following headers:

CSeq: number,
Session: number,
Range: (NPT) value.

The server returns the current position (NPT format) in a Range header.

Format

```
Pause-Response = RTSP-Version SP Status-Code SP Reason-Phrase CRLF
                 CSeq CRLF
                 Session CRLF
                 Range CRLF
                 CRLF
```

```
Reason-Phrase = *<TEXT, excluding CR, LF>
CSeq = "CSeq" “:” SP 1*9DIGIT ; 1 to 999999999

Session = "Session" “:” SP 1*20DIGIT

Range = "Range" “:” SP "npt= " npt-range

npt-range = npt-time "-"

npt-time = npt-sec

npt-sec = 1*DIGIT [ "." 1*3DIGIT ]
```

The first npt-time in the npt-range is the current position.

3.7 GET_PARAMETER

This method is used by the client to query the position of the current session stream.

The client must not overload the RTSP server with too many position queries. The default query interval should be 1 minute.

3.7.1 Client Request

The GET_PARAMETER request contains the following headers:

CSeq: number,
Session: number,
Content-Type: text/parameters,
Content-Length: length.

The body of the request contains the following parameters (one per line, order is not relevant):

Position

If the request is being used as a heartbeat the content length header should be set to 0, and the body should be empty.

Format

```
Get-Parameter-Request = "GET_PARAMETER" SP RTSP-URI SP RTSP-Version
CRLF
                        CSeq CRLF
                        Session CRLF
                        Content-TypeCRLF
                        Content-Length CRLF
                        CRLF
                        position
```

CSeq = "CSeq" “:” SP 1*9DIGIT ; 1 to 999999999

Session = "Session" “:” SP 1*20DIGIT

Content-Type = "Content-Type" “:” SP media-type

media-type = "text/parameters"

Content-Length = "Content-Length" “:” SP 1*9DIGIT

position = "position"

Content-Length is the number of bytes in the message payload. The payload in this case, is “position” and the content length is 8.

3.7.2 Server Response

The server response includes the following headers:

CSeq: number,
Session: number,
Content-Type: text/parameters,
Content-Length: length.

The response body contains one line per parameter requested:

position: NPT value

If this is a response to a heartbeat request, the content length header will be set to 0 and the message body will be empty.

Format

```
Get-Parameter-Response = RTSP-Version SP Status-Code SP Reason-Phrase CRLF
                         CSeq CRLF
                         Session CRLF
                         Content-Type CRLF
                         Content-Length CRLF
                         CRLF
                         Position
```

Reason-Phrase = *<TEXT, excluding CR, LF>

CSeq = "CSeq" ":" SP 1*9DIGIT ; 1 to 999999999

Session = "Session" ":" SP 1*20DIGIT

Content-Type = "Content-Type" ":" SP media-type

media-type = "text/parameters"

Content-Length = "Content-Length" ":" SP 1*9DIGIT

position = "position" ":" SP "npt= " npt-range

npt-range = npt-time "-"

npt-time = npt-sec

npt-sec = 1*DIGIT ["." 1*3DIGIT]

3.8 DESCRIBE

The DESCRIBE method is used by the client to get some information for a content (specified by the RTSP-URI). The client does not need to call this method if it knows the information from some other mean.

The client is expected to use the DESCRIBE method to get an accurate value of the content length.

3.8.1 Client Request

The DESCRIBE request contains the following headers:
    CSeq: number,
    Accept: application/sdp.

Note that there is not a Session header here.

Format

```
Describe-Request = "DESCRIBE" SP RTSP-URI SP RTSP-Version CRLF
                              CSeq CRLF
                              Accept CRLF
                              CRLF
```

CSeq = "CSeq" “:” SP 1*9DIGIT ; 1 to 999999999

Accept = "Accept" “:” SP media-type

media-type = “application/sdp”

3.8.2 Server Response

The server response includes the following headers (case-sensitive):

CSeq: number,
Session: number,
Content-Type: application/sdp,
Content-Length: length.

