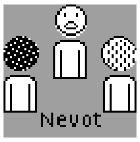
Dynamic Configuration of Conferencing Applications using Pattern-Matching Multicast



Henning Schulzrinne
GMD Fokus, Berlin
hgs@fokus.gmd.de

NOSSDAV'95 (Durham, NH) and IETF (Danvers, MA)

April, 1995

Overview

- motivation: diverse applications and control
- architecture: horizontal vs. vertical control
- protocol
- applications
- future work

What if Media Agents Could Talk to Each Other?

- “start-and-forget” ➠ continuous involvement
- audio follows video: stereo placement according to window location
- video follows audio: enlarge image of speaker
- audio enables video: send at higher rate when talking
- auxiliary applications: recorders, talk timers, …
- floor controller controls audio, video
- SNMP agent retrieves statistics, controls media agent

➠ must be easy to add, without access to source code
➠ independent of conference control mechanism

Architecture

control flow in multimedia conferences:

horizontal: conference control (CC), between participants

vertical: local, one participant, between applications

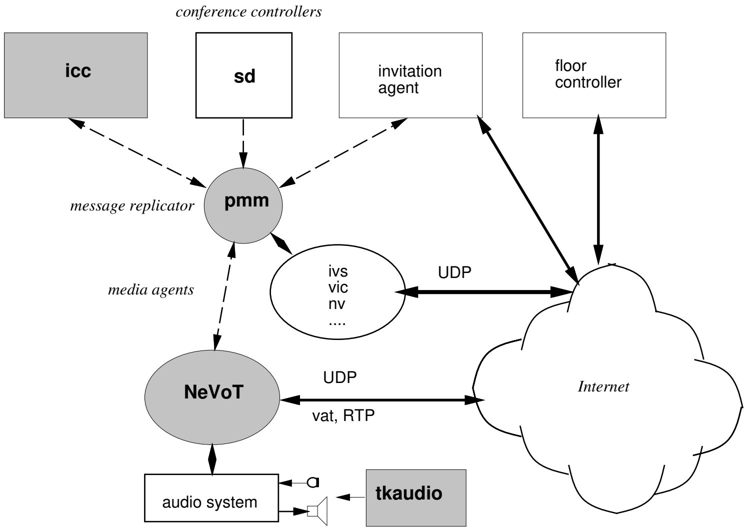
components:

controller: conference controller, floor controller, …: $\gg 1$!

media agents: audio, video, whiteboard, …

➠ reuse same media agents ↔ different CC protocols, styles

Architecture



Mechanism

- largely independent of lower layers:

  replicator process via TCP: pattern-matching filter,

  local UDP multicast: per-conference, per-user, per-media?

- media agents/controllers declare interest in receiving particular messages
- sender does not know receiver identity
- can be (carefully) extended beyond single host

related: vic/vat notifications

Protocol

- commands are idempotent
- currently, text-based: *object*, *operation*, *parameters*
- directly Tcl-interpretable (Tk `send` philosophy)
- works equally well for audio, video, whiteboard, …
- uses RTP/RTCP terminology, but not dependent on it

Example

```
M:   + session */audio/*
C:   + C/audio/3
C:   + C/audio/3/*
C:   create C/audio/3
M:   C/audio/3 created
C:   C/audio/3 ttl 128
C:   C/audio/3 media PCMU 1 8000 0.02
C:   C/audio/3 ...
C:   C/audio/3 open 224.2.0.1
M:   C/audio/3 opened
M:   C/audio/3 hello hgs@fokus.gmd.de
F:   C/audio/3/hgs@fokus.gmd.de send 1
M:   C/audio/3/hgs@fokus.gmd.de sending 1
M:   C/audio/3/hgs@fokus.gmd.de active
M:   C/audio/3/hgs@fokus.gmd.de !alive
M:   C/audio/3/hgs@fokus.gmd.de close
C:   C/audio/3 close
M:   C/audio/3 closed
M:   – C/audio/3
```

Bootstrapping Applications

Must handle

① single application per media session (video)

② several media sessions per application (NEVOT)

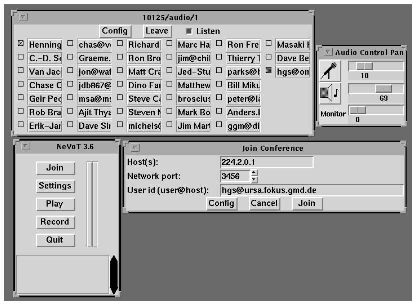
③ several media per application (audio/video)

replicator knows if `*/audio/*` registered; if not, create

multicast: send `create`; wait for reply; if none, start

Implementation

- NEVOT MBONE audio terminal
- controllers: interface to `sd`, telephone-style CC
- example: active badge: who's in the room?
- extended conference controllers: audio stereo location



Future Work

- asymmetric applications? (shared X-style)
- security ➠ encryption
- performance issues: replicator vs. local multicast
- late joiners: "who's out there"? periodic announcement?

Conclusion

- communication between multimedia components for
- coherent applications with minimal assumptions for each part
- ideas also applicable to other applications (event notification, resource sharing)