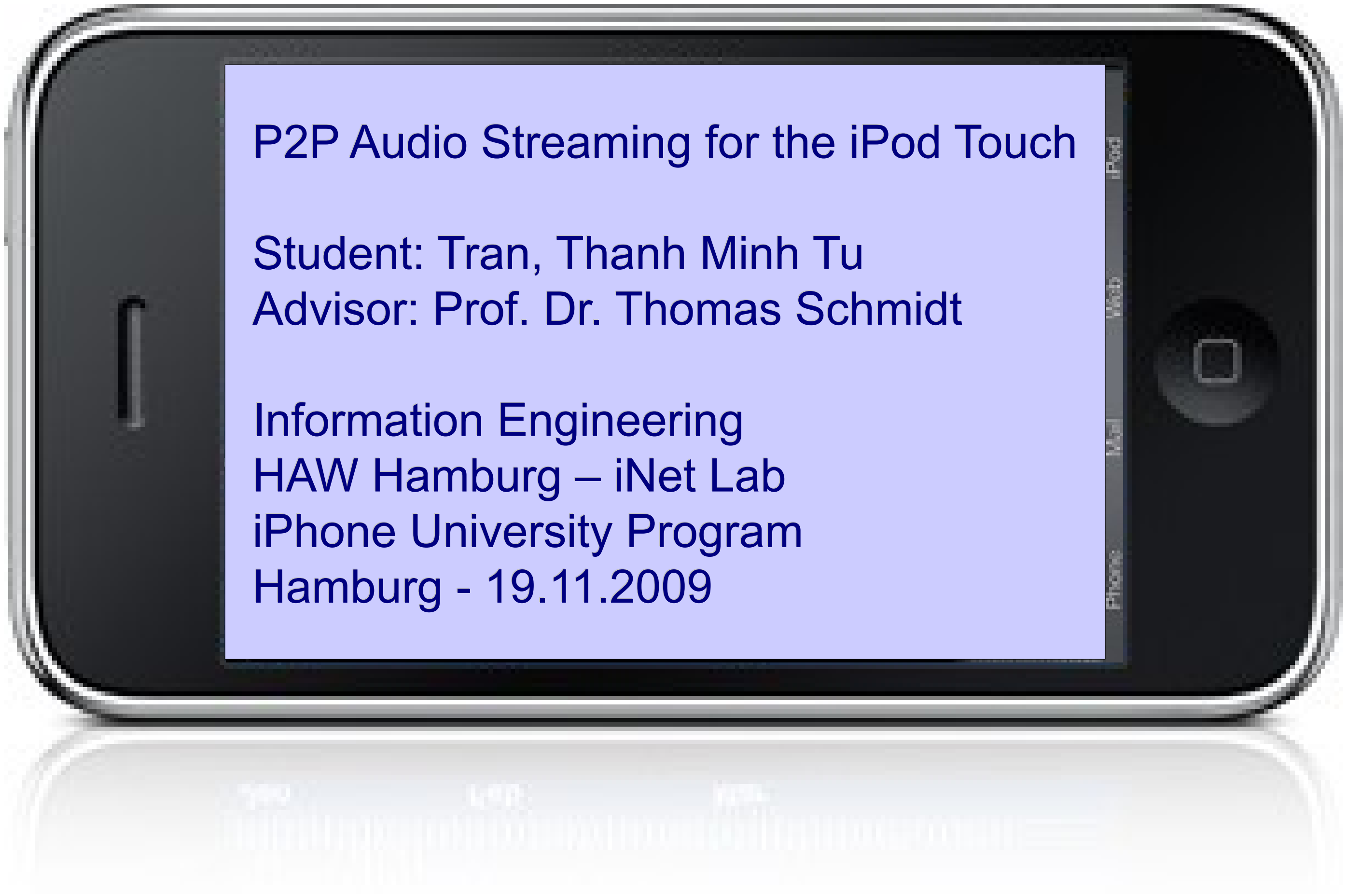
iNET

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences



P2P Audio Streaming for the iPod Touch

Student: Tran, Thanh Minh Tu
Advisor: Prof. Dr. Thomas Schmidt

Information Engineering
HAW Hamburg – iNet Lab
iPhone University Program
Hamburg - 19.11.2009

Overview

- Introduction
- Overlay P2P Network
- Application Layer Multicast (ALM)
- Audio Processing on the iPhone
- P2P Audio Streaming for the iPod Touch (PAS4i)
- Discussion

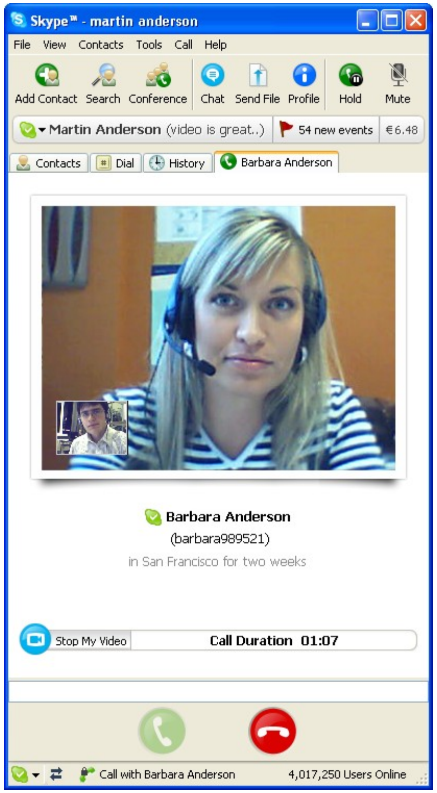
Introduction

- The P2P file sharing systems
- The P2P media streaming systems
- Our story
- The PAS4i application

The P2P file sharing systems

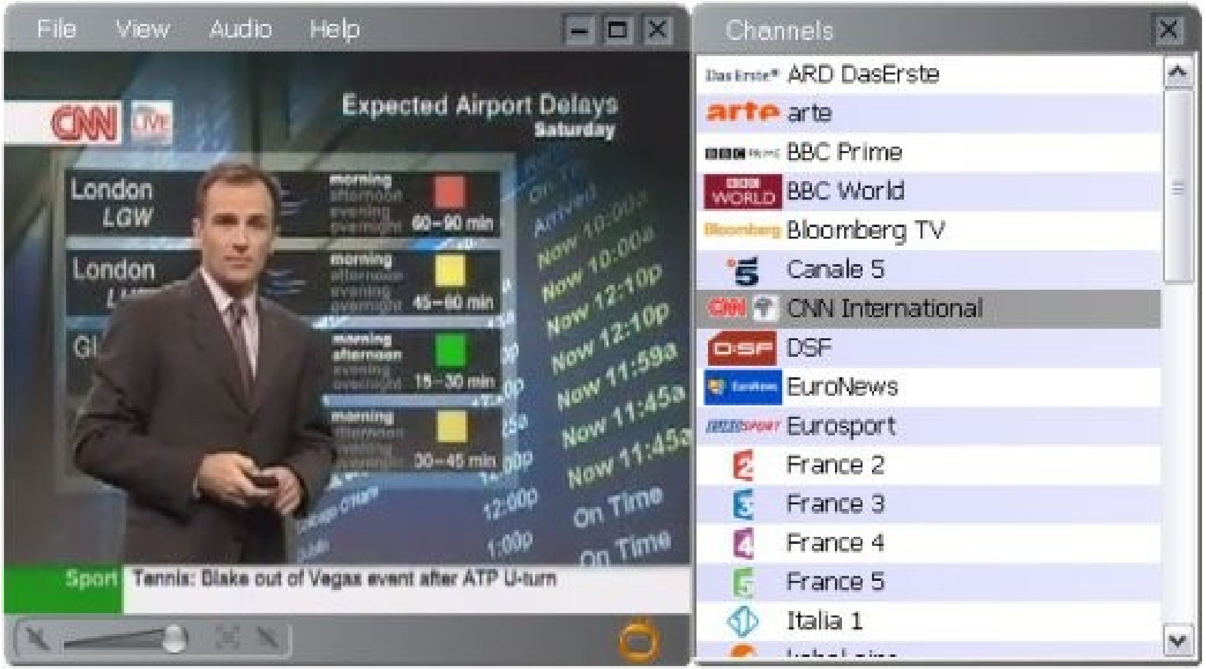
- Napster (1999)
- Gnutella
- BitTorrent

The P2P media streaming systems


Skype™ - martin anderson
File View Contacts Tools Call Help
Add Contact Search Conference Chat Send File Profile Hold Mute
Martin Anderson (video is great..)
54 new events
€6.48
Contacts Dial History Barbara Anderson
Barbara Anderson
(barbara989521)
in San Francisco for two weeks
Stop My Video
Call Duration 01:07
Call with Barbara Anderson
4,017,250 Users Online

