

RTSP Negotiation for Reliable RTP and Instant-On

Intimation of available features

Reliable RTP

If the QuickTime Streaming Server is capable of supporting Reliable RTP (versions 3.0 and higher), it will return an "X-Accept-Retransmit" header with the value "our-retransmit" in the DESCRIBE response to the client.

```
x-Accept-Retransmit: our-retransmit
```

Dynamic Streaming Rate / Instant-On

The QuickTime Streaming Server is capable of streaming at a rate different than the encoded rate of the movie. If the version of the server supports the dynamic rate, the server will return an "X-Accept-Dynamic-Rate" header with a value of "1" in the DESCRIBE response to the client. The dynamic rate feature of the server allows the client to start instant playback of the stream. Instant-On feature of the client will only work if the server sends this header in the DESCRIBE response.

```
x-Accept-Dynamic-Rate: 1
```

Negotiation

SETUP Request

If the client supports Reliable RTP and can send RTCP ACK packets to the server while receiving the RTP stream from the server, it will acknowledge to the server by sending the "x-retransmit" header with the value "our-retransmit" in each SETUP request. It will not send this header if it doesn't support Reliable RTP, which will cause the server to not use Reliable RTP as the transport, but simply use regular RTP.

```
x-retransmit: our-retransmit
```

If the client can handle a dynamic streaming rate from the server, instead of the rate at which the stream is encoded, the client will echo back the "x-dynamic-rate" header with a value of "1". If the client wants the stream at the rate at which it was encoded, the client will send back the "x-dynamic-rate" header with a value of "0". If the client doesn't send the "x-dynamic-rate" header it defaults to "1" for Reliable RTP, and to "0" otherwise (HTTP tunneled and RTP).

```
x-dynamic-rate: 1
```

The client also sends a "x-transport-options" header with a "late-tolerance" value. The late-tolerance value is in seconds and is used by the server to determine how late the packet can be before it is dropped. The client initially sends a pre-determined value (based on the how instant the stream should be played back and the available bandwidth preference in the client) to the server in the SETUP request.

```
x-transport-options: late-tolerance=0.150000
```

The client recalculates the late-tolerance after it has received some RTP data from the stream and resends the header with a new value in a SET_PARAMETER request.

OPTIONS request

When the server receives the SETUP request from the client and if client supports dynamic rate (by sending the x-dynamic-rate header), the server first sends an OPTIONS request to the client with some random data to calculate the

round-trip time between the server and the client. It sends this information to the client as the "rtt" parameter of the "x-dynamic-rate" header it sends to the client in the SETUP response.

```
x-Dynamic-Rate: 1;rtt=94
```

The OPTIONS request is only sent for the first SETUP request. The remaining SETUP requests are handled as if the x-dynamic-rate header weren't sent, by sending a SETUP response immediately.

PLAY request

The client sends a very high late-tolerance value in the PLAY request if it decides to support Instant-On. This enables the server to send data without dropping any intermediate packets in the case of a bad network connection. The client waits until it receives the expected amount of data and then recalculates the late-tolerance value.

```
x-transport-options: late-tolerance=10
```

SET_PARAMETER request

The client sends the new late-tolerance value soon after it receives the first few RTP packets.

```
x-transport-options: late-tolerance=1.284000
```

When the stream is PAUSED and the client issues a new PLAY request, the late-tolerance values are recalculated again. A new SET_PARAMETER request is sent by the client to reset the late-tolerance for the stream in the server.

S->C: RTSP/1.0 200 OK
Server: QTSS/4.1.2 (Build/412.42; Platform/MacOSX)
Cseq: 2
Last-Modified: Sat, 27 Jul 2002 10:01:21 GMT
Cache-Control: must-revalidate
Session: 7873850505607127735
Date: Mon, 18 Nov 2002 03:21:35 GMT
Expires: Mon, 18 Nov 2002 03:21:35 GMT
Transport: RTP/AVP;unicast;client_port=6974-6975;source=127.0.0.1;server_port=6970-6971;ssrc=5CCC1742
x-Transport-Options: late-tolerance=0.150000
x-Retransmit: our-retransmit
x-Dynamic-Rate: 1;rtt=94

C->S: SETUP rtsp://127.0.0.1/sample_300kbit.mp4/trackID=4 RTSP/1.0
CSeq: 3
Transport: RTP/AVP;unicast;client_port=6972-6973
x-retransmit: our-retransmit
x-dynamic-rate: 1
x-transport-options: late-tolerance=0.150000
Session: 7873850505607127735
User-Agent: QTS (qtver=6.0.2;cpu=PPC;os=Mac 10.2.2)
Accept-Language: en-US

S->C: RTSP/1.0 200 OK
Server: QTSS/4.1.2 (Build/412.42; Platform/MacOSX)
Cseq: 3
Session: 7873850505607127735
Last-Modified: Sat, 27 Jul 2002 10:01:21 GMT
Cache-Control: must-revalidate
Date: Mon, 18 Nov 2002 03:21:35 GMT
Expires: Mon, 18 Nov 2002 03:21:35 GMT
Transport: RTP/AVP;unicast;client_port=6972-6973;source=127.0.0.1;server_port=6970-6971;ssrc=7F9F1189
x-Transport-Options: late-tolerance=0.150000
x-Retransmit: our-retransmit
x-Dynamic-Rate: 1

