Packet multimedia experiments since 1980s
- Audio/video tools + protocols for A/V over IP
- Conference announcement and control protocols

First IETF Audiocast (1992)

Since then: IETF sessions on the Mbone
- Audio + video (+ sometimes slides)

Other uses of Mbone conferencing
- Lectures, seminars, project meetings, …
- Broadcasting NASA missions, concerts, …
IETF Conferencing Architecture

Conference Control  Audio Video  Media Streaming  Shared Apps  Session Direct.
RSVP  RTP / RTCP  RTSP  SAP  SIP  HTTP  SMTP
UDP  TCP
IP / IP Multicast
Integrated / Differentiated Services Forwarding

IETF Conferencing Model

Workshop 1. Create
Descr.: IRR 2001 SIP Congress
Orig.: Jörg Ott jo@ipdialog.com
Info: http://www.ipdialog.com/
Start: 21.05.2001 / 09.00
End: 24.05.2001 / 17.00
Media: Audio GSM 224.1.6.7/49000
Media: Video H.263 224.1.6.8/49100

2a. Disseminate
SAP
NNTP
HTTP

2b. Invite
E-Mail
SIP

3. Join

4. Media Streams
SIP and Conferencing over Time...

- Origin: MMUSIC
  Multiparty Multimedia Session Control

- From Invitation… to initiation, modification, and termination
- From Multiparty… to point-to-point-focused
- From Multimedia… to voice-centric

The latter is not SIP — but it is the way SIP is looked at today in many cases.

The Role of SIP in Conferencing

- INITIATE a call or conference
- JOIN a conference
- LEAVE a conference
- INVITE participants
- EXPEL participants?
- CONFIGURE media streams
- SHARE state? CONTROL conference?
SIP and Multiparty Conferencing

- SIP signaling relationships
  - Central (bridge, endpoint) vs. mesh
- Media distribution
  - Unicast vs. multicast
- Media mixing
  - Centralized (bridge, endpoint) vs. decentralized
- Conference creation
  - ad-hoc vs. scheduled
  - “dial-in” vs. “dial-out” vs. equal peers

Centralized Signaling: Bridge

Conference Bridge / MCU (SIP UA)

- SIP UA (Ann)
- SIP UA (Dave)
- SIP UA (Carol)

SIP / SDP

SIP may convey membership

Feels like a point-to-point call
Centralized Signaling: Endpoint

Feels like a point-to-point call

(Only) Carol’s UA is aware of the conference

SIP may convey membership

Decentralized Signaling: Mesh

All endpoints know about the conference

SIP conveys membership
Centralized Media: Bridge

Conference Bridge / MCU (SIP UA)

SIP UA (Ann)

SIP UA (Carol)

SIP UA (Dave)

Membership from RTCP

Feels like a point-to-point call

RTP Audio

A C+D D A+C

C A+D

Local mixing and source selection

Centralized Media: Endpoint

SIP UA (Ann)

SIP UA (Carol)

SIP UA (Dave)

Membership from RTCP

Feels like a point-to-point call

RTP Audio

A C+D D A+C

A+C D

Local mixing and source selection
Decentralized Media: Multi-Unicast

Membership from RTCP

SIP UA (Ann)

A

C

D

RTP Audio

SIP UA (Carol)

Local mixing and source selection

SIP UA (Dave)

Decentralized Media: Multicast

Membership from RTCP

SIP UA (Ann)

A

C

D

RTP Audio

SIP UA (Carol)

Local mixing and source selection

SIP UA (Dave)
Conference Creation

- Ad-hoc expansion of a SIP call
  - INVITE further participants
  - Re-configure media streams
  - Introduce a mixer if necessary (e.g. by SIP server)

- Advance reservation of a bridge / MCU
  - (Reservation itself out of scope)
  - Call in to conference URL
  - Call out from bridge to list of participants
  - Repeatedly (re-)configure media streams as needed

(Scheduling and Announcement with SAP)

Model Transition

- Conference starts off as a call
  - Endpoints can’t do mixing

- Conference grows larger
  - Than the mixing endpoint can deal with

- Conference bridge no longer needed

“Call Transfer” for all participants
  - INVITE and BYE, REFER
  - Re-direct (and re-configure) media streams
Example 1: Conference Bridge

- Use only basic SIP features
  - SIP URL for identification
  - point-to-point calls for control and media
- Conferencing: application of SIP in the bridge
  - may hide or expose media differences
    - transcoding vs. media re-negotiation
  - may hide or expose participants’ identities
- Make a conference “feel” like a phone call
- Works with SIP phones today!
Conference Setup

- Ann uses a web browser to set up the conference
- She creates / obtains a URL for the conference
  - to send to Carol and Dave
  - to put on a web page
- Bridge registers with SIP server using the URL
  - when the conference is supposed to start

Ann calls in (1st)

INVITE
sip:irr@example.com
Caps (A)

INVITE
sip:irr2001@example.com
irr2001@example.com

REGISTER
sip:irr2001@example.com

INVITE
sip:irr@example.com
Caps (A)

INVITE
200 OK
Caps (B)

ACK
100 Trying

Conference with Caps (A ∩ B)

Conference (A ∩ B)
Carol calls in (2nd)

Conf (A ∩ B)

INVITE
sip:irr@example.com
Caps (C)

Conference
with Caps (A ∩ B ∩ C)

100 Trying

200 OK
Caps (B ∩ C)

check caps!

re-INVITE
Caps (B ∩ C)

re-INVITE
Caps (B ∩ C)

200 OK
Caps (A)

ACK

ACK

Conference (A ∩ B ∩ C)

Conf (A ∩ B ∩ C)

Dave calls in (3rd)

INVITE
sip:irr@example.com
Caps (D)

Conference
with Caps (A ∩ B ∩ C ∩ D)

100 Trying

200 OK
Caps (A ∩ B ∩ C)

check caps: Nothing to do!