The P2P media streaming systems

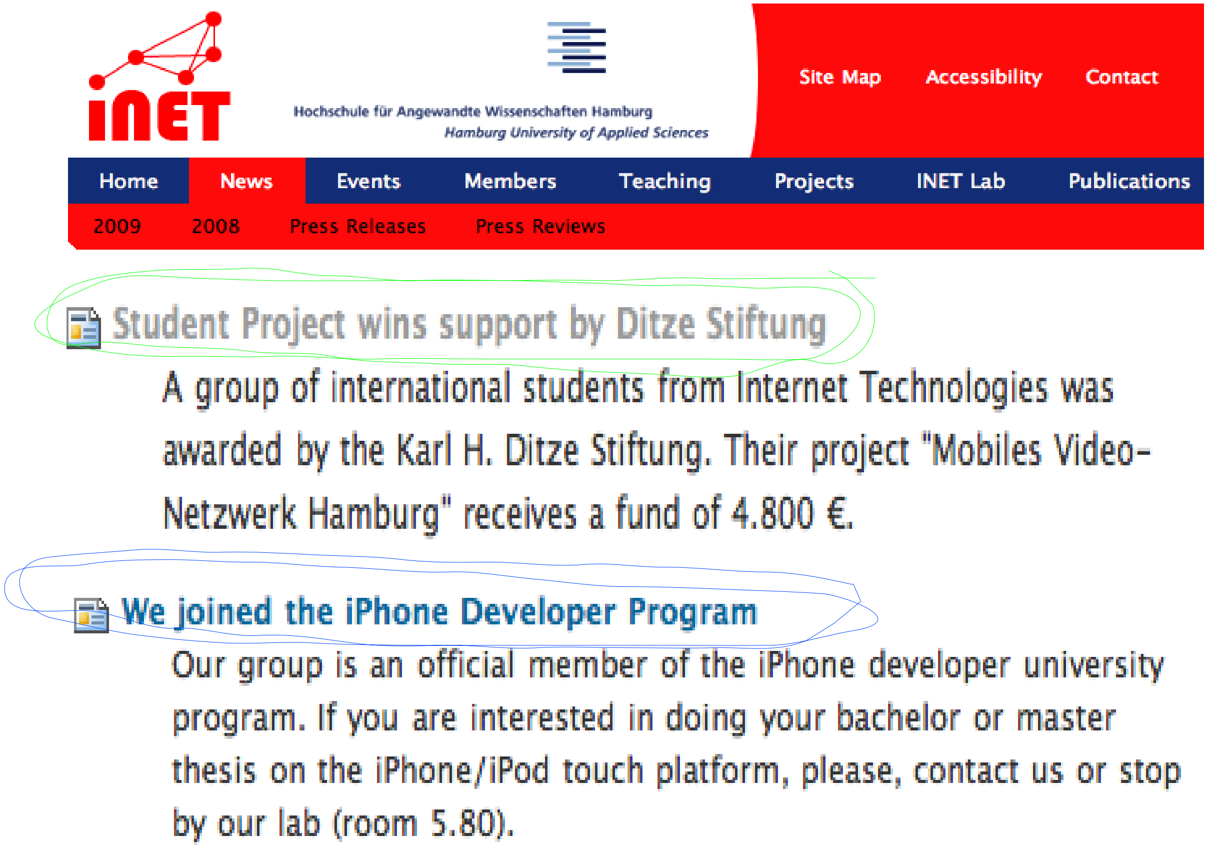
ZATTOO



Channels

- ARD DasErste
- arte
- BBC Prime
- BBC World
- Bloomberg TV
- Canale 5
- CNN International
- DSF
- EuroNews
- Eurosport
- France 2
- France 3
- France 4
- France 5
- Italia 1

Our story

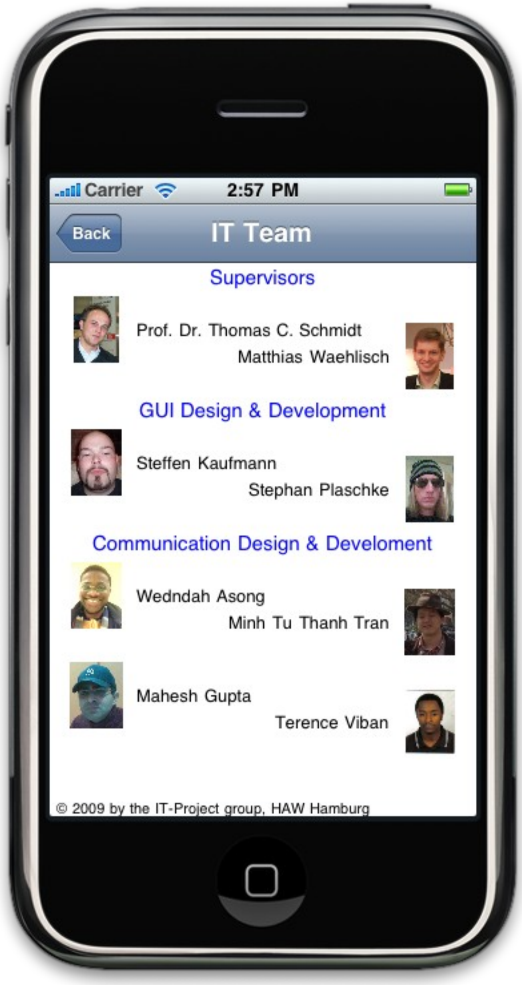


Student Project wins support by Ditze Stiftung

A group of international students from Internet Technologies was awarded by the Karl H. Ditze Stiftung. Their project "Mobiles Video-Netzwerk Hamburg" receives a fund of 4.800 €.

We joined the iPhone Developer Program

Our group is an official member of the iPhone developer university program. If you are interested in doing your bachelor or master thesis on the iPhone/iPod touch platform, please, contact us or stop by our lab (room 5.80).



Supervisors

Prof. Dr. Thomas C. Schmidt
Matthias Waehlisch

GUI Design & Development

Steffen Kaufmann
Stephan Plaschke

Communication Design & Develoment

Wedndah Asong
Minh Tu Thanh Tran

Mahesh Gupta
Terence Viban

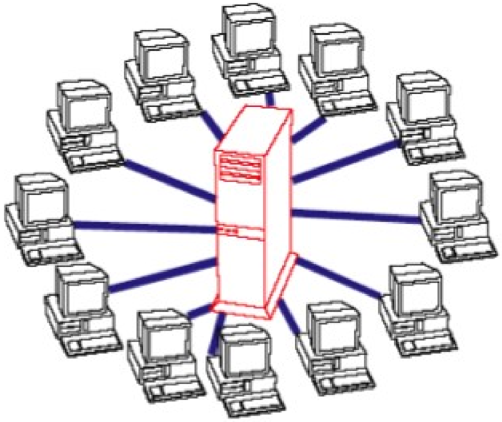
© 2009 by the IT-Project group, HAW Hamburg

The PAS4i application

- PAS4i: “P2P Audio Streaming for the iPod Touch”

Canada
United States
Europe
Jordan
China
Egypt
Bahrain
India
Philippines
Guam
Paraguay
Uruguay
Argentina
South Africa

Client-Server model



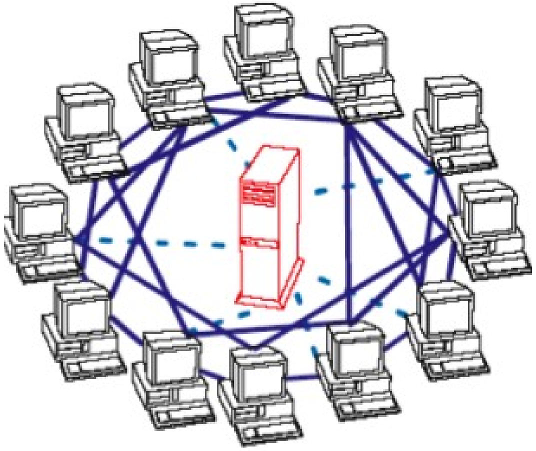
Client-Server

For large scale system:
High deployment cost!!!

Overlay P2P Network

- Unstructured P2P
  - Centralized P2P
  - Pure P2P
  - Hybrid P2P
- Structured P2P
  - DHT-based: Chord, Pastry, CAN etc.

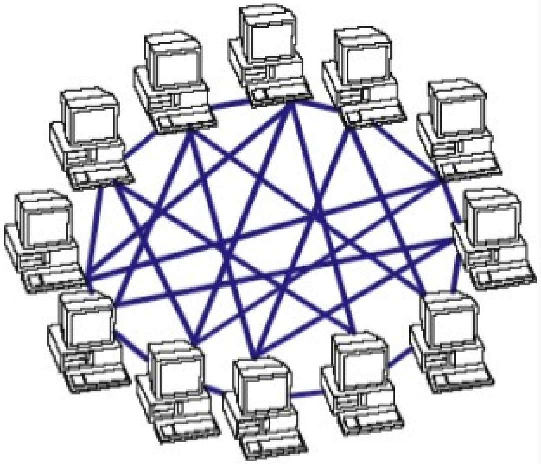
Unstructured P2P



Centralized P2P

Central entity (indexing) provides the service

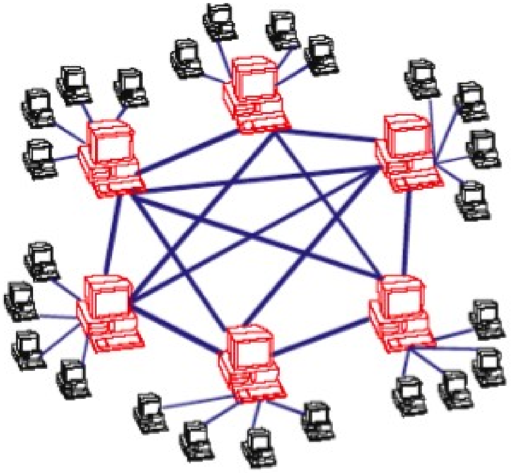
Central server:
single point of failure



Pure P2P

No central entity
Any peer can be removed

Flooding:
communication overhead

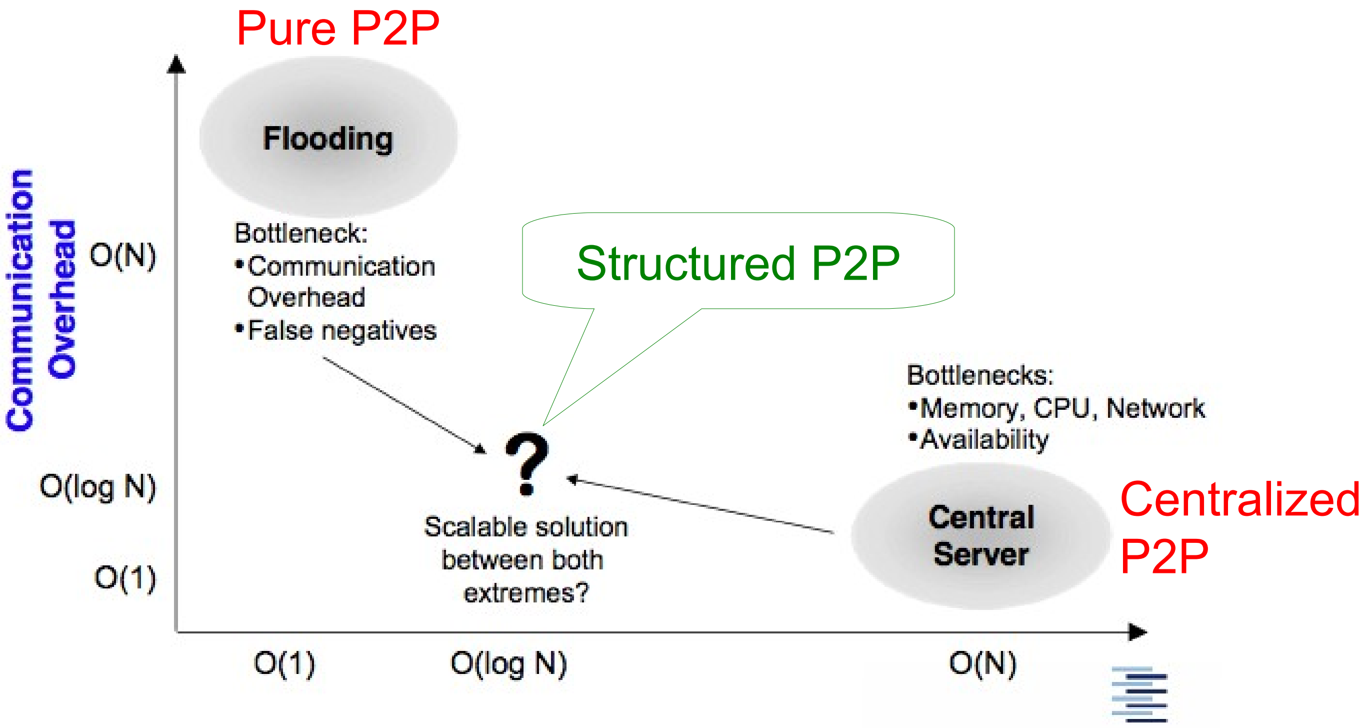


Hybrid P2P

Dynamic central entities
Any peer can be removed

Pros and Cons solution of both Centralized & Pure P2P

Unstructured P2P



Pure P2P

Flooding

Bottleneck:
- Communication Overhead
- False negatives

Structured P2P

?

