

End-to-End SIP Session-ID at IETF86

~~draft-jones-insipid-session-id-02~~
draft-ietf-insipid-session-id-00

13 March 2013

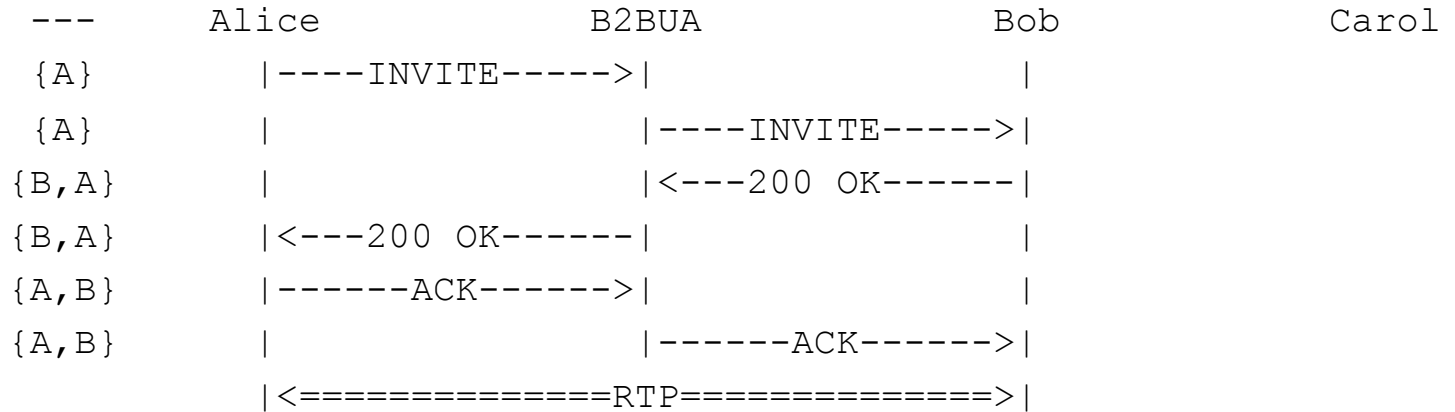
James Polk, Paul Jones
Gonzalo Salgueiro, Chris Pearce

Why an E2E Session-ID?

- Identify issues in the network (debugging)
- Track sessions as they move (e.g., transfer)
- Enable monitoring or recording of sessions (with proper end-to-end identification)
- Associate sessions that are related (e.g., participants in a multipoint conference or part of a targeted single or multi-party session to be recorded)

Basic Session-ID Construction with 2 UUIDs

Session-ID



Known Open Issues from Atlanta (1/2)

- Fix ABNF
- Could show a multiple-focus conferencing scenario
- ~~Need RTCP text if there is an RTCP requirement.~~
- Need more text covering when and when not to allow intermediaries to insert UUIDs.
 - Including what to do if a intermediary figures out one UA is not Session-ID aware.

Known Open Issues from Atlanta (2/2)

- Get more info about “the old version” reactions for greater interoperability
 - BTW we don’t want to use the term “version 0”

Addressed in -02 (and in WG -00)

- we changed the ABNF (seen on the next slide)
- clarified where appropriate when a call flow starts the a communications session already exchanging RTP packets, that it was set-up with the call flow in s9.1 which established the Session-ID for the communications session.
- clarified what purpose the reINVITEs have in each call flow that have reINVITEs.
- Clarified when intermediaries can and cannot insert a UUID on behalf either UA.
- changed in s10 (Compatibility with Previous versions) “v0” and “v1” to "old" and "new", because we don't have a version field in the Session-ID that would make this appropriate.
- Added a comment in the Security Cons section that this spec (or the Session-ID itself) MUST NOT be used for billing.
- we made minor corrections to nits

Current ABNF for -02 (and in WG -00)

session-id = "Session-ID" HCOLON local-uuid
*(SEMI sess-id-param)

local-uuid = sess-uuid

remote-uuid = sess-uuid

sess-uuid = 32(DIGIT / %x61-66) ;32 chars of [0-9a-f]

sess-id-param = remote-param / generic-param

remote-param = "remote" EQUAL remote-uuid

Cascading Conference Bridges (1/2)