200 OK
Caps (A ∩ B ∩ C)

ACK

ACK

Conference (A ∩ B ∩ C ∩ D)
Leaving and Terminating

- Leave a conference: BYE
- Expel a participant: Bridge sends BYE
  - Invocation triggered by other participant?
  - Extensions needed + policies + …

- Terminate a conference:
  - Bridge BYEs all
  - (Bridge de-registers from SIP server)

Example 1: Conference Bridge

- PRO
  - Endpoints need not be aware of conference
    - (if media distribution is handled centrally as well)
  - Can be done with SIP today
  - Endpoints can leave at will
  - Simple!

- CON
  - Central entity required (find it, book it, access it, …)
  - Single point of failure
Example 2: Endpoint as “Mixer”

- Logically similar to centralized bridge
- Endpoint creates two calls and bridges locally
- Perfect solution for small ad-hoc conferences
- With decentralized media: processing power less an issue
- Implemented in SIP Phones today!
Example 2: Endpoint as “Mixer”

- **PRO**
  - Endpoints need not be aware of conference
    - (if media distribution is handled by mixing endpoint)
  - Can be done with SIP today
  - Simple!

- **CON**
  - Mixing endpoint cannot leave
    - or will terminate the signaling relationships
  - Mixing endpoint has to handle many streams (b/w)
  - Single point of failure

Example 3: Meshed Conference

[Diagram showing a meshed conference with three SIP UAs (Ann, Carol, Dave) connected via RTP audio and SIP/SDP signaling.]
Example 3: Meshed Conference

PRO
- No centralized server required
- No single point of failure
- Participants may leave at will

CON
- More sophisticated endpoints required
- Each endpoint has to handle multiple streams (b/w)
- Complex protocol

Not yet completely defined!
SIP and Conference Control

For conferences of limited size:
- Share conference state information
  - Membership, media, encryption keys
  - Other?
- Manage the course of the conference
  - Floor control, conference policies, …
- Use SIP for state, but not for management
- Should there be another control protocol?

SIP for State Synchronization?

- Media configuration handled by SIP anyway
- Membership is straightforward
  - Done for full-mesh conferences
- Use SUBSCRIBE/NOTIFY for other
  - Membership and other conference state
  - Not perfectly efficient
    - But there is currently not so much state
    - Need not scale to arbitrarily large conferences
  - Seek another solution only when really needed
SIP for Conference Management?

- Current perception: don’t do that!
- Instead: devise a conference control protocol when needed
  - Could be carried in SIP
  - Or as one of the media
- Idea of conference control around for years
- BUT: no real (commercial) interest yet

SIP and Multimedia

First of all:

SIP supports ANY media!

But:

Need the other protocols & applications

And:

Need a way to “signal” them
Media Protocols

- Audio
- Video
- Tones (DTMF etc.)
- Text chat
- Fax
- Pointers
- ...

DONE

MISSING

- Shared Whiteboard
  - LBL WB, ...
- Shared Text
  - UCL NTE, emacs, ...
- Application Sharing
  - ITU-T T.128
  - Sun VNC
- ...

Session Description Protocol (SDP)

- Has enabled SIP and streaming application
  - works fine for many cases
  - makes many implicit assumptions

- BUT: Designed for Session Announcements
  - rather than for interactive “negotiations”

- Many recent extensions
  - to better support SIP, MEGACO in the short-term
  - General solution being worked out
SDP Next Generation (SDPng)

- Being designed to address SDP's flaws...
  - Limited expressiveness
    - For individual media and their combination
    - Often only very basic media descriptions available
  - No real negotiation functionality
  - Limited extensibility (clumsy, hard to coordinate)
  - No semantics for media sessions (only implicit)

- Also: Avoid second system syndrome!
  - Simple, easy to parse, extensible, limited scope

SDPng Structure

- Definitions
  - "optional" may be "imported"
- Potential and Actual Configurations
  - SDP m= blocks refers to definitions
- Constraints
  - on configurations "optional"
- Session Attributes
  - SDP session attr's + stream semantics
**SDPng Status**

- Requirements agreed upon in MMUSIC
  - Also input from SIP, MEGACO
- Basic structure agreed upon
- XML-based syntax chosen
- Strawman proposal available
- Draft spec expected for 51st IETF
- Next steps: definitions (media, transport, …)

**Conclusion**

- For TODAY, we are ok!
  - Audio(visual) conference bridges
  - Small group ad-hoc conferencing
  - End points may but need not support conferences.
- For TOMORROW, there is a long way to go…
  - SIP conferencing support and SDPng
  - Conference control?
  - Media protocols
- And we NEED APPLICATIONS that use it…!