The body of the response contains information about the selected presentation. The format of the body is defined by the SDP (Session Description Protocol, RFC 2327).

The following parameters are supported:

a=range:npt=begintime-endtime (number of seconds, NPT format)

The length of the VOD presentation will be returned in the endtime field of the “a=range:npt=” parameter. The begintime will be zero (eg, 0.0).

Other parameters are optional, and can be ignored by the client.

Format

```
Describe-Response = RTSP-Version SP Status-Code SP Reason-Phrase CRLF
                              CSeq CRLF
                              Session CRLF
                              Content-Type CRLF
                              Content-Length CRLF
                              CRLF
                              Content-Body
```

Reason-Phrase = *<TEXT, excluding CR, LF>

CSeq = "CSeq" “:” SP 1*9DIGIT ; 1 to 999999999

Session = "Session" “:” SP 1*20DIGIT

Content-Type = "Content-Type" “:” SP media-type

media-type = "application/sdp"

Content-Length = "Content-Length" “:” SP 1*9DIGIT

Content-Body = *(parameter CRLF)

parameter = type “=” value

type = ALPHA

value = *<TEXT, excluding CR, LF>

The following parameters are understood by the server:
“a=range:npt=” npt-time “-“ npt-time

Note that there is no space between “range:” and “npt=”.

3.9 ANNOUNCE

This is a method sent by the RTSP servers. It is used to signal events to the client.

3.9.1 Server Request

The ANNOUNCE requests are sent asynchronously.

The ANNOUNCE request contains the following headers:
CSeq: number,
Session: number,
x-notice: string.

The x-notice header includes an Event-Code, an Event-Phrase and an Event-Date:
x-notice = "x-notice" “:” SP notify
notify = Event-Code SP Event-Phrase SP “Event-Date=” utc-time
Event-Code = 4DIGIT
Event-Phrase = *<TEXT, excluding CR, LF>
utc-time is described in the RTSP RFC

The known Event-Code are:

- 2101 End-of-Stream Reached
- 2104 Start-of-Stream Reached: can happen in case of rewind
- 4400 Error Reading Content Data: the play out has stopped. The client must issue a Teardown of the session.

- o 5402 Client Session Terminated: teardown has been initiated by the server, the session is closed.

The ANNOUNCE request uses '*' to denote current RTSP-URI.

Format

```
Announce-Request = "ANNOUNCE" SP "*" SP RTSP-Version CRLF
                   CSeq CRLF
                   Session CRLF
                   x-notice CRLF
                   CRLF
```

CSeq = "CSeq" ":" SP 1*20DIGIT

Session = "Session" ":" SP 1*20DIGIT

x-notice = "x-notice" ":" SP notify

notify = Event-Code SP Event-Phrase SP "Event-Date=" utc-time

Event-Code = 1*4DIGIT

Event-Phrase = *<TEXT, excluding CR, LF>

3.9.2 Client Response

The client response includes the following headers:
 CSeq: number,
 Session: number.

Format

```
Announce-Response = RTSP-Version SP Status-Code SP Reason-Phrase CRLF
                    CSeq CRLF
                    Session CRLF
                    CRLF
```

CSeq = "CSeq" ":" SP 1*20DIGIT

Session = "Session" ":" SP 1*20DIGIT

Status-Code = "200"

