

Three-Way Calling using the Conferencing-URI

Introduction

With the deployment of VoIP users expect to have the same functionality and features that are available with a landline phone service. This document will discuss how an AOS voice Sip Trunk can establish a 3-Way conference call. It is assumed that two calls have already been established with a single analog voice user on the AOS voice device (A device running Adtran OS capable of SIP functionality) as an endpoint for each of the two calls. Endpoint A can be the origin or destination of call 1. Endpoint B is an AOS voice device. Call 2 must be originated by Endpoint B destined for Endpoint C. Both Endpoint A and Endpoint C must be reached through the same Sip Trunk on Endpoint B. A diagram for the call setup is shown in figure 1 below.

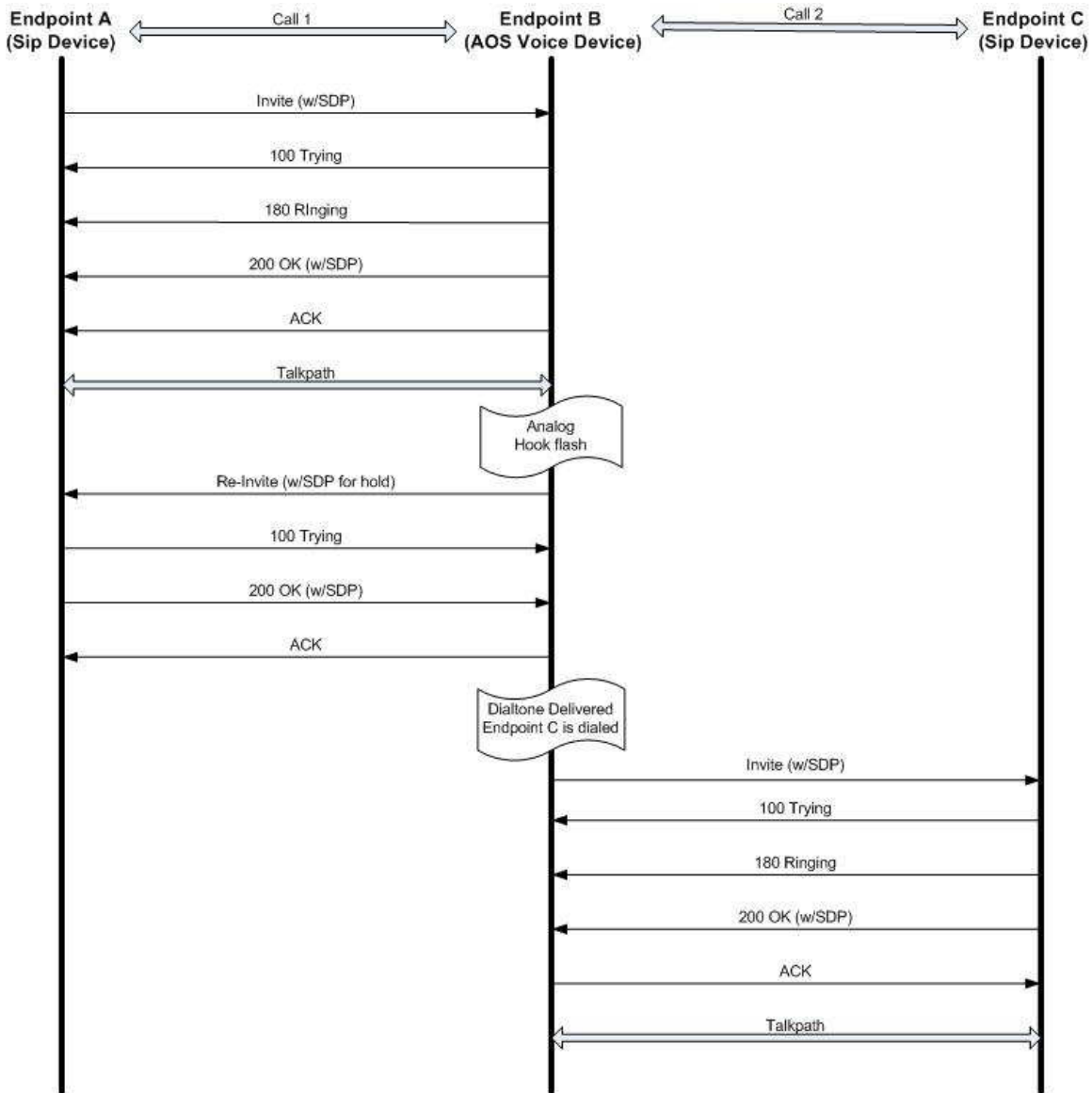


Figure 1. Establishing two individual calls to be conferenced

Initiating the 3-Way call

Assuming the calls above, a three-way call is initiated by issuing a second hook flash from Endpoint B. AOS devices do not act as media servers and as a result the audio streams for the three endpoints can not be combined internally. This requires a third party conference server to act as the media server and combine the three audio paths so all parties can talk to one another simultaneously.

