Enhancements to Collaborative Media Streaming with IETF Protocols

Verena Kahmann

Institute of Operating Systems and Computer Networks
Technical University of Braunschweig

Dagstuhl Seminar - Content Distribution Infrastructures, 2004

Outline

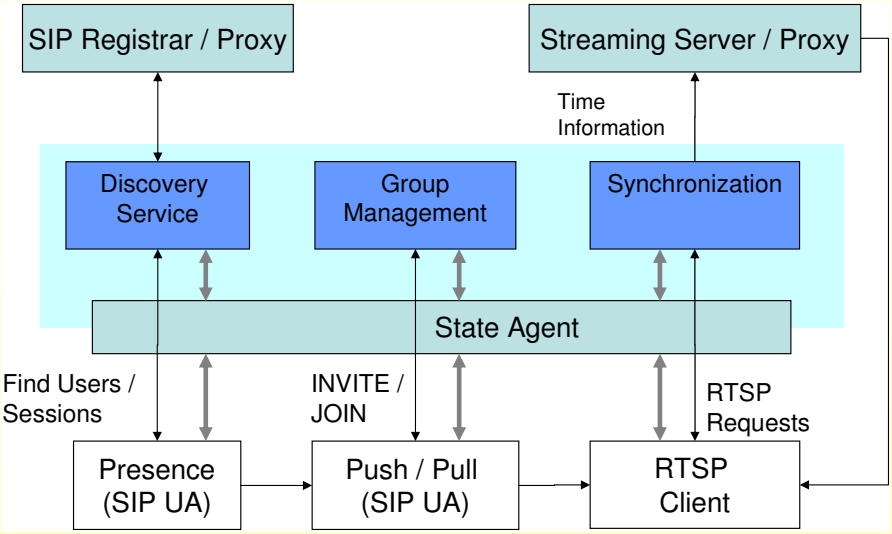
Scenarios

- Home Networks
  - Copy / move streaming session to device (e.g. other display)
  - Invite visitor to streaming session
    → Invitation
    → Different Tracks / Qualities
    → Synchronization
- Learning Environments
  - Users watch streamed learning session collaboratively
  - Each user may jump individually to interesting chapters
    → Group communication
- Spontaneous Meetings
  - Copy streaming session to user
    → Discovery
    → Mobility

General Requirements

- Use IETF Standard Protocols (i.e. RTSP, SIP, RTP, . . .)
- Streaming
  - Individual stream control
  - No upload of media data
- Collaboration
  - Invite user → Push to device
  - Join session → Pull to device
  - Synchronize to group view
- Session sharing vs. individual control
  - Cannot use Multicast without adaptation
  - Cannot use off-the-shelf applications
  - Adaptation of signaling environment

Overview



Association Service Responsibilities

- Processing RTSP/SIP requests
- Group management
  - Initialize association on RTSP SETUP
  - Bind peer user on SIP INVITE or JOIN
  - Update timeline on RTSP PLAY
- Synchronization to group timeline
  - Calculate position according to timeline
  - Submit position to RTSP proxy in "Range" header

Synchronization

- Save start time and position of peers
- For joining peers, calculate position
- Sometimes, strict synchronization needed!
- Problems:
  - Variation in networking delay
  - Different play-out delays
- Possible solutions:
  - Delay measurements for initial synchronization
  - Synchronization Protocol
  - Synchronize at certain points / events

Group Management

Group streaming (e.g. for learning environments, split audio/video)
vs. Individual streaming (e.g. for spontaneous meetings)
$\rightarrow$ SIP does not include group communication

- Set tag in invitation (e.g. session description)
  - Position update for group
  - Position update individually
  - Position update on request
- Group update: Association Service issues RTSP PLAY with new position
- Individual update: Association Service forms a new subgroup
- On-request update: Users send Re-INVITE before position change

Requirements

- Need to discover
  - Devices (passive)
  - Users (active)
  - Content
  - User-related content (for pull – give me the URL of your stream)
- Requirements
  - Find only relevant devices
  - Content search filters
  - Privacy vs. Service access
  - User-friendliness vs. easy implementation
- One-for-all solution hard to find

Discovery Framework

- Several discovery protocols available
  - SIP Instant Messaging / Presence for user discovery (registration with SIP Registrar)
  - SLP for device discovery (also initial user discovery?)
  - Content search not standardized yet (approaches e.g. in UPnP)
- For user-related content, use SIP Event Notification
  - Peers can subscribe to events defined in an event package
  - Peers distribute notification of state changes
  - State changes quite small here (which movie/tracks are started)

Mobility Provisions

- Flexible Architecture: Different Servers and Proxies usable
- Mobility:
  - Sessions can survive several TCP connections
  - Media transport should survive handover
  - Inherent IETF "Session Layer" ability, precondition: network mobility available
- Up to now: one Association Service → point of failure
  - Distribute Association State on Clients
  - State needed:
    - URL to content
    - Position in timeline
    - Push/pull partner, Call-ID
  - SIP event notification used

Splitting and Merging Networks

- Splitting (Signaling connection to peers lost):
  - Media still arriving
  - Peers working locally (saving jumps)
- Joining new network (No connection to old peer network):
  - Open new call if new collaboration desired
- Merging (Signaling connection to peers rebuilt):
  - Redistribute state (e.g. SIP Event Notification)
  - Form subgroup if one of peers changed position
  - Possibly leave call if any partner left stream

SIP Event Package

- Stream Events
  - Stream started (time)
  - Position change (time, position)
  - Stream ended (time)
- Group Events
  - Joined group
  - Left group
  - Form subgroup
- Mobility Events
  - Lost signaling connection
  - New network found
  - Rebuilt signaling connection

Notification and Actions

- State Changes
  - Group members
  - Stream URL, Tracks
  - Start times, start positions of (sub-)groups
- Notifications sent on State Changes
- Association Service serving as State Agent
  - Aggregation of notifications
  - Acting on behalf of thin clients

Open Topics

- RTSP and SIP: two different protocols
  - Different addressing
  - Difference session / call
  - Transfer state from one client application to the other
- Synchronization: how strict
  - Optimize for main scenarios?
  - Find a generic solution?
- Application weaknesses
  - Compromise between standard and implementation
  - Often restrictions on users
  - Display and inter-stream sync sometimes annoying

Conclusion

- SIP Event Notification enhances Collaborative Streaming
  - Flexibility with loose grouping concept (no conferencing necessary)
  - User-friendly discovery support
  - Mobility support
- Some issues remain to be solved
  - Strict synchronization
  - Experiments with proxy caches and multicast
  - Experiments with SIP Event Notification