Multiple Packetization Time in SDP Problem statement & Requirements New solution proposal

draft-garcia-mmusic-multiple-ptimes-problem-02.txt

IETF 71, MMUSIC WG

Philadelphia, 3/2008

Marc Willekens

Miguel Garcia

Peili Xu

Problem

- SDP defines the ptime/maxptime
  - common parameter for all media formats in m-line
  - not possible to specify this in f(codec)

```
m=audio 49170 RTP/AVP 0 4 8
a=ptime:30
a=maxptime:60
```

Changes in version 02

- New clarification section
- New solution section

New proposal

- don’t introduce extra complexity
  - no signaling overkill
  - don’t punish implementations that conform to the RFCs
  - IETF architecture principle:
    - “be strict when sending” & “be tolerant when receiving”
- keep ptime/maxptime on media level
- Decouple the ptime/maxptime value that the endpoint wants to use for reception (i.e. what is expressed in SDP) from the value that the endpoint uses for sending)
- New algorithm helps to determine the actual value of ptime and maxptime that the sender should use.

New sources for ptime/maxptime

- Static
  - Default values or manually defines values in the end-device.
- Dynamic
  - Defined by the network architecture.
- Indicated
  - Proposed value from the receiving side.

Algorithm - parameters

- **Codec independent** parameters
  - p   vector with all provided ptime values
    static, dynamic, indicated
  - mp   vector with all provided maxptime values

- **Codec dependent** parameters
  - fc   frame size codec related
  - mc   maxptime codec related
    f(codec, frame size, frame datarate, MTU)

Algorithm - method

- packetization time for media transmission
  pt = f(p, mp, fc, mc)

- Take min. value of “mp” and “mc”
- Take max. value of “p”
- Normalize in function of the codec frame size.

Examples

p	mp	fc	mc	pt
20	60	30	100	30
20	20	30	100	0
30	30	30	100	30
60	80	30	100	60
120	200	10	100	100
[40,50,20]	200	10	100	50
[40,50,20]	[40,50,20]	10	100	20
[120,40]	[150,200,100]	10	100	100

Advantages

- Basic idea of ptime is kept.
  - No new SDP parameters
  - No new interpretations
- Strict in sending, tolerant in receiving
  - Sends with maximum allowed ptime <= minimal maxptime
- Different sources for ptime/maxptime
  - static, dynamic, indicated
- Local policy in end-device
- Same algorithm for sending/receiving
- Small and straight forward algorithm
  - distinction between coded dependent/independent parameters

Next steps

- Document his methodology in a separate draft/RFC?
- Include this methodology in a newer version of the SDP when/if that happens?
- Throw this draft on the garbage bin and forget about it?