C->S: PLAY rtsp://127.0.0.1/sample_300kbit.mp4 RTSP/1.0
CSeq: 4
Range: npt=0.000000-66.470000
x-prebuffer: maxtime=2.000000
x-transport-options: late-tolerance=10
Session: 7873850505607127735

User-Agent: QTS (qtver=6.0.2;cpu=PPC;os=Mac 10.2.2)

S->C: RTSP/1.0 200 OK

Server: QTSS/4.1.2 (Build/412.42; Platform/MacOSX)

Cseq: 4

Session: 7873850505607127735

RTP-Info:

url=trackID=3;seq=47653;rtptime=487090573,url=trackID=4;seq=13140;rtptime=1699959952

C->S: SET_PARAMETER rtsp://127.0.0.1/sample_300kbit.mp4 RTSP/1.0

CSeq: 5

x-transport-options: late-tolerance=1.284000

Session: 7873850505607127735

User-Agent: QTS (qtver=6.0.2;cpu=PPC;os=Mac 10.2.2)

S->C: RTSP/1.0 200 OK

Server: QTSS/4.1.2 (Build/412.42; Platform/MacOSX)

Cseq: 5

Session: 7873850505607127735

Appendix B: QTSS Ports

Usage	Ports	Protocols	Notes
Responding to messages from clients (such as Play and Pause)	TCP (client initiates -> QTSS) 554,7070, 8000, 8001, 80)	RTSP, RTP, RTCP, MP3	Main port is 554. 80 is supported in the QT client as an alternative TCP port. These ports also send data to clients.
Sending media and receiving client status	<ul style="list-style-type: none"> • UDP data (QTSS -> client): 6970–6999, even numbers • UDP status (QTSS <-> client): 6971–6999, odd numbers • TCP data & status (QTSS <-> client): 554, 7070, 8000, 8001, 80 	RTP RTCP RTSP, RTP,RTCP	Status is required to maintain a connection; if blocked, the server disconnects the client. Same ports used to respond to messages.
Receiving broadcasts	<ul style="list-style-type: none"> • UDP data (broadcaster -> QTSS): 6972–65535, even numbers • UDP RTCP status (broadcaster <-> QTSS): 6973–65535, odd numbers • TCP (broadcaster initiates -> QTSS): 554, 7070, 8000, 8001, 80 	RTP RTCP RTSP, RTP, RTCP	<ul style="list-style-type: none"> • Ports depend on the broadcaster configuration. • Status is required to maintain a connection; if blocked, the server disconnects the broadcaster. • Broadcasters can broadcast over their TCP message connection with the server instead of using UDP ports.
Streaming through server	TCP (client initiates -> QTSS): 554, 7070, 8000, 8001, 80	RTSP, RTP, RTCP, MP3	Same ports used to respond to messages and receive TCP broadcasts.
MP3 broadcasts (typical default)	TCP (client -> QTSS): 8000	HTTP (Icecast)	
Managing QTSS remotely with Server Admin	TCP (admin client initiates -> server): 311		
Managing QTSSremotely with Web Admin	TCP (web browser client intiates -> server): 1220	HTTP	

Appendix C: Common Streaming Error Codes (Client)

This is a list of common streaming error codes. Applications that play streaming movies should deal gracefully with these errors.

Streaming Errors

- 5400 - qtsBadSelectorErr
- 5401 - qtsBadStateErr
- 5402 - qtsBadDataErr
- 5403 - qtsUnsupportedDataTypeErr
- 5404 - qtsUnsupportedRateErr
- 5405 - qtsUnsupportedFeatureErr
- 5406 - qtsTooMuchDataErr

Network Errors

- 5420 - connection failed (couldn't connect to the server)

Open Transport (Mac only: -3150 to -3180, -3200 to -3285)

look for the full range of OT error codes in OpenTransport.h

- 3150 - kOTBadAddressErr - a bad address was specified
- 3170 - kOTBadNameErr; couldn't do name lookup

Windows only

- 10061 - couldn't connect to server

Mac Toolbox Errors

Streaming can return all QuickTime and Mac Toolbox errors. Common error codes returned include:

QuickTime Errors

- 223 - siInvalidCompression; codec not installed

Windows Specific Errors

- 2092 - componentDllEntryNotFoundErr; Windows specific error when component is loading

RTSP Errors

Most of these error codes are the same as for HTTP. These errors are usually generated from the server.

- 3xx - redirect
- 400 - bad request
- 401 - unauthorized
- 402 - payment required
- 403 - forbidden

404 - not found (specified movie not found on server side)
405 - method not allowed
406 - not acceptable
407 - proxy authentication required
408 - request time-out
409 - conflict
410 - gone
411 - length required
412 - precondition failed
413 - request entity too large
414 - request URI too large
415 - unsupported media type
451 - parameter not understood
452 - conference not found
453 - not enough bandwidth
454 - session not founds
455 - method not valid in this state
456 - header field not valid for resource
457 - invalid range
458 - parameter is read only
459 - aggregate operation not allowed
460 - only aggregate operation allowed
461 - unsupported transport
462 - destination unreachable
500 - internal server error
501 - not implemented
502 - bad gateway
503 - service unavailable
504 - gateway timeout
505 - version not supported
551 - option not supported