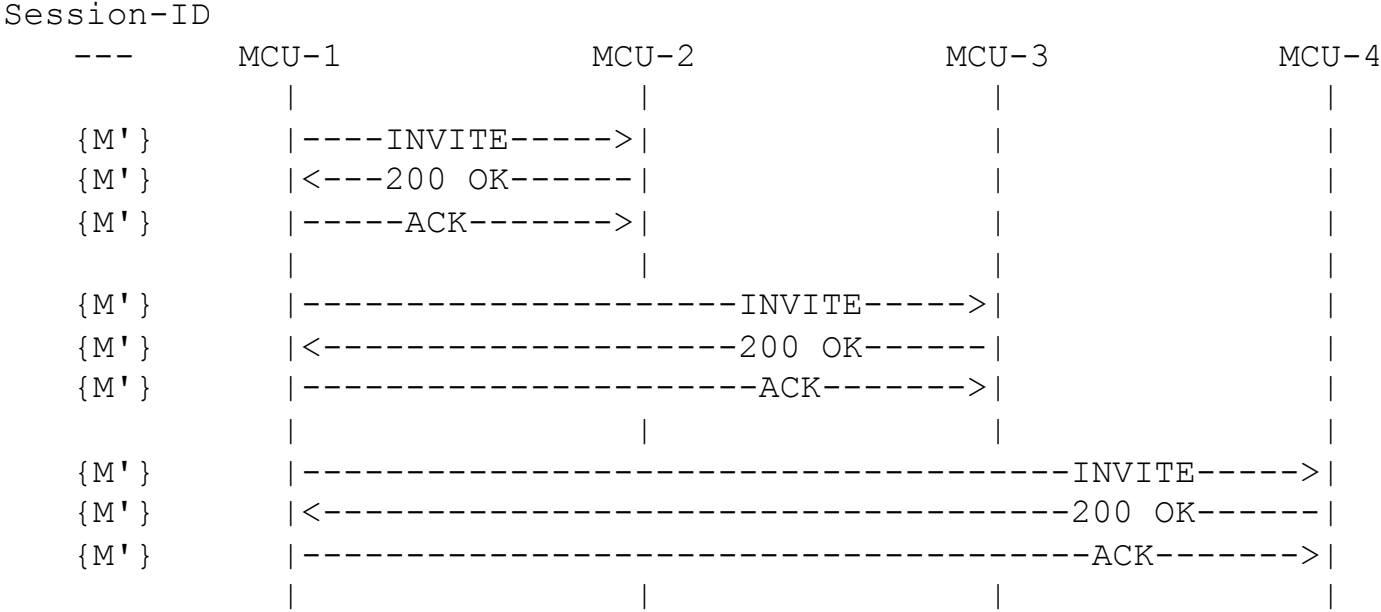
- just for Mary (just kidding), we included the flows necessary to allow Cascading Conference Bridges by using the RFC 4579 defined Contact header parameter "isfocus" to identify a communications session between MCUs meant to exchange the same UUID that a single MCU would provide.
- MCUs Communicating Session UUID for Bridge

```
Session-ID
---      MCU-1              MCU-2              MCU-3              MCU-4
      |                    |                    |                    |
{M' }   |-----INVITE----->|                    |                    |
{M' }   |<----200 OK-----|                    |                    |
{M' }   |-----ACK----->|                    |                    |
      |                    |                    |                    |
```