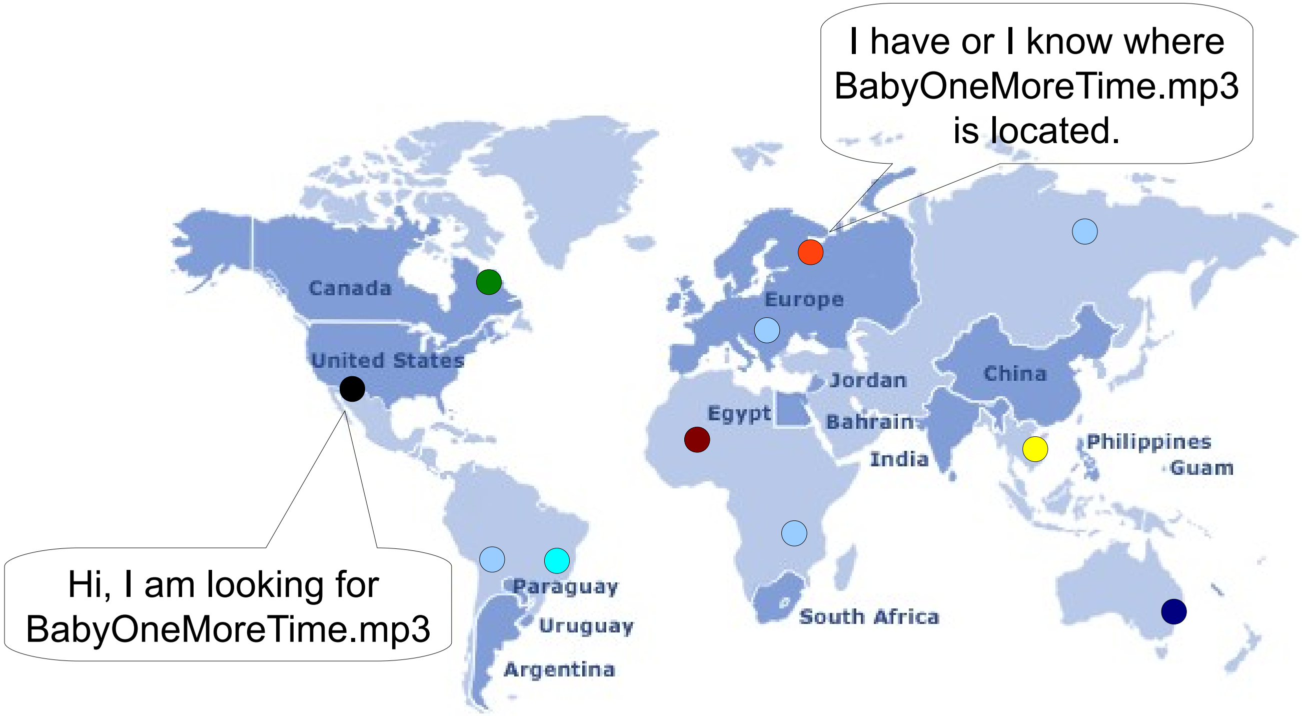
Scalable solution between both extremes?

