



62nd IETF - Minneapolis, MN, USA MMUSIC WG



Stream Tracking Description for Resource Management Guarantees in the Network

draft-guenkova-mmusic-sdp-ng-streamtrack-00

Ingo Wolf

Wolfl@t-systems.com

Teodora Guenkova-Luy

guenkova@vs.informatik.uni-ulm.de



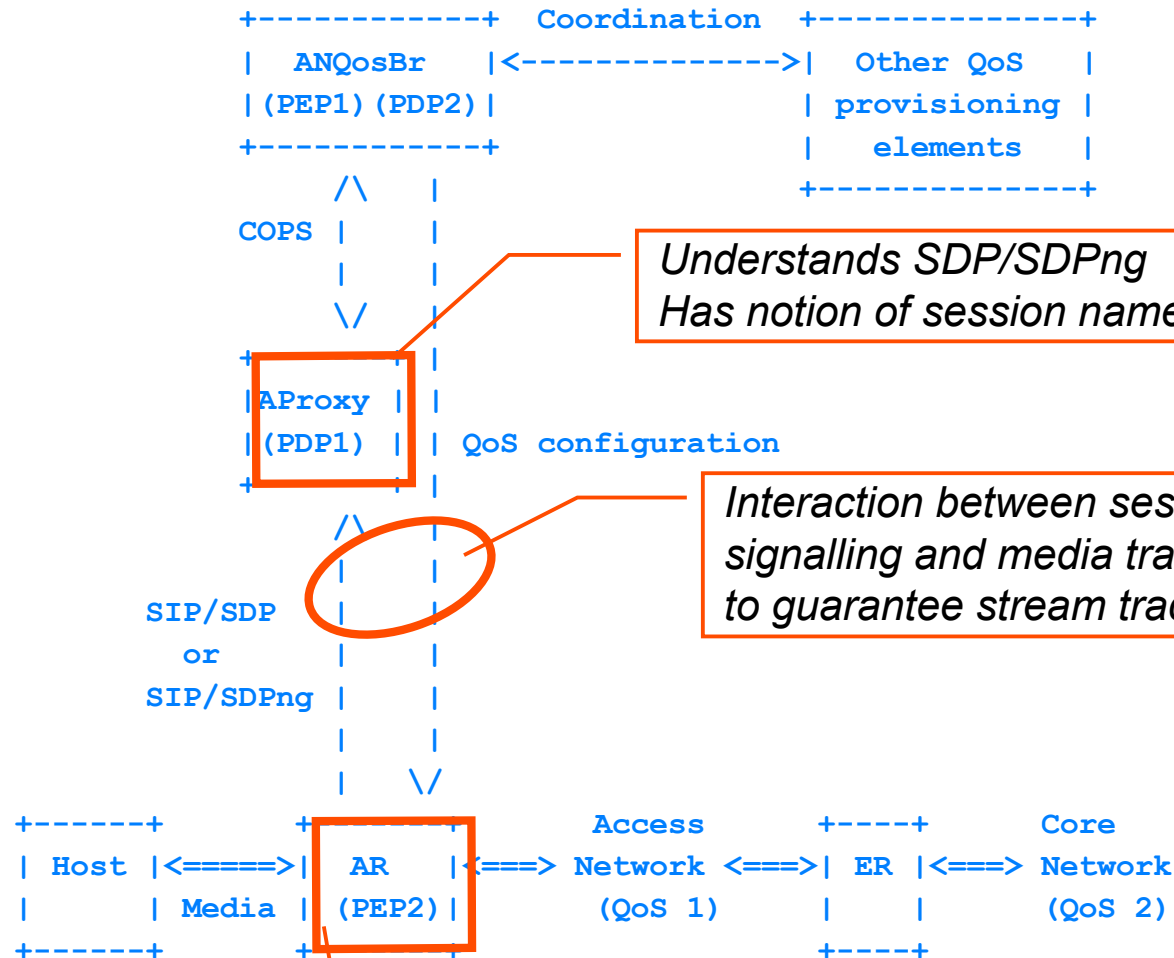
Contents

- **Motivation - Multimedia Service Provisioning Platform**
- **SDP description**
- **SDPng description**
- **SDP negotiation examples**
- **SDPng negotiation examples**
- **Security Considerations**





Motivation - Multimedia Service Provisioning Platform



*Understands SDP/SDPng
Has notion of session names*

Interaction between session signalling and media transport to guarantee stream tracking

*Does not understand SDP/SDPng
Identifies media streams to guarantee QoS upon the 5-tuple:
<src addr>, <src port>, <dest addr>, <dest port>, <transport protocol>*



SDP description



m=<media> <port>/<number of ports>/<sender port>/<number of sender ports> <proto> <fmt>