Cascading Conference Bridges

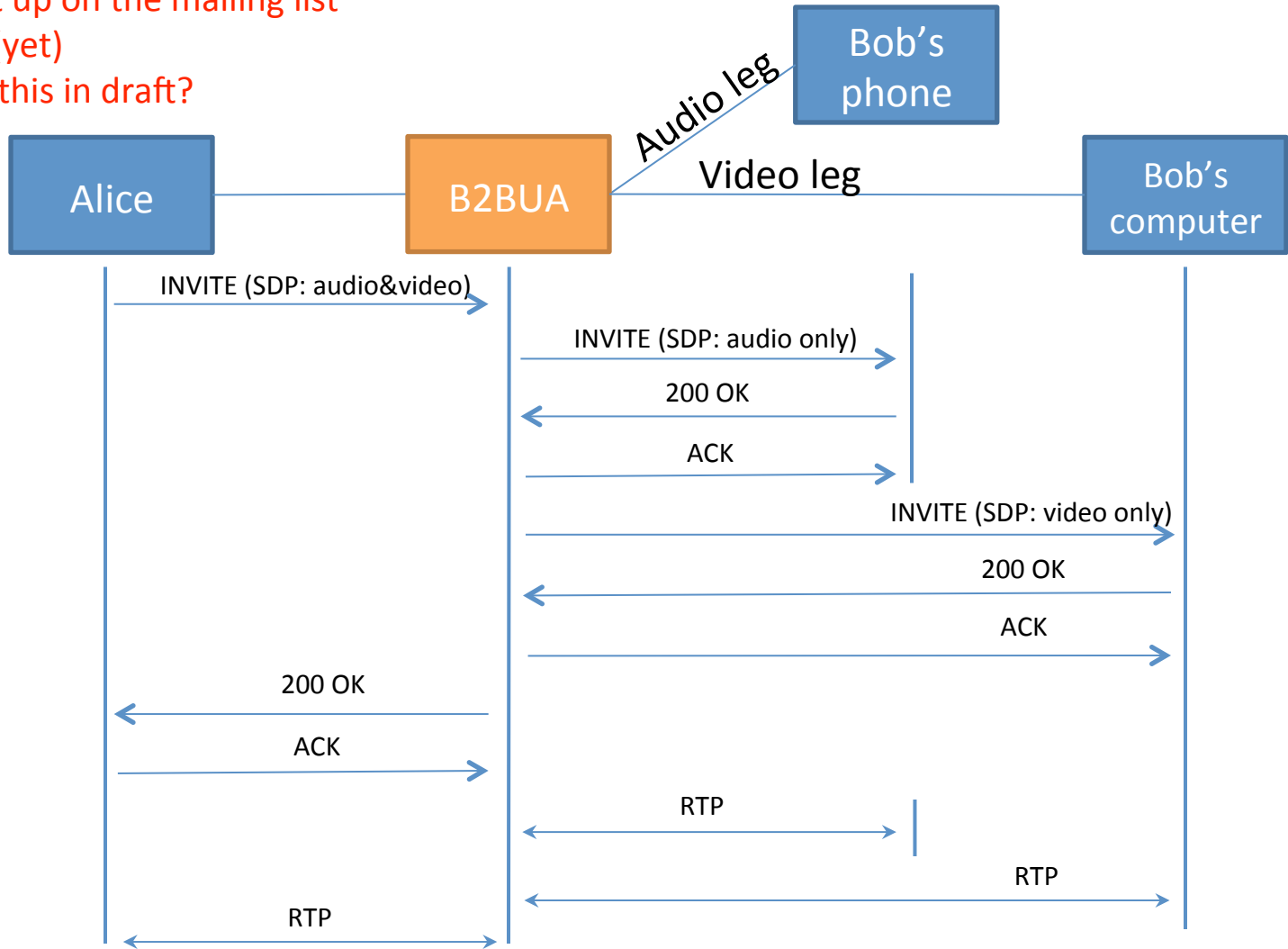
(2/2)

MCU Communicating Session UUID to More than One



Session-ID and Disaggregated Media

- Idea brought up on the mailing list
- Not in draft (yet)
- Do we want this in draft?



Known Open Issues

- We need to get more information about how an “old” version would interoperate with a “new” version.
 - Do we (as a WG) make the “old version” a Historic RFC?
- Potentially adding to next WG revision (i.e., WG -01):
 - “sess-uuid = 32(HEXDIG) ; HEXDIG defined in RFC 5234”
 - Normative text in section 5 or 6 *if* Cascading MCUs is acceptable to WG.
- Does the WG want to entertain the idea of how the Session-ID plays into “Disaggregated Media”?
 - Normative text in section 5 or 6 *if* Disaggregated Media is acceptable to WG.
- More...?