Bottlenecks:
- Memory, CPU, Network
- Availability

Central Server

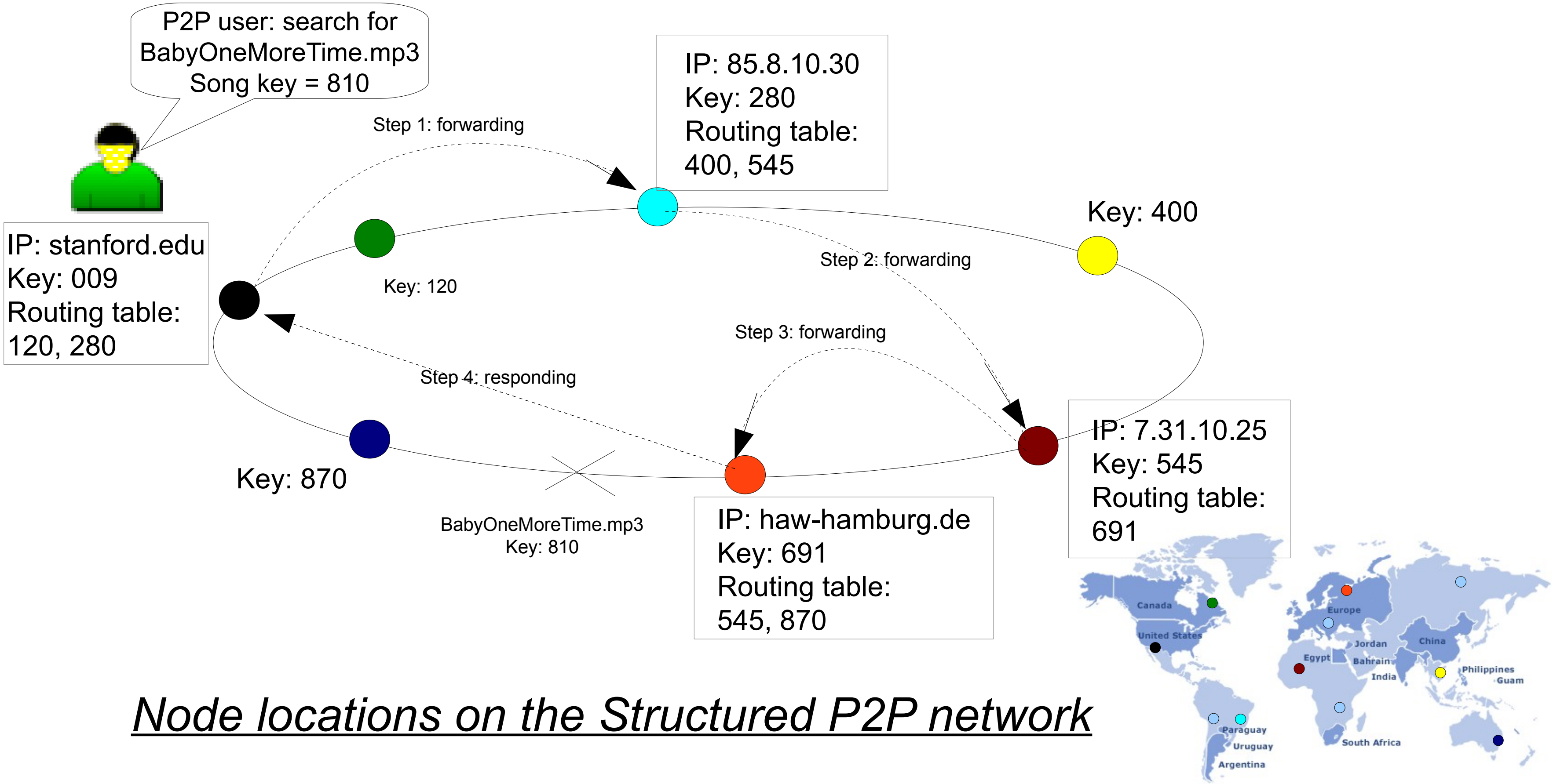
Centralized P2P

Structured DHT-based P2P



Node locations on the underlay IP network

Structured DHT-based P2P



P2P user: search for BabyOneMoreTime.mp3 Song key = 810

IP: stanford.edu
Key: 009
Routing table:
120, 280

IP: 85.8.10.30
Key: 280
Routing table:
400, 545

IP: 7.31.10.25
Key: 545
Routing table:
691

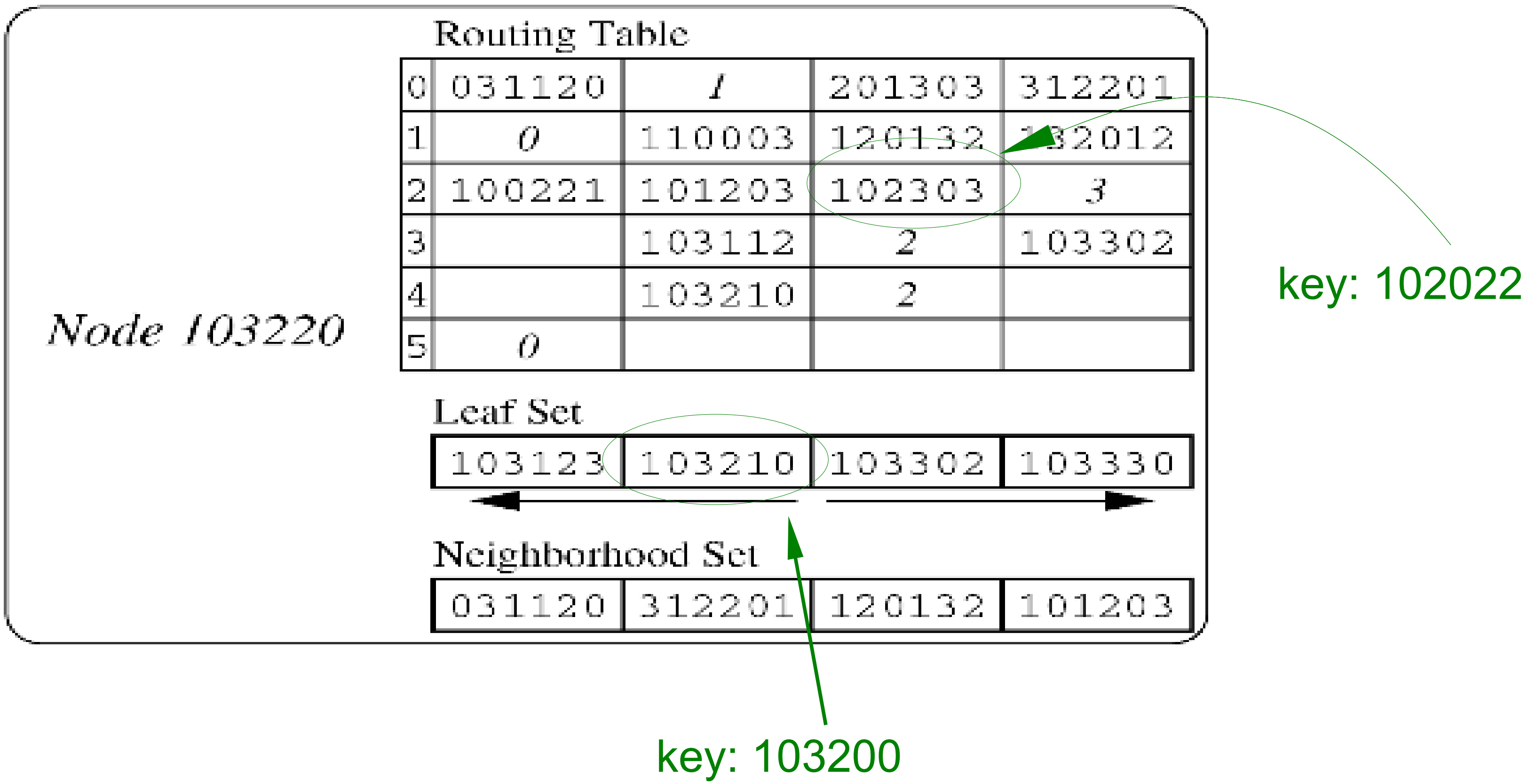
IP: haw-hamburg.de
Key: 691
Routing table:
545, 870

BabyOneMoreTime.mp3
Key: 810

Node locations on the Structured P2P network

Structured DHT-based P2P – Pastry approach

- Routing information of a Pastry node



Node 103220

Routing Table

0	031120	*1*	201303	312201
1	*0*	110003	120132	132012
2	100221	101203	102303	*3*
3		103112	*2*	103302
4		103210	*2*	
5	*0*			

Leaf Set

103123	103210	103302	103330

Neighborhood Set

031120	312201	120132	101203

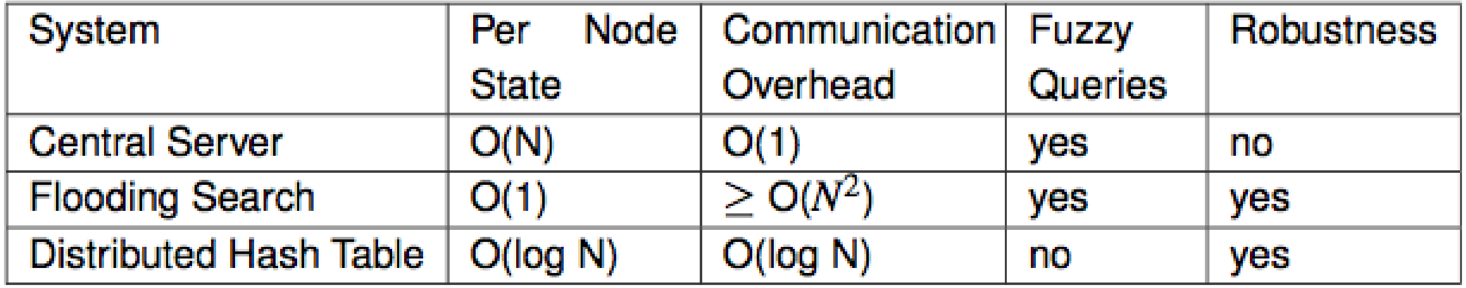
key: 102022

key: 103200

Structured DHT-based P2P – Pastry approach

- Pastry API
  - pastryInit(Credentials, Application)
  - route(msg, key)
  - send(msg, IP-addr)
  - deliver(msg, key)
  - forward(msg, key, nextId)
  - newLeafs(leafSet)

Structured DHT-based P2P



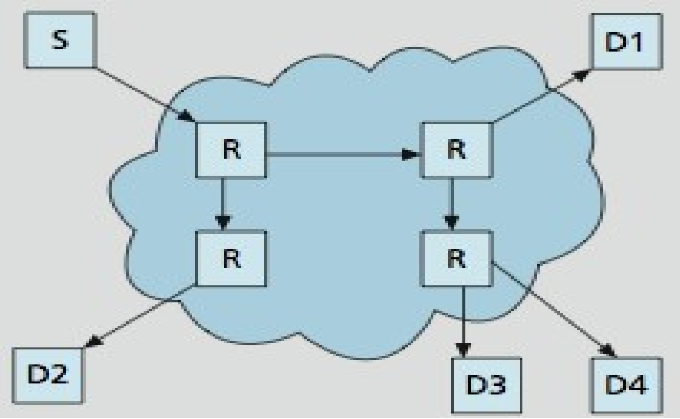
System	Per Node State	Communication Overhead	Fuzzy Queries	Robustness
Central Server	O(N)	O(1)	yes	no
Flooding Search	O(1)	$\geq$ O(N^2)	yes	yes
Distributed Hash Table	O(log N)	O(log N)	no	yes

Table 1: Comparison of central server, flooding search and distributed indexing (source: [23])

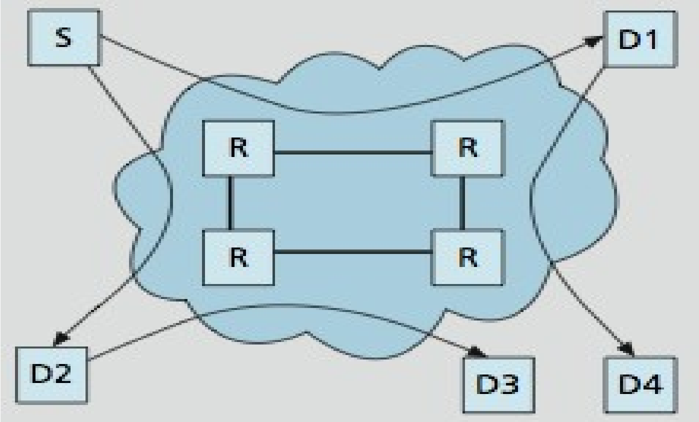
Application Layer Multicast (ALM)

- IP Multicast vs. ALM
- ALM characteristics
- ALM media Live streaming topologies
  - Tree-first vs. Mesh-first
  - Scribe – a Tree-first approach

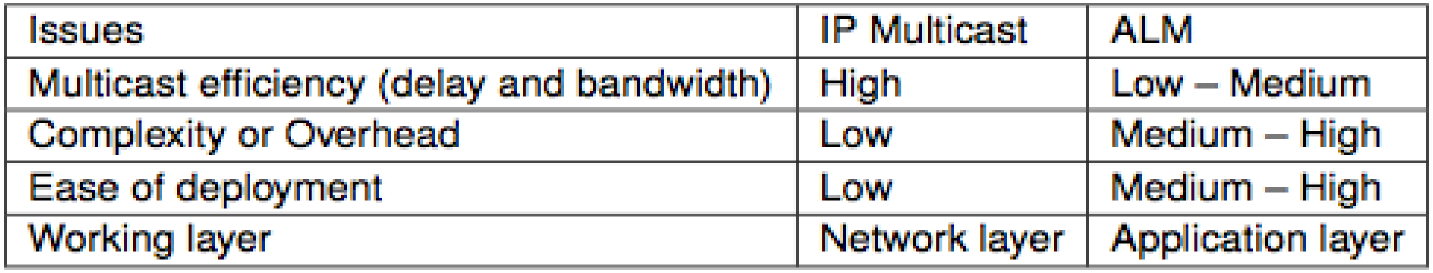
IP Multicast vs. ALM



IP Multicast



ALM



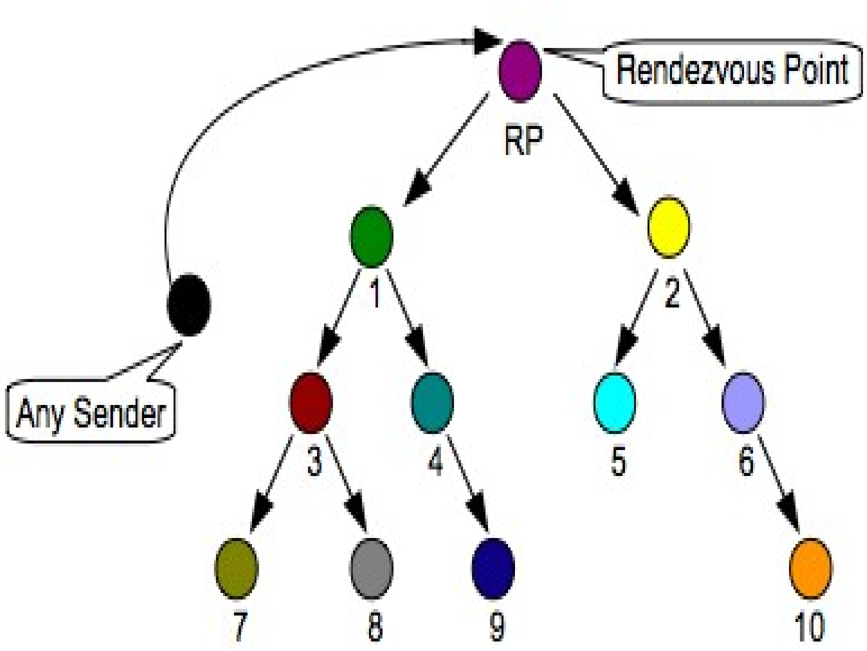
Issues	IP Multicast	ALM
Multicast efficiency (delay and bandwidth)	High	Low – Medium
Complexity or Overhead	Low	Medium – High
Ease of deployment	Low	Medium – High
Working layer	Network layer	Application layer