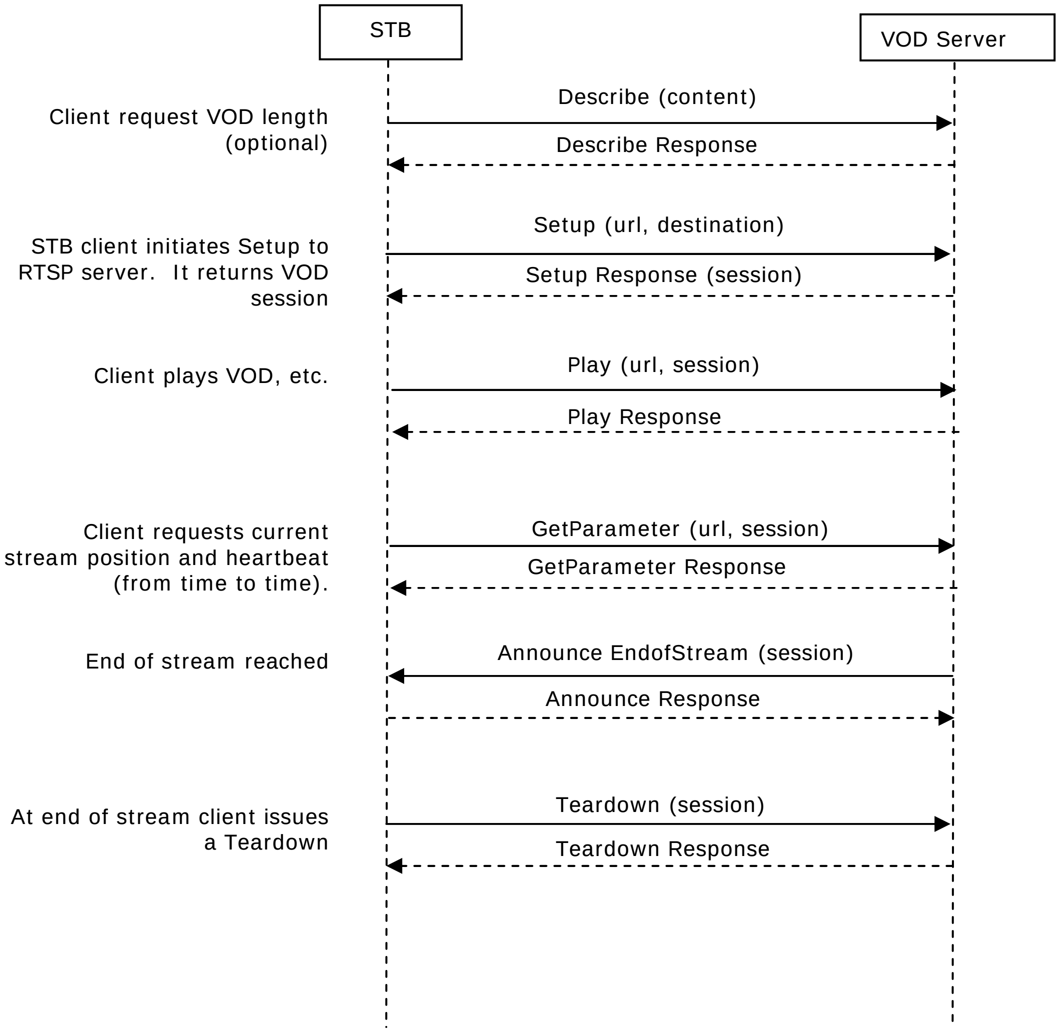
Reason-Phrase = "OK"

4 Usage

4.1 Normal Play-out

The following describes a high-level view of the requests involved in a normal play-out of a full length presentation. Note that the DESCRIBE request is optional.

