Multimedia Services on the Internet and on Cellular Networks - Current Trends -

Gonzalo.Camarillo@ericsson.com

Outline

• Introduction to the IETF
• Where are multimedia protocols developed?
• Current work in
  – MMUSIC
  – SIPPING
  – SIMPLE
  – XCON
• ICE
• Session Policies
• Push-to-Talk
IETF Structure

- Areas
  - Area directors + IETF chair = IESG

- Working Groups
  - Chairs
  - Charter
  - Mailing list

- IAB
Multimedia Work

- Traditionally in the Transport Area
  - AVT developed RTP
  - MMUSIC developed SDP, RTSP, SAP, and SIP
  - SIP, SIPPING, IPTEL, ENUM, and XCON were chartered later

- Application Area
  - SIMPLE
  - XMPP

Current Work in MMUSIC

- Internet Media Guide
  - Analysis of existing delivery protocols

- SDNng
  - XML-based

- ICE (Interactive Connectivity Establishment)
  - End-to-end STUN
SIPPING Design Teams

- Transcoding
  - B2BUA invocation
- Conferencing using SIP
  - Non-SIP work done in XCON
- Application Server interaction
  - Security
- Emergency calls
  - Routing
  - Location information conveyance

More SIPPING Work

- Configuration framework
- Session Policies
- End-to-middle security
- Request history
- Middle-to-end security
- Exploders
Current Work in SIMPLE

- MSRP
  - IM sessions
- XCAP
  - Configuration
- Different XML-based formats
- Partial notifications
- Filters
  - Limit contents and frequency

Current Work in XCON

- Conference Policy Control Protocol
- Floor Control Protocol
ICE

- ICE is a set of procedures to achieve connectivity in presence of NATs
  - The UAs gather a set of IP addresses
    - Local addresses
    - Using STUN, TURN, etc
  - Perform an offer/answer exchange
  - End-to-end STUN
    - May result in the discovery of extra IP addresses

STUN

<table>
<thead>
<tr>
<th>STUN client</th>
<th>NAT</th>
<th>STUN server</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.1.1</td>
<td>192.0.2.1</td>
<td>192.0.2.10</td>
</tr>
</tbody>
</table>

STUN Bind
s = 10.0.1.1:1010
d= 192.0.2.10:3478

STUN Response
s= 192.0.2.10:3478
d= 192.0.1.1:1010
M= 192.0.2.1:9988

STUN Bind
s= 192.0.0.1:9988
d= 192.0.2.10:3478

STUN Response
s= 192.0.2.10:3478
d= 192.0.2.1:9988
M= 192.0.2.1:9988

s = source address
d= destination address
M = mapped address
TURN

TURN client 10.0.1.1

TURN Alloc
s=10.0.1.1:1010
d=192.0.2.10:5556

TURN Response
s=192.0.2.1:1010
d=192.0.2.1:9988
M=192.0.2.10:8076

NAT 192.0.2.1

TURN Alloc
s=192.0.2.1:9988
d=192.0.2.10:5556

TURN Response
s=192.0.2.1:9988
d=192.0.2.1:19988
M=192.0.2.10:8076

TURN server 192.0.2.10

s = source address
d = destination address
M = mapped address

Offer

v=0
o=alice 2890844730 2890844731 IN IP4 host.example.com
s=
c=IN IP4 192.0.2.10
t=0 0
m=audio 8076 RTP/AVP 0
a=alt:1 1.0 : user 9kksj== 10.0.1.1 1010
a=alt:2 0.8 : user1 9kksk== 192.0.2.1 9988 192.0.2.1 9990
a=alt:3 0.4 : user2 9kksl== 192.0.2.10 8076

• TURN-derived address is the most likely to work
• RTP and RTCP ports are not adjacent in the STUN-derived address (the client did STUN twice)
Offer Arrives at the UAS

- UAS does STUN to
  - 10.0.0.1:1010
  - 192.0.2.1:9988 192.0.2.1:9990
  - 192.0.2.10:8076
- UAC does STUN to the UAS’s addresses
- Checks connectivity
- May find new addresses
  - Which may have higher q values
  - E.g., symmetric NAT and a UAS with a public (or STUN-derived IP address)

End-to-End STUN

STUN client 10.0.1.1

STUN Bind
s = 10.0.1.1:1010
d= 192.0.2.10:3478

STUN Response
s= 10.0.1.1:1010
d= 192.0.2.10:3478
M= 192.0.2.1:9988

STUN server 192.0.2.10

STUN Bind
s = 192.0.2.1:9988
d= 192.0.2.10:3478

STUN Response
s= 192.0.2.1:9988
d= 192.0.2.1:9988
M= 192.0.2.1:9988

STUN server 192.0.2.1

STUN Bind
s = 192.0.2.1:9988
d= 192.0.2.1:9988

STUN Response
s= 192.0.2.1:9988
d= 192.0.2.1:9988
M= 192.0.2.1:9988

STUN client 10.0.3.1

STUN Bind
s = 10.0.3.1:1010
d= 192.0.3.1:20000

STUN Response
s= 192.0.3.1:20000
d= 192.0.3.1:20000
M= 192.0.3.1:7777

STUN server 192.0.3.1

STUN Bind
s = 192.0.3.1:7777
d= 192.0.3.1:20000

STUN Response
s= 192.0.3.1:20000
d= 192.0.3.1:7777
M= 192.0.3.1:7777

NAT 192.0.2.1

STUN Bind
s = 192.0.2.1:9988
d= 192.0.2.10:3478

STUN Response
s= 192.0.2.1:9988
d= 192.0.2.1:9988
M= 192.0.2.1:9988

Offer/answer exchange
Session Policies

• Offer/answer happens between two UAs
  – But the network may have something to say
  – E.g., do not use high-bandwidth codecs
• Session policies allow the network to send policies to UAs
  – Session independent policies
  – Session specific policies
• The mechanism is work in progress

Push-to-Talk

• Half-duplex group communication
• Uses the IMS infrastructure
• Does not need conversational radio bearers
• Standardized by OMA
• Technology wise, a Push-to-Talk server is a conference server
SIP Signalling

INVITE (URI list) → PTT server (B2BUA) → UA
UA
UA

XML-Based URI List

--boundary1
Content-Type: application/resource-lists+xml
Content-Length: 315
Content-ID: <cn35t8j02@example.com>

<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <list>
    <entry uri="sip:bill@example.com" />
    <entry uri="sip:joe@example.com" />
    <entry uri="sip:ted@example.com" />
    <entry uri="sip:bob@example.com" />
  </list>
</resource-lists>
--boundary1--
Floor Control

(1) Request Floor
(2) Floor Granted
(3) Request Floor
(4) Queued