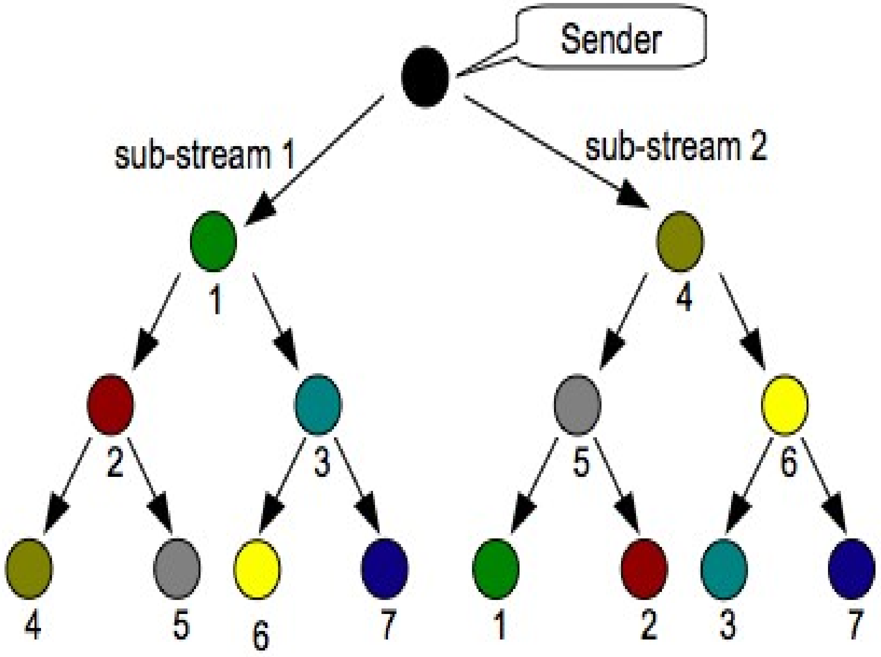
Table 3: Conceptual comparison of IP multicast and ALM (source [30])

ALM characteristics

- Deployment level (i.e., Proxy-based or End-system ALM)
- Application domain (e.g., number of user, latency constraint, network resource etc.)
- Multicast management approach
- Routing mechanism

Streaming topologies – Tree-first





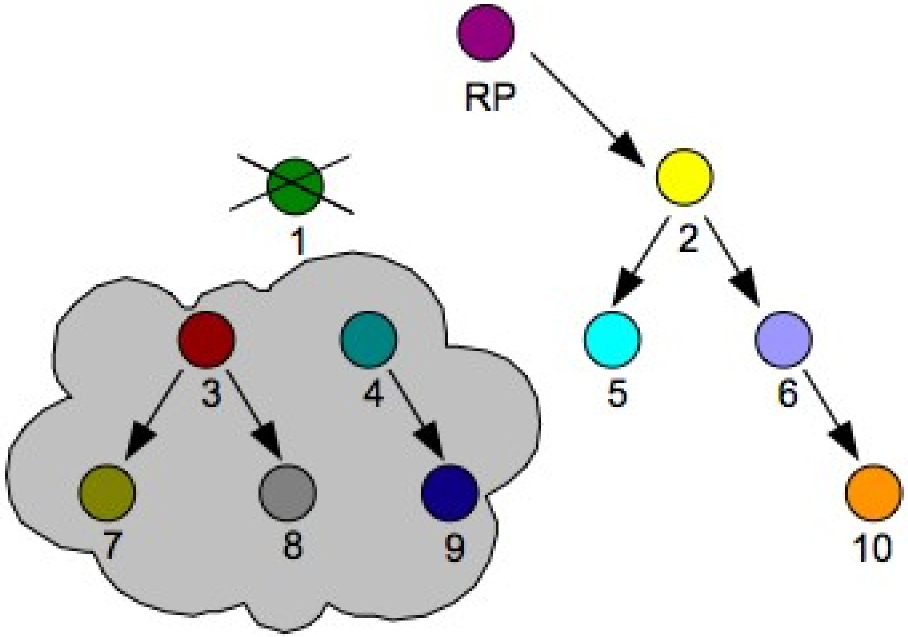
Single-tree approach

No upload bandwidth for leaf nodes

Multi-tree approach

All nodes contribute the upload bandwidth

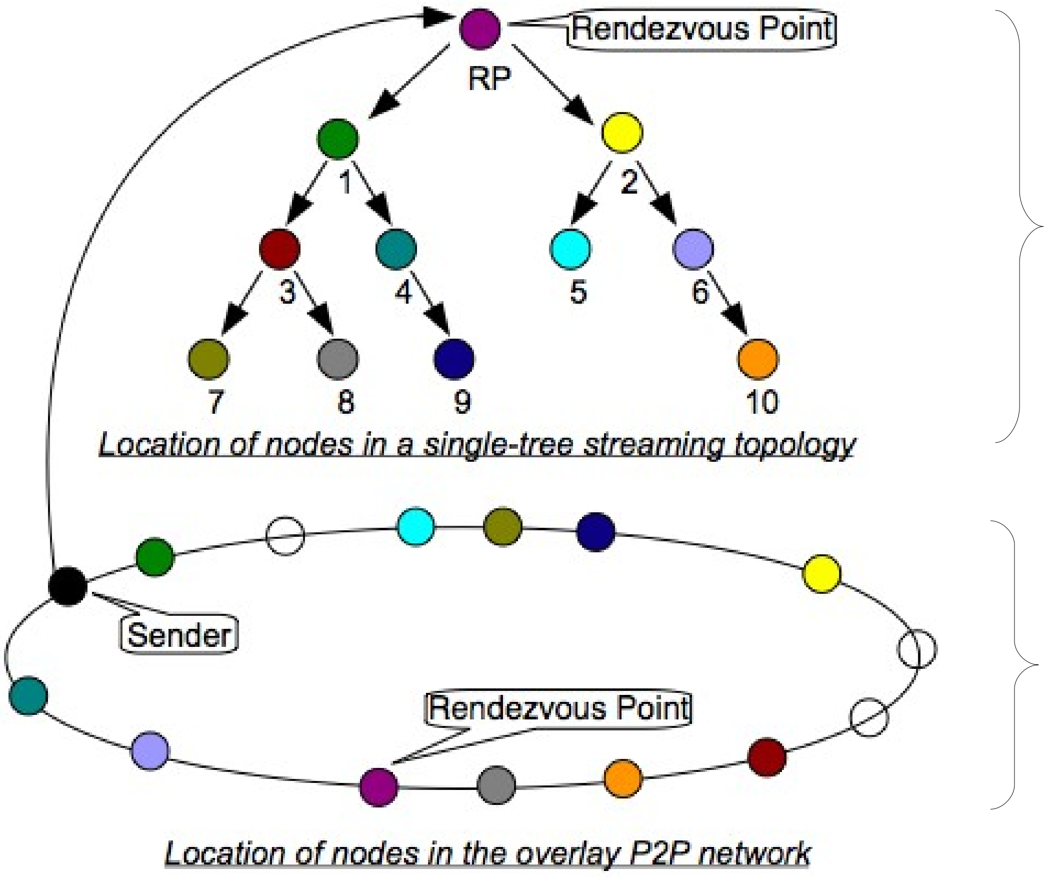
Streaming topologies – Tree-first



Peer Churn

Robustness in Tree-based approach?

Scribe – a Tree-based approach



Location of nodes in a single-tree streaming topology

Location of nodes in the overlay P2P network

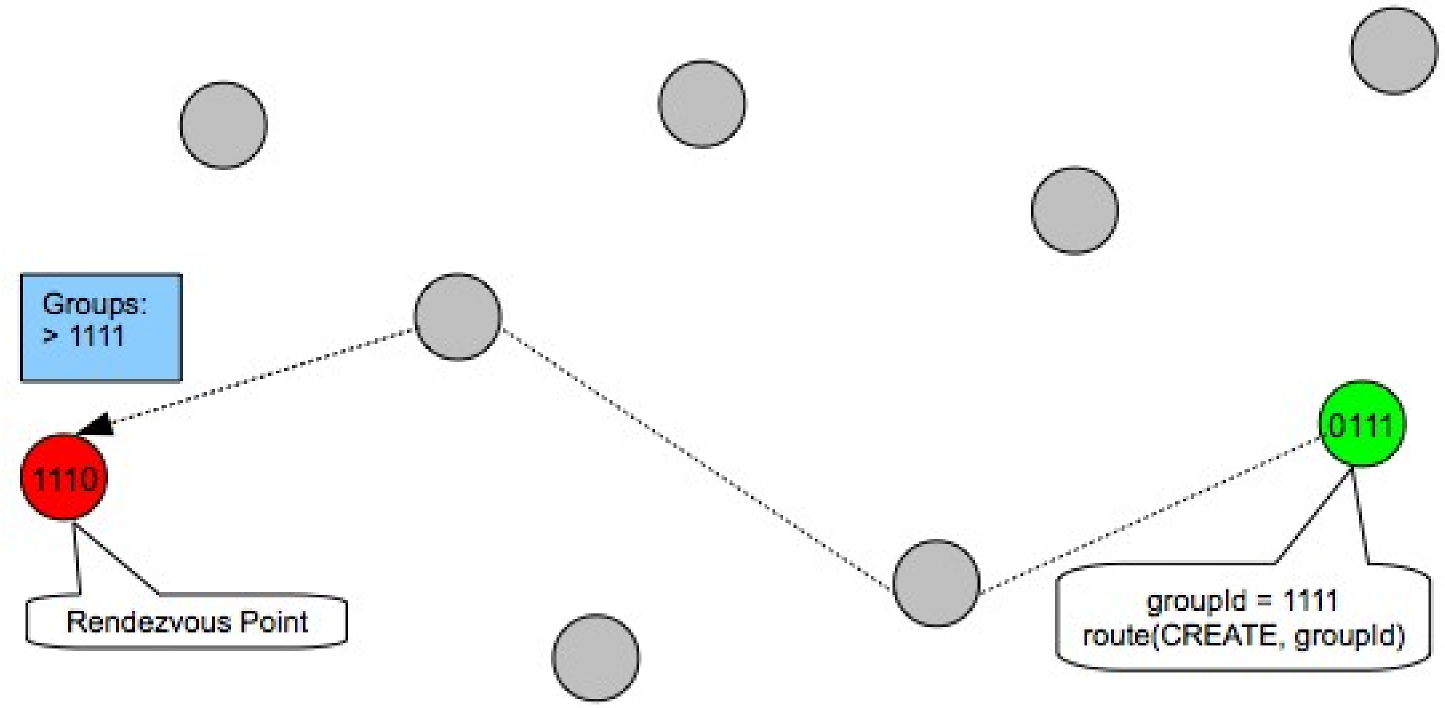
Streaming tree: Scribe
Built on the Pastry Overlay