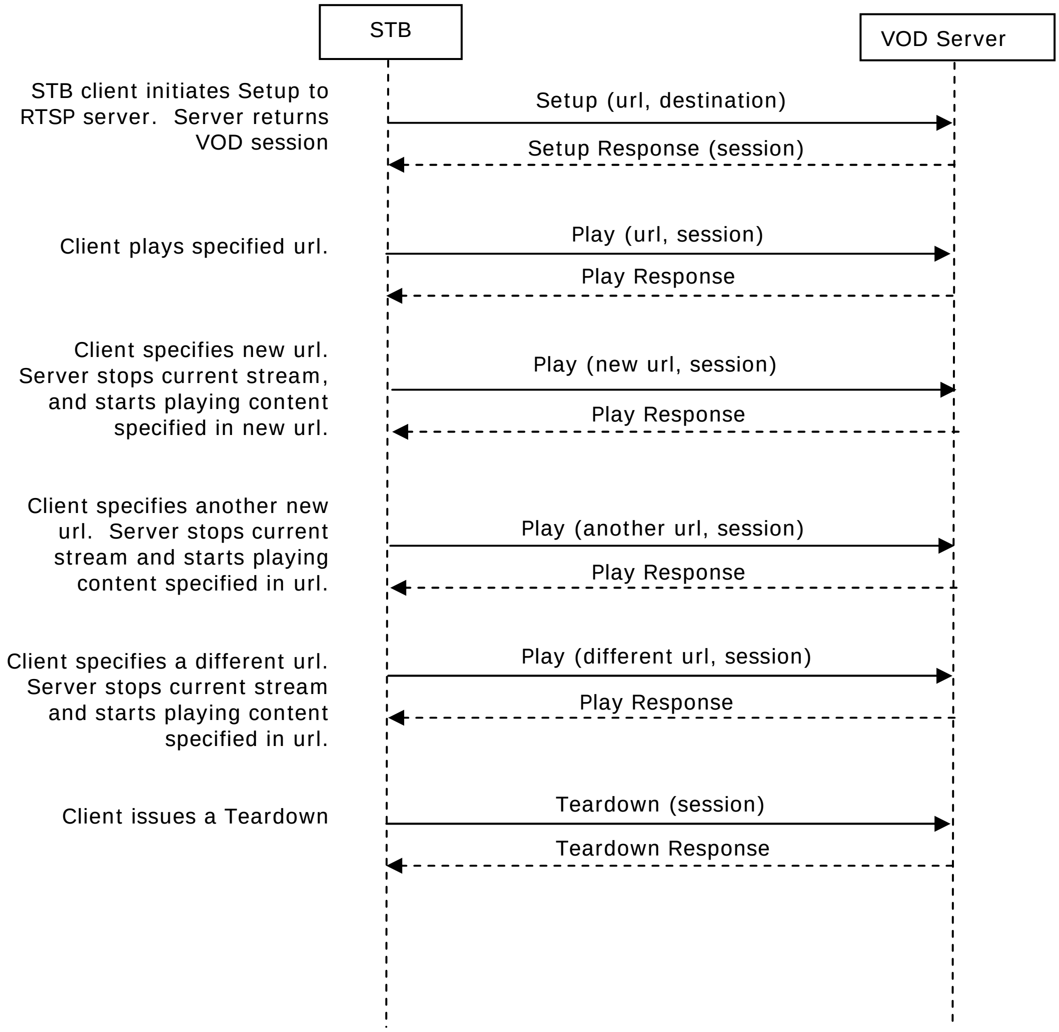
4.1.1 Sequence Diagram


STB
VOD Server
Client request VOD length (optional)
Describe (content)
Describe Response
STB client initiates Setup to RTSP server. It returns VOD session
Setup (url, destination)
Setup Response (session)
Client plays VOD, etc.
Play (url, session)
Play Response
Client requests current stream position and heartbeat (from time to time).
GetParameter (url, session)
GetParameter Response
End of stream reached
Announce EndofStream (session)
Announce Response
At end of stream client issues a Teardown
Teardown (session)
Teardown Response

4.2 Content Swap

The IPTV RTSP server allows the STB client to change the content being streamed for an existing session by issuing a PLAY request with a different content specified in the RTSP-URI field. It is assumed the bit-rate for all content is the same.

4.2.1 Sequence Diagram


STB
VOD Server
STB client initiates Setup to RTSP server. Server returns VOD session
Setup (url, destination)
Setup Response (session)
Client plays specified url.
Play (url, session)
Play Response
Client specifies new url. Server stops current stream, and starts playing content specified in new url.
Play (new url, session)
Play Response
Client specifies another new url. Server stops current stream and starts playing content specified in url.
Play (another url, session)
Play Response
Client specifies a different url. Server stops current stream and starts playing content specified in url.
Play (different url, session)
Play Response
Client issues a Teardown
Teardown (session)
Teardown Response