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 03

- Move of different sections to appendix:
  - Related RFCs for ptime
  - Ad-hoc solutions for multiple ptime
  - Some background info
  - Some small editorial changes

BCP Solution

- keep ptime/maxptime on media level
- No new parameters in the SDP
- Easy algorithm to determine ptime/maxptime
  - Differentiate between codec related and codec independent parameters
  - Rules for SDP parameter indications
  - Rules for packetization time for media transmission
Sources for ptime/maxptime

- **Static**
  - Default or manually defined values in the end-device.

- **Dynamic**
  - Defined by the network architecture.

- **Indicated**
  - Proposed value from the receiving side (from SDP)

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

<table>
<thead>
<tr>
<th>p</th>
<th>mp</th>
<th>fc</th>
<th>mc</th>
<th>pt</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>d</td>
<td>i</td>
<td>s</td>
<td>d</td>
</tr>
<tr>
<td>20</td>
<td>60</td>
<td>30</td>
<td>100</td>
<td>30</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>30</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>30</td>
<td>30</td>
<td>100</td>
<td>30</td>
</tr>
<tr>
<td>60</td>
<td>80</td>
<td>30</td>
<td>100</td>
<td>60</td>
</tr>
<tr>
<td>20</td>
<td>60</td>
<td>20</td>
<td>100</td>
<td>20</td>
</tr>
<tr>
<td>60</td>
<td>80</td>
<td>20</td>
<td>100</td>
<td>60</td>
</tr>
<tr>
<td>70</td>
<td>200</td>
<td>20</td>
<td>100</td>
<td>60</td>
</tr>
<tr>
<td>120</td>
<td>60</td>
<td>20</td>
<td>100</td>
<td>60</td>
</tr>
<tr>
<td>120</td>
<td>200</td>
<td>10</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>40</td>
<td>50</td>
<td>20</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>40</td>
<td>50</td>
<td>40</td>
<td>50</td>
<td>10</td>
</tr>
<tr>
<td>120</td>
<td>40</td>
<td>150</td>
<td>200</td>
<td>100</td>
</tr>
<tr>
<td>90</td>
<td>40</td>
<td>20</td>
<td>150</td>
<td>200</td>
</tr>
<tr>
<td>60</td>
<td>30</td>
<td>80</td>
<td>30</td>
<td>60</td>
</tr>
</tbody>
</table>
Next steps

- Is there interest in this method?
- Add use cases based on real-life problems and indicate how this BCP can solve different interworking issues