Next Steps

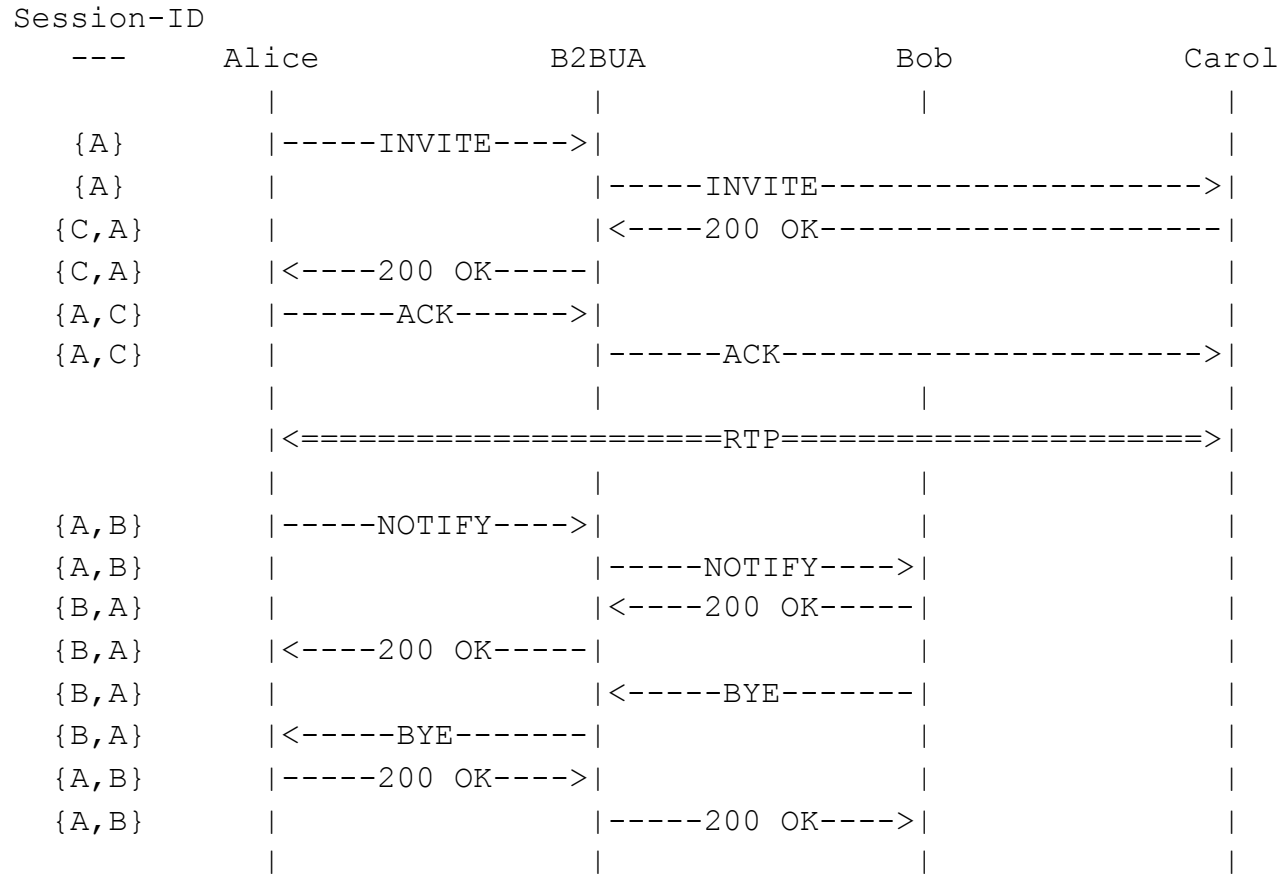
- ~~Make~~ Made WG item
- Meet in Berlin about this ID
- ...meet in Hawai'i...

Background Message Flows

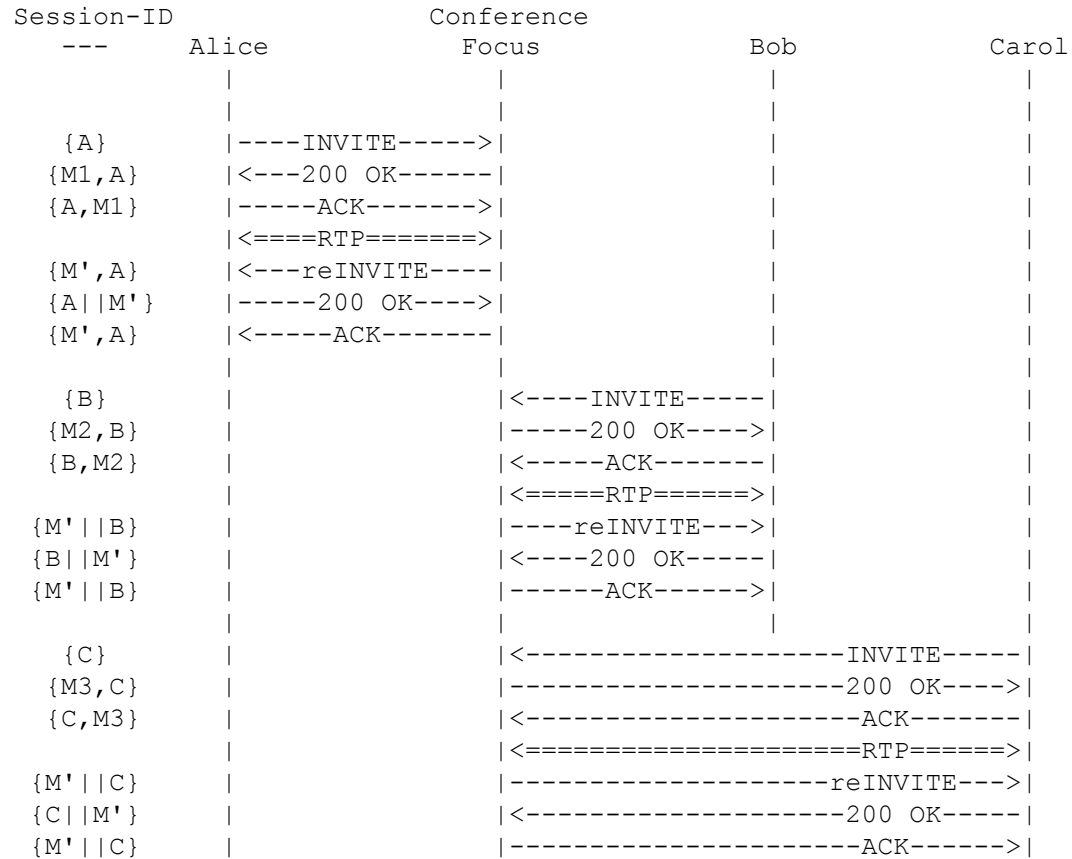
Basic Call Transfer using REFER (1/2)

```
Session-ID
---      Alice          B2BUA          Bob          Carol
        |                |                |                |
        |<=====RTP=====>|                |                |
{B,A}   |                |<---reINVITE---|                |
{B,A}   |<---reINVITE---|                |                |
{A,B}   |-----200 OK---->|                |                |
{A,B}   |                |-----200 OK---->|                |
{B,A}   |                |<-----ACK-----|                |
{B,A}   |<-----ACK-----|                |                |
        |                |                |                |
{B,A}   |                |<-----REFER-----|                |
{B,A}   |<-----REFER-----|                |                |
{A,B}   |-----200 OK---->|                |                |
{A,B}   |                |-----200 OK---->|                |
{B,A}   |                |<-----ACK-----|                |
{B,A}   |<-----ACK-----|                |                |
{A,B}   |-----NOTIFY---->|                |                |
{A,B}   |                |-----NOTIFY---->|                |
{B,A}   |                |<-----200 OK-----|                |
{B,A}   |<-----200 OK-----|                |                |
```