After the hook flash the AOS device will enter conferencing only if there is a configured conferencing-uri for the Sip Trunk being used. This is shown in figure 2 below. If there is no conferencing-uri configured the hook flash will cause the AOS device to swap media streams between endpoints A and C. After entering conferencing mode the first event will be an invite sent to the conferencing-uri from the AOS device. An example of this invite is shown below in figure 3.

```
voice trunk T02 type sip
no reject-external
sip-server primary 10.1.3.9
registrar primary 10.1.3.9
conferencing-uri "conf"
codec-group global
```

Figure 2. Sip trunk configuration

```
Tx: INVITE sip:conf@10.1.3.9:5060 SIP/2.0
From: ""<sip: Endpoint_B@10.1.3.9:5060;transport=UDP>;tag=29e7dc8-0-13c4-9448f-7467e87a-9448f
To: <sip:conf@10.1.3.9:5060;transport=UDP>
Call-ID: 2a01ad8-0-13c4-9448f-74a07d0d-9448f@10.1.3.9
CSeq: 1 INVITE
Via: SIP/2.0/UDP 10.19.210.6:5060;branch=z9hG4bK-9448f-2433cec6-43358e7b
Max-Forwards: 70
Supported: 100rel,replaces
Allow: ACK, BYE, CANCEL, INFO, INVITE, NOTIFY, OPTIONS, PRACK, REFER, REGISTER
User-Agent: ADTRAN_Total_Access_916/12.06.00.E
Contact: <sip: Endpoint_B@10.19.210.6:5060;transport=UDP>
Content-Type: application/SDP
Content-Length: 224

v=0
o=- 1168358206 1168358206 IN IP4 10.19.210.6
s=-
c=IN IP4 10.19.210.6
t=0 0
m=audio 10012 RTP/AVP 18 0 101
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

Figure 3. Invite to conferencing-uri

As seen in figure 3 above, the invite is sent to the configured *conferencing-uri@sip-server*. If the conferencing-uri contained a full URI including the domain, the Sip Server should not be appended. If the sip-server is the conference server it should handle this invite. If the conferencing server is a separate entity the sip server should send an invite to the conferencing server as though it were any other sip endpoint. The conferencing server will respond with a 100 Trying and then a 200 OK. The 200 OK is shown in figure 4 below.

```

Rx: SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.19.210.6:5060;branch=z9hG4bK-9448f-2433cf83-7f500340
CSeq: 2 INVITE
Call-ID: 2a01ad8-0-13c4-9448f-74a07d0d-9448f@10.1.3.9
From: "" <sip: Endpoint_B@10.1.3.9:5060;transport=UDP>;tag=29e7dc8-0-13c4-9448f-7467e87a-9448f
To: <sip:conf@10.1.3.9:5060;transport=UDP>;tag=02631518-1dd2-11b2-ac3a-b03162323164+02631518
Content-Type: application/sdp
Supported: timer
Contact: <sip:conf=02631518-1dd2-11b2-ac3a-b03162323164@10.1.3.9:5065;transport=udp>;isfocus
Allow: INVITE,BYE,ACK,CANCEL,PRACK,REFER,OPTIONS,REGISTER,NOTIFY
Content-Length: 187

v=0
o=- 127102580 127102580 IN IP4 10.1.3.9
s=SIP Call
c=IN IP4 10.1.3.11
t=0 0
m=audio 1130 RTP/AVP 0 101
a=rtpmap:0 pcmu/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15

```

Figure 4. 200 OK from conferencing server

Transferring Audio to the Conferencing Server

The 200 OK will contain the IP address of the conferencing server. At this point a third call now exists between the AOS device (endpoint B) and the conference server. This is a good start, but all three endpoints are now on completely different calls, far from the three-way conference that is the ultimate goal. In order to get the two original calls to the conference server the AOS device will transfer each call to the conference server. This sounds simple but can be a little complicated.

First there is a problem inherent with conference servers. You need to be able to program one conferencing-uri across the network which points back to one conference server, but that conference server must be able to handle multiple separate conferences. This is done by creating a new unique URI for each individual conference on the server. The conference server sends this URI back in the 200OK. This can be seen in figure 4 above in the Contact Field. Contact: <sip:conf=02631518-1dd2-11b2-ac3a-b03162323164@10.1.3.9:5065 > This contact field is where all new calls must be transferred to so they are associated with this conference. It is important to note that this contact field must contain a full URI or the conference will fail.