Pastry Overlay Network

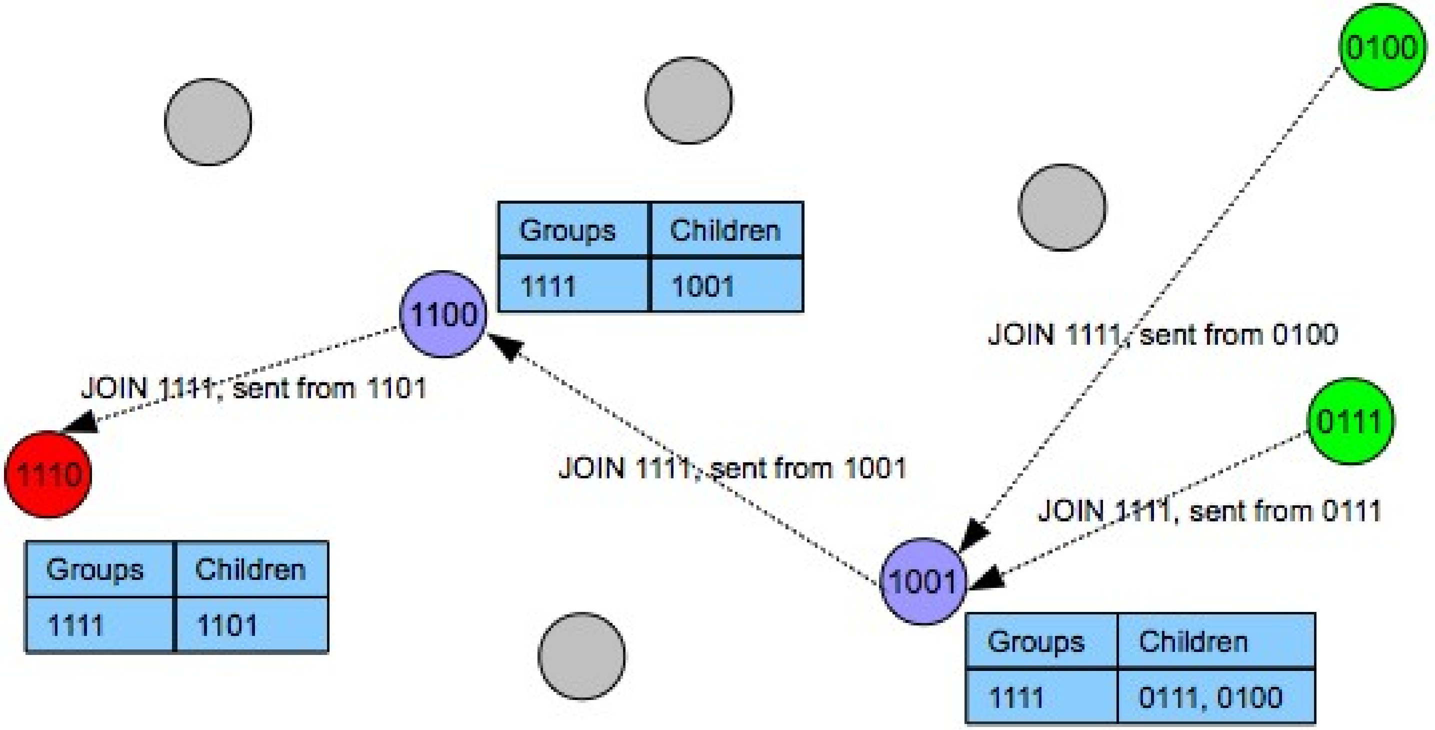
Scribe – a Tree-based approach

- Scribe API
  - create(credentials, groupId)
  - join(credentials, groupId, msgHandler)
  - leave(credentials, groupId)
  - multicast(credentials, groupId, msg)

Scribe – group creation



Scribe – group joining



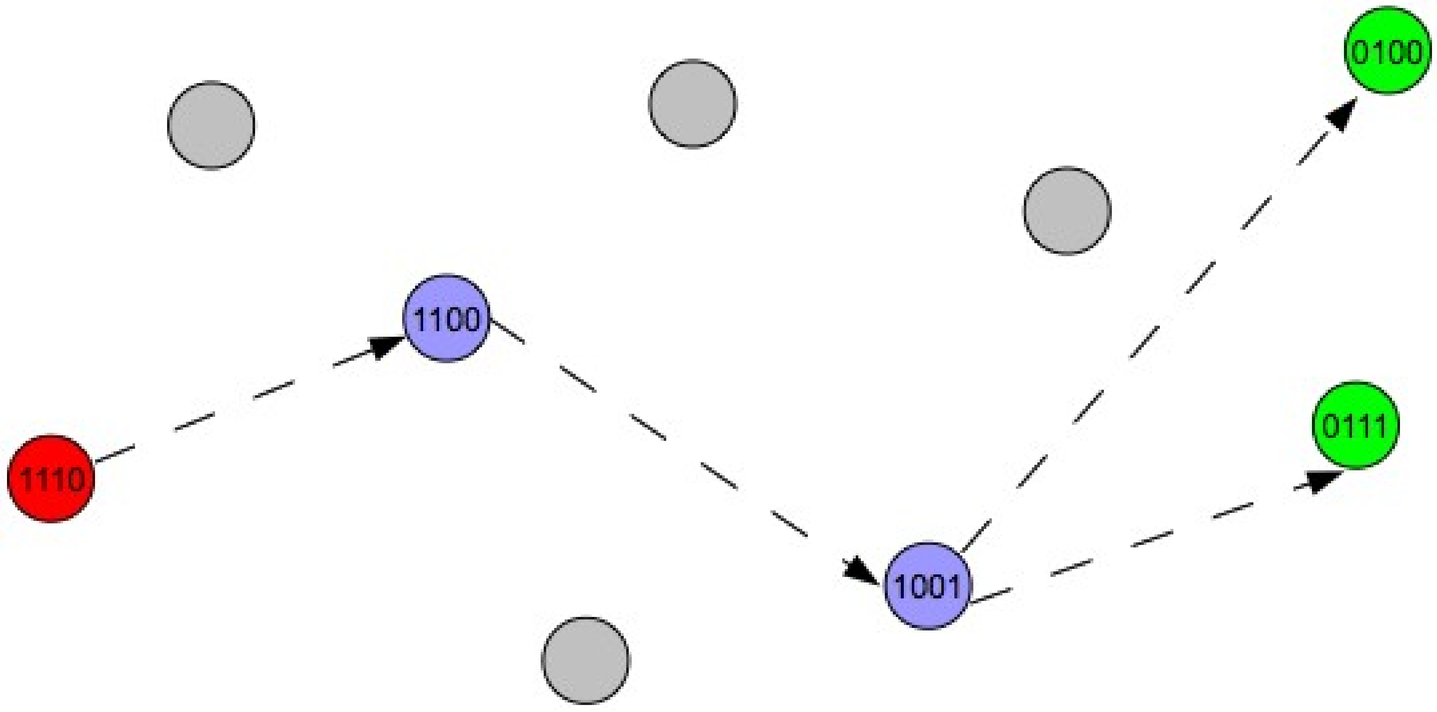
Groups	Children
1111	1001

Groups	Children
1111	1101

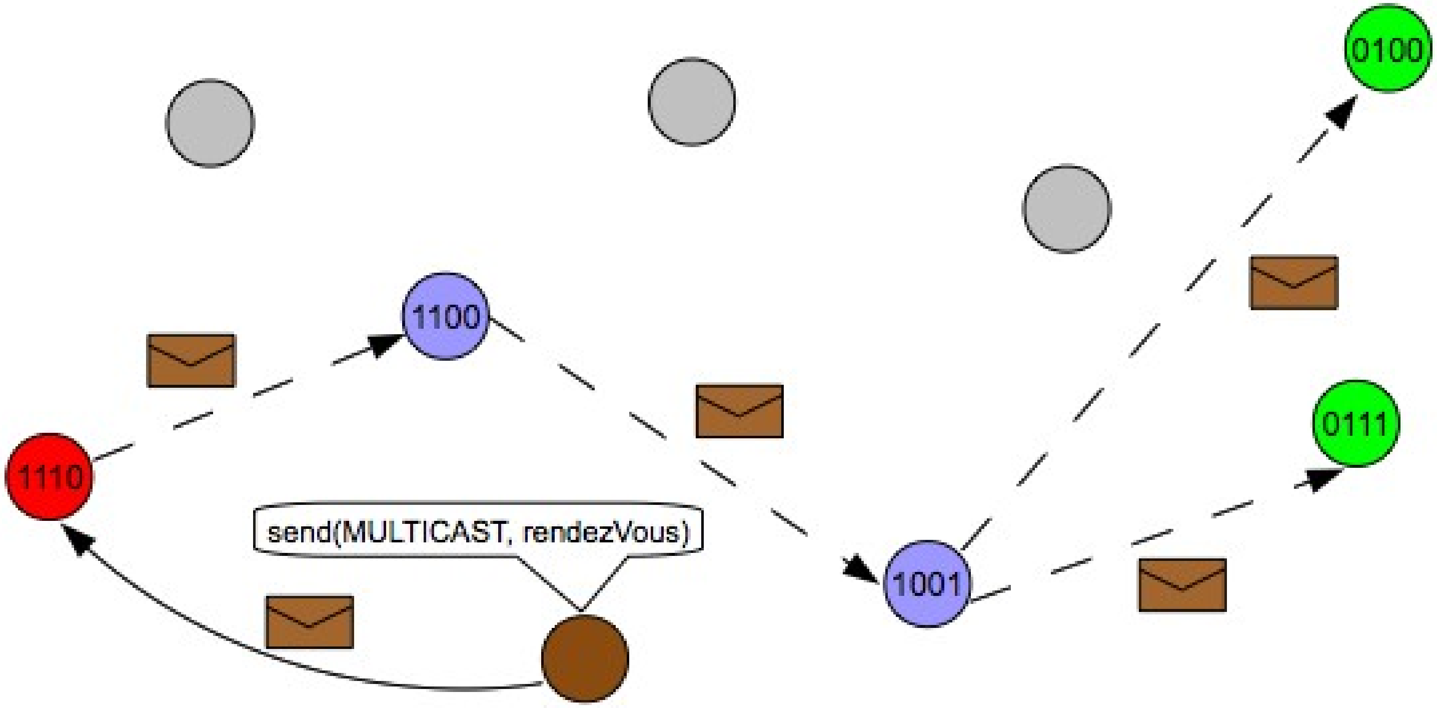
Groups	Children
1111	0111, 0100

0100
1100
JOIN 1111, sent from 0100
JOIN 1111, sent from 1101
0111
1110
JOIN 1111, sent from 1001
JOIN 1111, sent from 0111
1001

