Tunneling RTSP/RTP/RTCP in HTTP

(draft-gentric-avt-rtsp-http-00.txt)

Anne Jones, Apple
Philippe Gentric, Philips MP4Net

RTSP (RFC 2326 section 10.12) describes a TCP-based streaming mode (RTSP session control is interleaved with RTP data packets and RTCP control packets)

C->S: SETUP rtsp://foo.com/bar.file RTSP/1.0

CSeq: 2

Transport: RTP/AVP/TCP;interleaved=0-1

S->C: RTSP/1.0 200 OK

CSeq: 2

Date: 05 Jun 1997 18:57:18 GMT

Transport: RTP/AVP/TCP;interleaved=0-1

Session: 12345678

C->S: **PLAY** rtsp://foo.com/bar.file RTSP/1.0

CSeq: 3

Session: 12345678

S->C: RTSP/1.0 200 OK

CSeq: 3

Session: 12345678

Date: 05 Jun 1997 18:59:15 GMT

RTP-Info: url=rtsp://foo.com/bar.file;

seq=232433;rtptime=972948234

S->C: $\000{2 byte length}{"length" bytes data, w/RTP header}

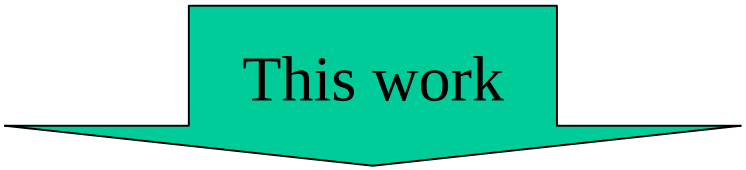
S->C: $\000{2 byte length}{"length" bytes data, w/RTP header}

S->C: $\001{2 byte length}{"length" bytes RTCP packet}

RTSP
RFC 2326
Section 10.12

Interleaved
RTP/RTCP packets
Inside RTSP/TCP

Works fine, except for HTTP-only firewalls



Extend it for HTTP-only firewalls

+ Pragmatic approach
+ traverse ANY firewall NOW, including
  - HTTP 1.0
  - « Poor » implementations

Motivation/Context

ISMA: (Internet Streaming Media Alliance)

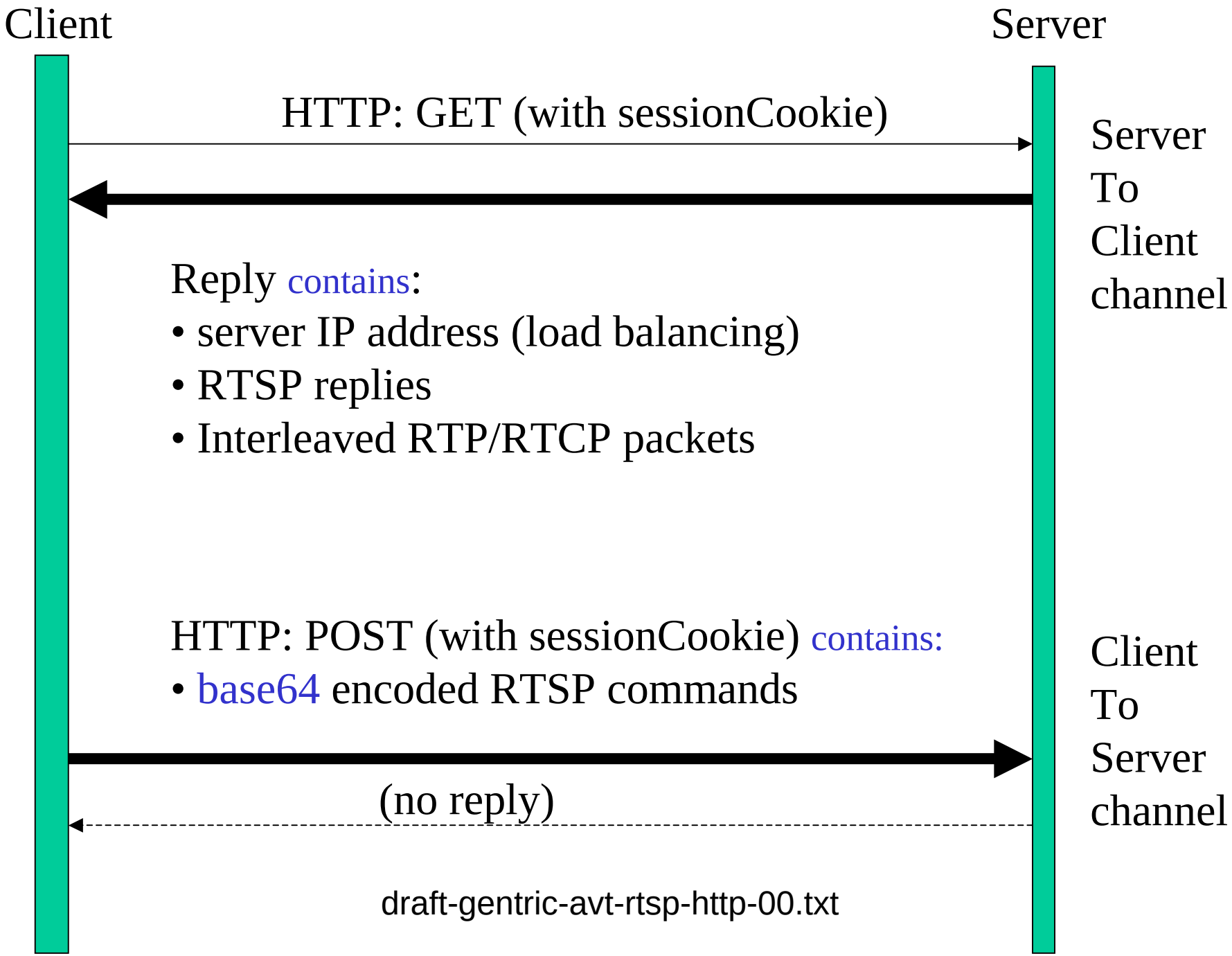
→Media distribution

→Standard based

→Basic product requirement

Solution outline:

- Use HTTP to create 2 channels
    - One Client-to-Server
    - One Server-to-Client
- Use Cookie to correlate both
- Server sends its IP address to enable load balancing schemes
- « Hide » (in Base 64) the client-to-server RTSP syntax to prevent proxies from blocking on it


Client
Server
HTTP: GET (with sessionCookie)
Server
To
Client
channel
Reply contains:
• server IP address (load balancing)
• RTSP replies
• Interleaved RTP/RTCP packets
HTTP: POST (with sessionCookie) contains:
• base64 encoded RTSP commands
Client
To
Server
channel
(no reply)
draft-gentric-avt-rtsp-http-00.txt

Implementation status

- Has been implemented
- Has been deployed
- Works fine

Security issues

- Same as HTTP
- Same as RTSP etc …
- base64 encoding aims at preventing some proxy implementation from blocking on trying to parse RTSP …
- the mime type can be used for filtering

Traffic on the reflector:
• suggests use HTTP CONNECT to open tunnel in proxy/firewall (this feature is used for https so unless it is implement with drastic restrictions, it would work ?)

More ideas ?

How to proceed ?