```

Tx: REFER sip: Endpoint_A@10.1.3.9:5075;transport=udp SIP/2.0
From: <sip: Endpoint_B@pq.adtran.com>;tag=29e02e0-0-13c4-9445c-38cbc93e-9445c
To: <sip: Endpoint_A@pq.adtran.com>;tag=e4064f0e-1dd1-11b2-ac3a-b03162323164+e4260
Call-ID: 882873-1177735231@10.1.3.9
CSeq: 2 REFER
Via: SIP/2.0/UDP 10.19.210.6:5060;branch=z9hG4bK-9448f-2433d102-7b685005
Refer-To: <sip:conf=02631518-1dd2-11b2-ac3a-b03162323164@10.1.3.9>
Max-Forwards: 70
Supported: 100rel,replaces
Allow: ACK, BYE, CANCEL, INFO, INVITE, NOTIFY, OPTIONS, PRACK, REFER, REGISTER
User-Agent: ADTRAN_Total_Access_916/12.06.00.E
Contact: <sip:10.19.210.6:5060;transport=UDP>
Content-Length: 0

```

Figure 5. Refer sent to Endpoint A

To initiate the transfer a Refer is sent to the sip server for each of the existing calls. Figure 5 above shows the refer that is sent for Call 1, a nearly identical refer is sent for Call 2. Notice that the refer-to header contains the contact header of the 200OK. After the

refer is sent it is expected that the sip server will send a 202 Accepted followed by a Notify for each refer that was sent. The Notify is necessary because the Refer creates an implicit subscription between Endpoint B and Endpoint A. The Notify message updates Endpoint B on the status of the transfer. It is expected that the Notify will contain a response code of 200 OK and the state will be terminated as shown in figure 6 below. The response code of 200 OK means that the transfer was a success and the terminated state means that no further messages will be sent on that subscription. The AOS Device will respond to this Notify with a 200 OK. The purpose of this is to assure the transfer succeeds before the calls are terminated. If the Notify contained a failure message instead of the 200 OK it would be possible to resume the calls.

```
Rx: NOTIFY sip: Endpoint_B@10.19.210.6:5060;transport=UDP SIP/2.0
From: <sip: Endpoint_A@10.1.3.9:5060;transport=UDP>;tag=fa679136-1dd1-11b2-ac3a-b03162323164+fa679
To: "" <sip: Endpoint_B@10.1.3.9:5060;transport=UDP>;tag=29d60c8-0-13c4-94481-1f1f5499-94481
Call-ID: 882873-1177735231@10.1.3.9
CSeq: 2 NOTIFY
Via: SIP/2.0/UDP 10.1.3.9:5065;branch=z9hG4bK1279596437357852
Max-Forwards: 70
Event: refer
Content-Type: message/sipfrag
Subscription-State: terminated
Supported: timer
Content-Length: 14

SIP/2.0 200 OK
```

Figure 6. Notify message from the sip server

Aside from the messages above, when the sip server receives the refer it will invite the endpoint in the refer-to header, which is the conference server URI with the SDP of Endpoint A. Once it has SDP information from the conference server it will re-invite the Endpoint A using the SDP information from the conference server. This causes Endpoint A to send and receive RTP from the conference server instead of Endpoint B. Endpoint A does not know that it is talking to a new endpoint necessarily, just that it is sending its RTP to a new IP address. Figure 7 shows the re-invite that Endpoint A receives. The same process is used for Endpoint C.

```
Rx: INVITE sip: Endpoint_A@10.19.210.6:5060;transport=UDP SIP/2.0
From: <sip: Endpoint_B@10.1.3.9:5060;transport=UDP>;tag=e4064f0e-1dd1-11b2-ac3a-b03162323164+e4064
To: "" <sip: Endpoint_A@10.1.3.9:5060;transport=UDP>;tag=29e3da8-0-13c4-9445c-5c0317e2-9445c
Call-ID: 2a00f38-0-13c4-9445c-22a14f06-9445c@10.1.3.9
CSeq: 2 INVITE
Via: SIP/2.0/UDP 10.1.3.9:5065;branch=z9hG4bK2015072984463083
Contact: <sip: Endpoint_B@10.1.3.9:5065;transport=udp>
Max-Forwards: 70
Content-Type: application/sdp
Supported: timer
Min-SE: 1800
P-Asserted-Identity: "conf=02631518-1dd2-11b2-ac3a-b03162323164" <sip:conf=02631518-1dd2-11b2-ac3
Content-Length: 187

v=0
o=- 127058719 127058721 IN IP4 10.1.3.9
s=SIP Call
c=IN IP4 10.1.3.11
t=0 0
m=audio 1138 RTP/AVP 0 101
a=rtpmap:0 pcmu/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

Figure 7. Re-Invite received by Endpoint A

Cleaning Up the Old Calls

Now that all parties have been connected to the conference server it is necessary to clean up any remnants of calls that are no longer necessary. Endpoint B still thinks that it has calls established to Endpoints A and C. In order to clear these calls, the sip server sends a Bye for each call after it accepts the Refer from Endpoint B. This causes Endpoint B to tear down Call 1 and Call 2. This Bye can be seen below in figure 8. The AOS Device should respond to this Bye with a 200 OK.

```
Rx: BYE sip: Endpoint_B@10.19.210.6:5060;transport=UDP SIP/2.0
From: <sip: Endpoint_C@10.1.3.9:5060;transport=UDP>;tag=fa679136-1dd1-11b2-ac3a-b03162323164+fa679
To: "" <sip: Endpoint_B@10.1.3.9:5060;transport=UDP>;tag=29d60c8-0-13c4-94481-1f1f5499-94481
Call-ID: 2a016f8-0-13c4-94481-52dd9b10-94481@10.1.3.9
CSeq: 1 BYE
Via: SIP/2.0/UDP 10.1.3.9:5065;branch=z9hG4bK601050435356403
Max-Forwards: 70
Supported: timer
Content-Length: 0
```

Figure 8. Bye received from sip server