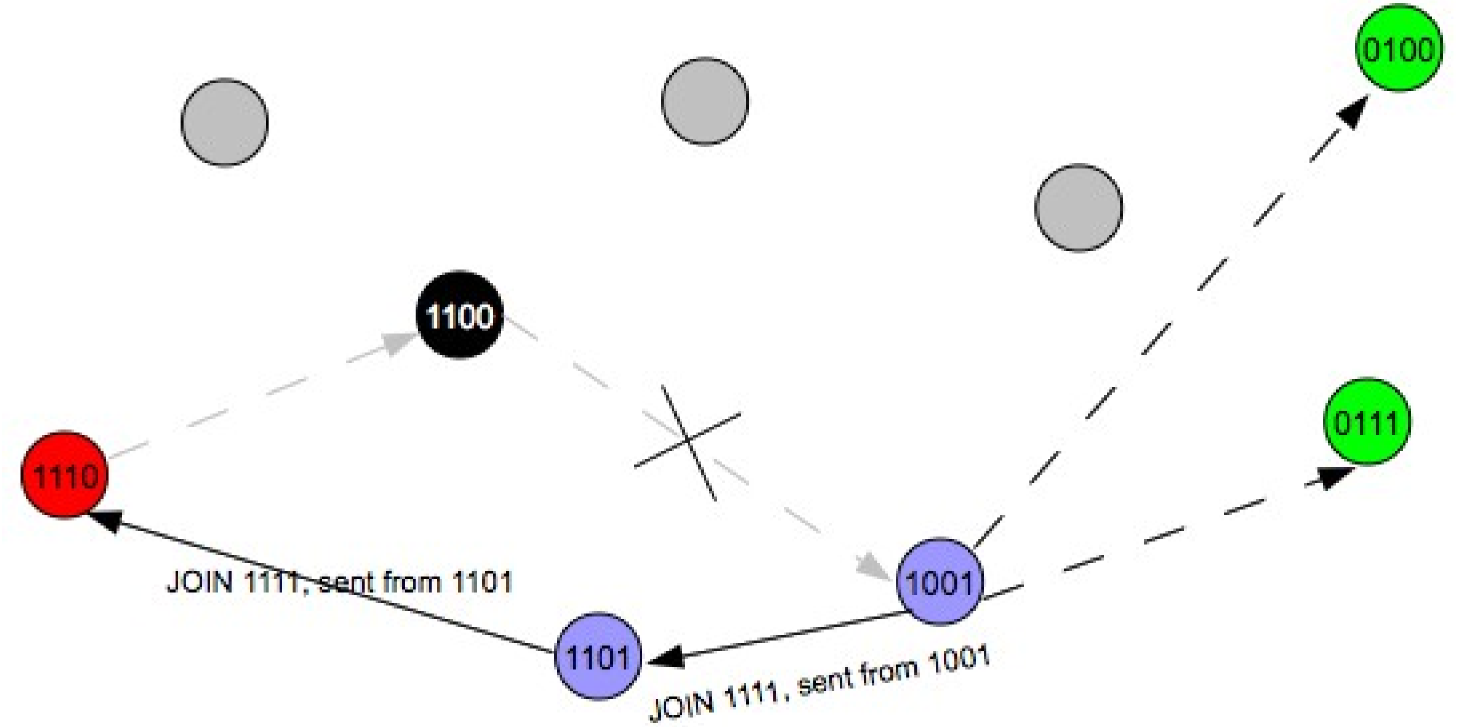
Scribe – multicast tree



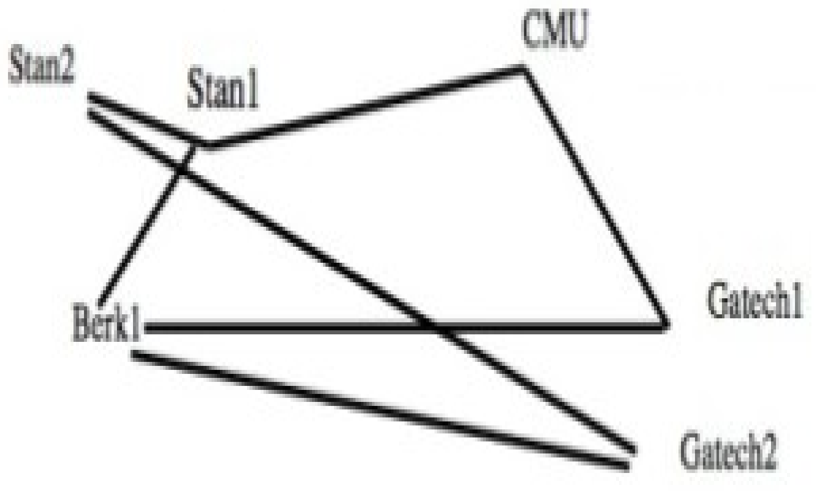
Scribe – group multicasting



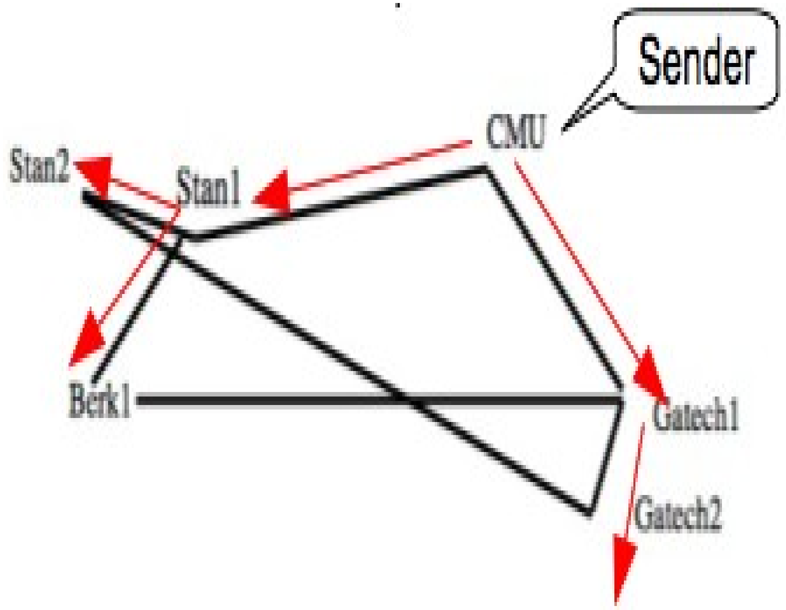
Scribe – multicast tree repairing



Streaming topologies – Mesh-first



Poor mesh topology



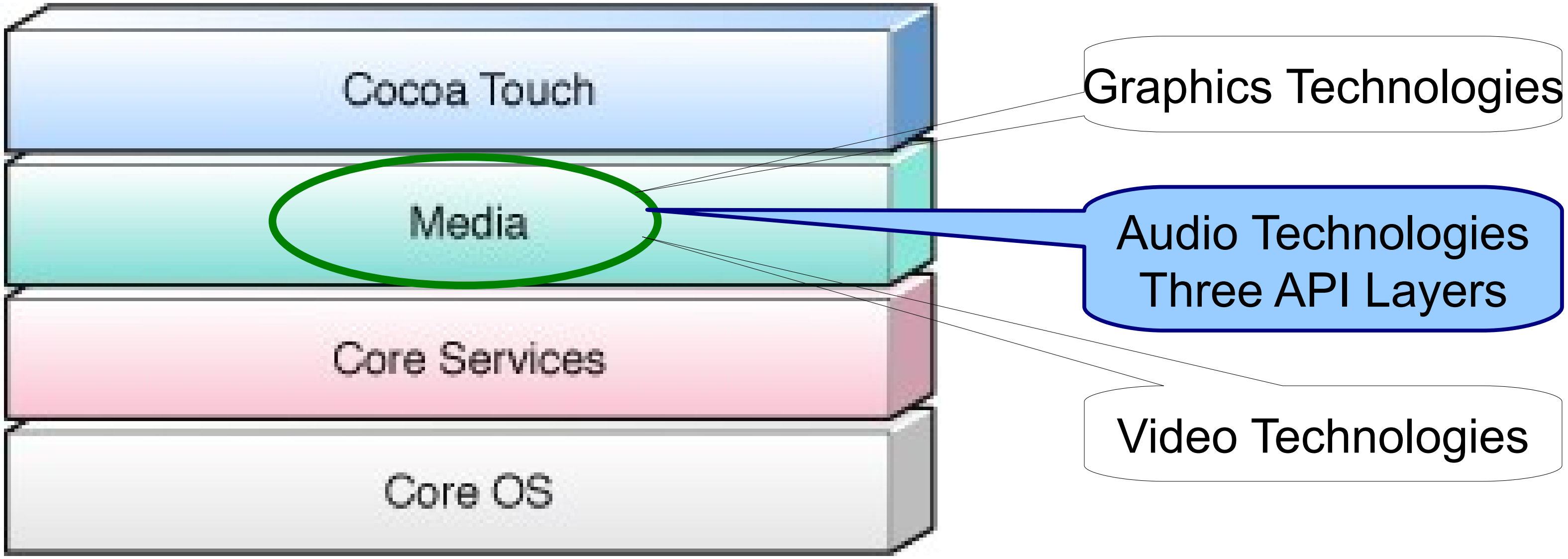
Streaming tree is built
On improved mesh

Communication overhead?

