Multiparty Multimedia Session Control Working Group (mmusic)

Chairs:
Eve Schooler <schooler@isi.edu>
Abel Weinrib <abel@bellcore.com>

1,2 November 1993

Agenda

Meeting #1:

Overview of mmusic—*Abel Weinrib*
- goals
- framework

Managing shared session state—*Scott Shenker*
- policy specification
- agreement algorithm

Discussion

Meeting #2:

Session control above the agreement protocol—*Eve Schooler*
- usage of agreement framework

Discussion

Other topics
- an agreement protocol—*Bala Rajagopalan*
- relation to RTCP—*Steve Casner*
- session control work at BBN—*Julio Escobar*
- action items
  - accomplishing old ones
  - setting new ones

MMusic Overview

Motivation for work

Working group charter

- focus

Milestones (from last time)

8/93 Post an Internet Draft describing the Session Control requirements

11/93 Post an Internet Draft of the Session Control Protocol

3/94 Submit a revised Internet Draft based on implementation experience

6/94 Submit the Session Control protocolÔ77773340 to the IESG for considera
as an Experimental Protocol

Conference Styles

point-to-point call

impromptu hallway meeting

architecture design review

classroom

panel discussion

seminar

lecture

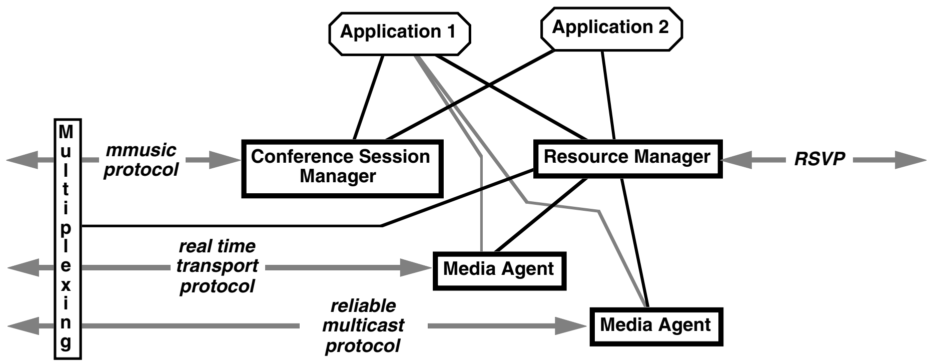
pay-per-view

TV broadcast

more coordination
more rigid
more overhead
smaller sessions
more static

less coordination
fewer policies
less overhead
larger sessions
more dynamic

Framework



State of a Session

1. **Session identifier**
2. **Member list** (M)
   - each member appears only once
   - at least one member (for tight controlled session)
3. **Session policies** (P)
   - voting
   - correctness
   - initiator of operations
4. **Other domain-specific state variables not "understood" by session manager**
   - media encodings and communication topology
   - security: encryption key distribution
   - floor control
   - common name
   - etc.

Session Control Protocol Requirements

Distributed session management

- establish, terminate, merge
  - once established, a session has a unique name (for example, to advertise the session in a directory service)
- schedule—negotiate the session in advance

Dynamic state management

- add member, delete member
- modify policies
- modify other state variables

Protocol Assumptions

One Session Manager per workstation port

- usually a single one at a well-known port
- a Session Manager may serve multiple users
- a Session Manager is not required to maintain state across *its* failures

There is an *initiator* of a session

- anyone may propose initiation of a session
- does the initiator have to be a member of the resulting session?

The initiator proposes the initial session state

- the proposal is forwarded to all potential members
- each proposed member knows up-front what it is agreeing to
- the voting policy defines who must agree for session establishment to succeed

More Protocol Assumptions

Once established, a member may propose changes to the session state

- only members may make such proposals
  - a non-member that wants to join a session asks a member to propose the required change
- voting policy dictates whether proposal is approved

While a proposal is outstanding, the proposer coordinates the outcome

- once the outcome is determined, the proposer reverts to having no special role

A member may always leave the session without negotiation *(as a last resort)*

Other Session Protocol Messages

Useful messages that are not part of Agreement Protocol:

Query

- sent by a session manager to request the session state from another session manager
- the receiving session manager responds with a status message

Status

- sent by a session manager to share the state of the session
- no reply is required

Session Protocol	
Agreement	**State Sharing**

Underlying Transport Mechanism

Assume reliable but asynchronous message delivery

Internet realities

- packet loss
- delay variations

Communication building blocks

- UDP vs. TCP
- unreliable vs. reliable “datagrams”
- point-to-point vs. multipoint
  - CCP reliable multipoint
- causal/atomic multicast
  - ISIS group services

Adaptability to delay variations

- adapt peer-to-peer timeouts to dynamic network conditions

Outstanding Issues (from last time)

Mobility at session establishment and during the session

Distributed system issues

- keep alive messages using Announce
- state synchronization (instead of strict correctness)
- join by contacting a *doorman* or *doormen*
- replace terminate session by using delete all members
- error handling
- serializability conflicts and fault tolerance
- (new) correct handling of timeouts for fault recovery

Transport infrastructure

- UDP, TCP, Sun RPC, VMTP, TP4, ISIS, etc.?

Schedule and preschedule

Floor control

Richness of policies

Other uses?

Action Items (from last time)

Documents

- requirements
  - framework
  - terminology
  - conversation styles
  - protocol requirements
- specification