Example:

m=video 49170/2/50080/2 RTP/AVP 31

**media=video;
 RTPport1=49170; RTCPport1=49171
 RTPport2=49172; RTCPport2=49173
 RTPsendport1=50080; RTCPsendport1=50081
 RTPsendport2=50082; RTCPsendport2=50083
 protocol=RTP/AVP
 format=31**



SDP description



- Application of “a=rtcp:” (RFC3605) is problematic
 - Supports only single port definitions
 - Redefinition in a form like “m=” for multiple streams (micro-flows) necessary
 - If “a=rtcp:” is redefined - association of the port groups might be difficult

Hence:

- AProxy and AR need to have the same notion of the RTP/RTCP pairs in their application logic, e.g.
 - RTCP receiver and sender ports are ALWAYS derived algorithmically as $RTCP_recv=RTP_recv+1$ and $RTCP_send=RTP_send+1$
- SDPng is better for complex port descriptions and associations



SDPng description



```
<rtp:udp name="rtp-cfg001">
  ...
  <rtp:ip-addr ip="134.60.70.80">
    <rtp:rtp-port>49170</rtp:rtp-port>
    <rtp:rtcp-port>49171</rtp:rtcp-port>
    <rtp:rtp-port-offset>1</rtp:rtp-port-offset>
    <rtp:rtp-sendport>50080</rtp:rtp-sendport>
    <rtp:rtcp-sendport>50081</rtp:rtcp-sendport>
    <rtp:rtp-sendport-offset>1</rtp:rtp-sendport-offset>
  </rtp:ip-addr>
  ...
</rtp:udp>
```

RTCP receiver/sender ports are specified EXPLICITLY (like in draft-ietf-mmusic-sdpng-07)

Explicit association of the pairs of RTP and RTCP sender and receiver ports in the <rtp:ip-addr> container (unlike draft-ietf-mmusic-sdpng-07)

<rtp:udp> specifies only unidirectional stream descriptions

Example means the same as the previous SDP example



SDPng description



Port Offsets:

In case of n-port offsets:

```
<rtp:rtp-port>49170</rtp:rtp-port>
<rtp:rtcp-port>49171</rtp:rtcp-port>
<rtp:rtp-port-offset>n</rtp:rtp-port-offset>
```

every n-th pair of RTP/RTCP

ALWAYS calculated as $RTP_n = RTCP_{1+n}$ and $RTCP_n = RTP_{1+n+1}$

The same definition holds true for RTP/RTCP sender ports (i.e. `<rtp:rtp-sendport>`, `<rtp:rtcp-sendport>` and `<rtp:rtp-sendport-offset>`).



SDP negotiation examples



- **Example 1**

- Offer (Receiver)

- m=video 49170/2 RTP/AVP 31

- Answer (Sender)

- m=video 49170/2/50080/2 RTP/AVP 31

- **Example 2**

- Offer (Sender)

- m=video //50080/2 RTP/AVP 31

- Answer (Receiver)

- m=video 49170/2/50080/2 RTP/AVP 31

Zero has special meaning in SDP, hence for unknown receiver ports the indication “//” in the port definition

The “m=” lines can specify only unidirectional streams. Problem with “a=sendrecv” definitions.



SDPng negotiation examples



Example 1

- Offer (Receiver)

```
<rtp:ip-addr ip="134.60.70.80">
  <rtp:rtp-port>49170</rtp:rtp-port>
  <rtp:rtcp-sendport>50081</rtp:rtcp-sendport>
</rtp:ip-addr>
```

- Answer (Sender)

```
<rtp:ip-addr ip="134.60.70.80">
  <rtp:rtp-port>49170</rtp:rtp-port>
  <rtp:rtcp-port>49171</rtp:rtcp-port>
  <rtp:rtp-sendport>50080</rtp:rtp-sendport>
  <rtp:rtcp-sendport>50081</rtp:rtcp-sendport>
</rtp:ip-addr>
```

Example 2

- Offer (Sender)

```
<rtp:ip-addr ip="134.60.70.80">
  <rtp:rtcp-port>49171</rtp:rtcp-port>
  <rtp:rtp-sendport>50080</rtp:rtp-sendport>
</rtp:ip-addr>
```

- Answer (Receiver)

```
<rtp:ip-addr ip="134.60.70.80">
  <rtp:rtp-port>49170</rtp:rtp-port>
  <rtp:rtcp-port>49171</rtp:rtcp-port>
  <rtp:rtp-sendport>50080</rtp:rtp-sendport>
  <rtp:rtcp-sendport>50081</rtp:rtcp-sendport>
</rtp:ip-addr>
```

SDPng gives the possibility to group different descriptions in the <cfg> container, e.g. for “sendrecv” definitions.



Security Considerations



- The explicit definition of RTP/RTCP receiver/sender ports allows the QoS entities in the network to prove upon the session descriptions that the terminals are in line with their QoS specifications
- Firewalls and NATs
 - Need to associate signalling and session traffic
 - Necessary architecture as the one proposed in the Motivation