Advanced Codec Negotiation

George Joseph
Asterisk Core Developer
Why Mess With It?

- Codec Negotiation is one of the least understood areas of Asterisk.
  - What we think it does, what we think it should do, and what it really does are all different.
- It doesn’t allow the granularity of control we need.
- There’s no feedback to the calling call leg.
- It’s hard to modify and debug.
  - It’s spread out over many modules: app_dial, features, channel, bridge, chan_pjsip, res_pjsip_session, res_pjsip_sdp_rtp, etc.
Call Flow

The Core
app_dial, dialplan, bridging, etc.

Alice's Session
res_pjsip_sdp_rtp
res_pjsip_session

Alice's Channel
chan_pjsip

Alice's Codec Endpoint Config
res_pjsip_session
res_pjsip_sdp_rtp

Bob's Session
res_pjsip_sdp_rtp
res_pjsip_session

Bob's Channel
chan_pjsip

Bob's Codec Endpoint Config
res_pjsip_session
res_pjsip_sdp_rtp
Fewer Modules Involved

Asterisk

Alice’s Session
- res_pjsip_sdp_rtp
- res_pjsip_session

Alice’s Channel
- chan_pjsip

Alice’s Codec Endpoint Config

Alice

The Core
- app_dial
- dialplan
- bridging, etc.

Bob’s Session
- res_pjsip_sdp_rtp
- res_pjsip_session

Bob’s Channel
- chan_pjsip

Bob’s Codec Endpoint Config

Bob

SDP Offer
- Alice
- Bob

SDP Answer
Well Defined Control Points

Alice

res_pjsip_session
res_pjsip_sdp_rtp
Alice’s Session

Alice’s Codec Endpoint Config
chan_pjsip

Alice’s Channel

Asterisk

The Core
app_dial, dialplan, bridging, etc.

Bob

res_pjsip_session
res_pjsip_sdp_rtp
Bob’s Session

Bob’s Codec Endpoint Config
chan_pjsip

Bob’s Channel

SDP Offer
SDP Answer
SDP Offer
SDP Answer
We’re Always Resolving Two Topologies
# How Are We Resolving?

<table>
<thead>
<tr>
<th>CP</th>
<th>Pending Topology</th>
<th>Configured Topology</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alice’s SDP</td>
<td>Alice’s Endpoint Config</td>
</tr>
<tr>
<td>3</td>
<td>What was received from the core</td>
<td>Bob’s Endpoint Config</td>
</tr>
<tr>
<td>4</td>
<td>Bob’s SDP Answer</td>
<td>Bob’s SDP Offer</td>
</tr>
<tr>
<td>5</td>
<td>What was received from the core</td>
<td>What we sent to the core</td>
</tr>
</tbody>
</table>

## Option | Description | Values
---|-------------|---------
Preference | Which topology is preferred? | ● Pending  
|           |                         | ● Configured  
Operation | What are we going to to with them? | ● Intersect: Only use the codecs common to both topologies with the ones in the preferred list first.  
|           |                         | ● Merge: Use all codecs in both topologies with the ones in the preferred list first.  
|           |                         | ● Preferred Only: Just use the ones in the preferred topology.  
|           |                         | ● Non-preferred Only: Just use the ones in the non-preferred topology.  
Keep | Do we want to keep all codecs or just the first one? | ● All: Keep everything that made it past the operation.  
|       |                         | ● First: Keep only the first codec that made it past the operation.  
Transcode | Do we allow it? | ● Allow  
|         |                         | ● Prevent  

© Copyright 2020 Sangoma Technologies
Alice Offers g722, ulaw, alaw, g729

We send ulaw, alaw, g722 to the core

We got ulaw, g722 from the core

Bob’s incoming answer prefs: prefer pending, intersect, first, allow

We got ulaw from the core

We send ulaw to the core

Bob’s outgoing offer prefs: prefer pending, intersect, first, allow

Alice’s incoming offer prefs: prefer configured, intersect, all, allow

We answer Alice with ulaw

Alice’s outgoing answer prefs: prefer pending, intersect, first, allow

Bob answers with ulaw

We offer Bob ulaw
Transcoding Success Example

Alice Offers g722, ulaw, g729

Alice’s incoming offer prefs: prefer configured, intersect, all, allow

We send ulaw, g722 to the core

We got ulaw, g722 from the core

Bob’s outgoing offer prefs: prefer pending, intersect, first, allow

Bob’s incoming answer prefs: prefer pending, intersect, first, allow

We send alaw to the core and set up alaw <> ulaw transcoding

No codecs in common but transcoding is allowed by both endpoints

We offer Bob alaw

Bob answers with alaw

We answer Alice with ulaw

Alice’s outgoing answer prefs: prefer pending, intersect, first, allow

Alice’s incoming offer prefs:
prefer configured, intersect, all, allow

The Core
app_dial, dialplan, bridging, etc.

We send ulaw, g722 to the core

We got ulaw, g722 from the core

Bob’s outgoing offer prefs:
prefer pending, intersect, first, allow

Bob’s incoming answer prefs:
prefer pending, intersect, first, allow

Alice’s Allow ulaw, alaw, g722, opus

Opus

Bob’s Allow: alaw, g729

SDP Offer

SDP Answer

SDP Offer

SDP Answer

SDP Offer

SDP Answer
Transcoding Failure Example

Alice’s incoming offer prefs: prefer configured, intersect, all, deny

We send alaw, g729 to the core

We got alaw, g729 from the core

Bob’s outgoing offer prefs: prefer pending, intersect, first, allow

We can’t proceed! Both topologies need to allow transcoding.

We send 5XX back to Alice

Alice Offers alaw, g729
Wrap Up

Make sure your scenarios are covered by the logic!
Thank You! Contact us.

🌐 sangoma.com

✉️ gjoseph@sangoma.com
Lunch Break until 1:15PM -0500