Basic Call Transfer using REFER (2/2)



Single Focus Conferencing



Single Focus Conferencing using WebEx

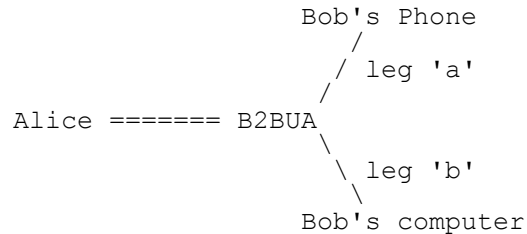
Session-ID	Alice	Conference	Bob	Carol
---		Focus		
		< ** HTTPS ***** >		
		Transaction		
{M}		<-----INVITE----->		
{A M}		-----200 OK----->		
{M A}		<-----ACK----->		
		<=====RTP=====>		
		< ** HTTPS ***** >		
		Transaction		
{M}		-----INVITE----->		
{B M}		<-----200 OK----->		
{M B}		-----ACK----->		
		<=====RTP=====>		
		<***** HTTP ***** >		
		Transaction		
{M}		-----INVITE----->		
{C M}		<-----200 OK----->		
{M C}		-----ACK----->		
		<=====RTP=====>		

Basic 3PCC for two UAs

```
Session-ID
---      Alice                B2BUA                Bob                Carol
      |                        |                        |
      |                        |                        |
{X}     |<-----INVITE-----|                        |
{A,X}   |-----200 OK----->|                        |
{A}     |                        |-----INVITE----->|
{B,A}   |                        |<----200 OK-----|
{B,A}   |<-----ACK-----|                        |
{A,B}   |                        |-----ACK----->|
      |<=====RTP=====>|
```

Session-ID and Disaggregated Media

- From a Session-ID point of view, I believe if you are wanting the service Brett is describing, then it'll look something like this *logically*



Where it is the case that the B2BUA merges the two responses from Bob's 2 UAs into a single one response, unless there are conflicts; then Brett's right, some choices will need to be made, including dropping one signaling leg.

A Session-ID can (right now) only have two parts, and I believe from Alice's point of view, she ought to see a consistent other UUID making up the full Session-ID.

However,

- from the B2BUA to Bob's phone, the Session-ID should be (M,Bp)

and

- from the B2BUA to Bob's computer, the Session-ID should be (M,Bc)

Just as the B2BUA merges the SDPs together (and at least changes header-values), it also maintains a single Session-ID towards Alice.

Now, how does the B2BUA pick which of Bob's UAs to send towards Alice... do we want to make that uniform? Can we make that uniform?

Answers: probably can't - unless we treat this call as a conference

caveat: someone clever might come up with a way, but for every scenario of Bob's UAs...