Audio Processing on the iPhone

- Layers of iPhone OS
- The three API layers of Core Audio
  - Audio Queue Services
  - Audio File Stream Services
  - Audio Unit Services (Converter unit, Multichannel Mixer unit, remoteIO unit)
  - Audio Processing Graph Services
  - etc. (not used in Pas4i)
- Audio Codec

Layers of iPhone OS

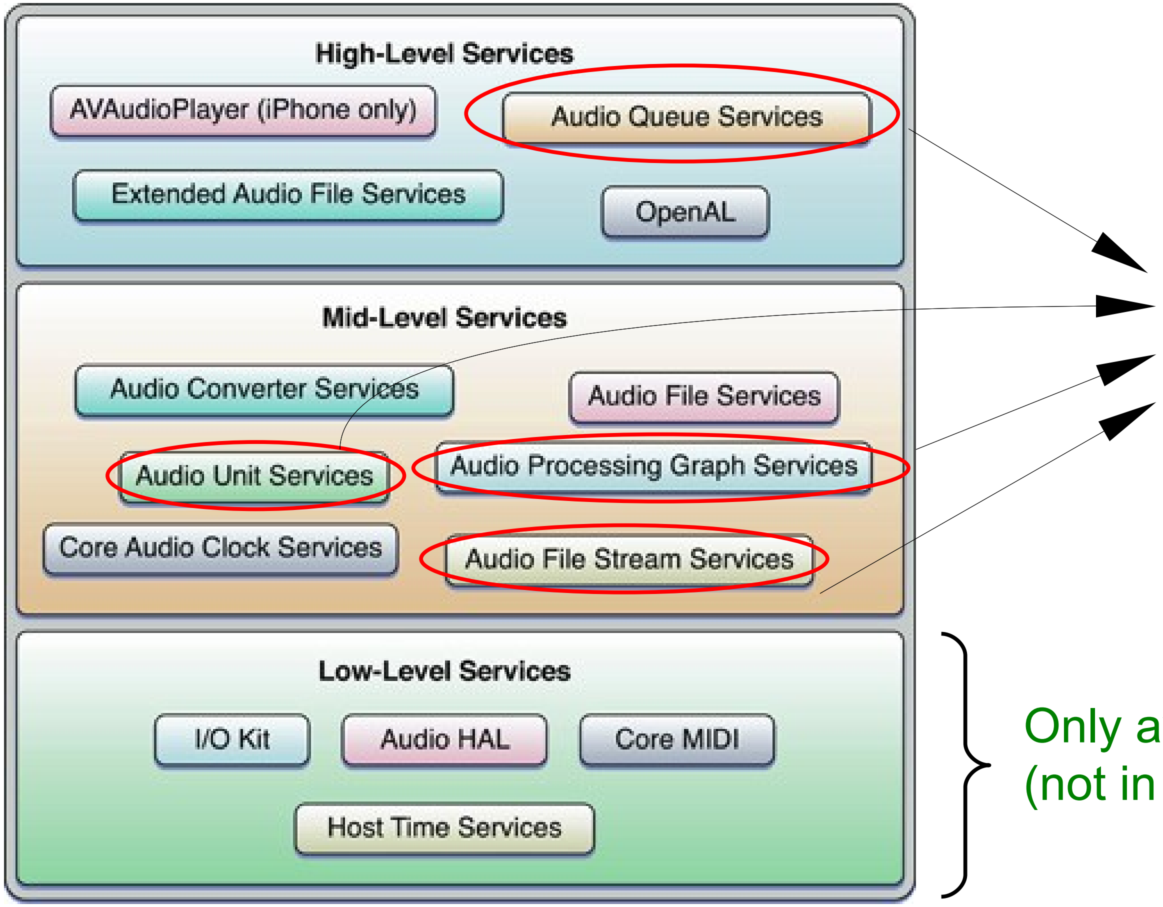


Graphics Technologies

Audio Technologies
Three API Layers

Video Technologies

The three API layers of Core Audio



Used by PAS4i

Only available for Mac OS X
(not in iPhone OS)

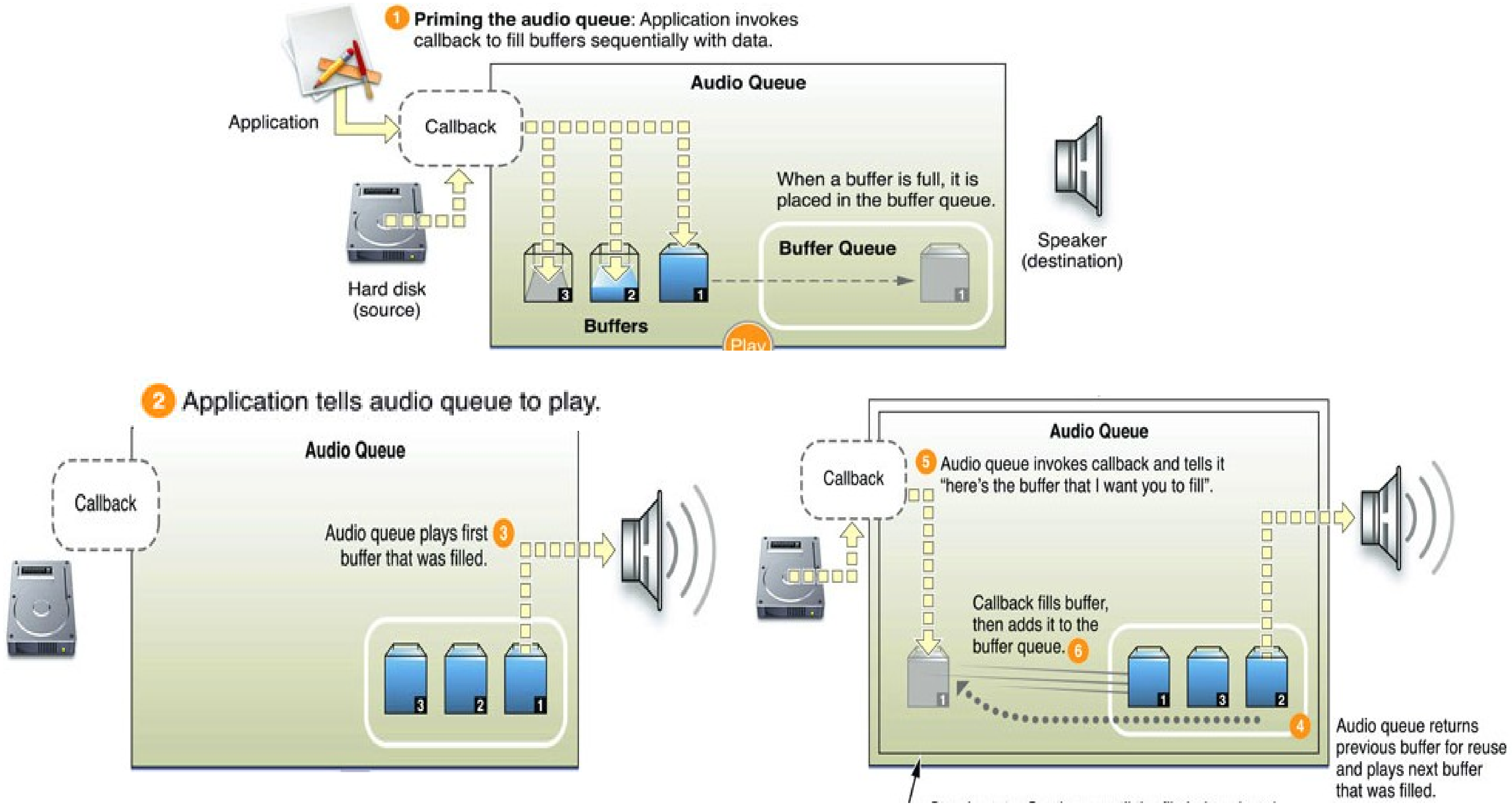
Audio File Stream Services (1st)

- To play streamed audio content (i.e., from network connection)
- Used with Audio Queue Services
- Parse audio packets and metadata in a network bitstream or on-disk files (two callback functions for audio packet and metadata)
- Supported formats: MP3, AAC, WAVE, MPEG-4 etc.

Audio Queue Services (AQS) (2nd)

- Provides low overhead way to record and play audio
- AQS and AVAudioPlayer class: the only ways to play compressed audio (e.g., MP3)
- Uses with Audio File Services or Audio File Stream Services
- Audio Queue Callback Functions for Recording and Playback

Playback with AQS (2nd)



1 **Priming the audio queue**: Application invokes callback to fill buffers sequentially with data.

When a buffer is full, it is placed in the buffer queue.

2 Application tells audio queue to play.

Audio queue plays first 3 buffer that was filled.

5 Audio queue invokes callback and tells it "here's the buffer that I want you to fill".

Callback fills buffer, then adds it to the buffer queue. 6

4 Audio queue returns previous buffer for reuse and plays next buffer that was filled.

Recording with AQS (2nd)

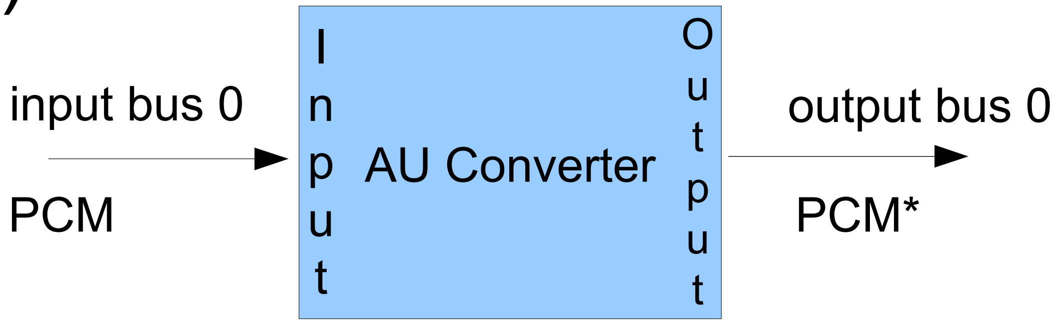
- For recording, the audio callback has a converse role
- Not used in the PAS4i: WHY?
  - recording callback occurred only every 16384 samples (latency of 0.37s for 44.1 kHz)
  - unacceptable for real-time audio processing
  - => have to use Audio Unit remoteIO (callback occurred for every 1024 samples)

Audio Unit Services (AU) (3rd)

- Provides lowest latency audio, especially when doing simultaneous input and output e.g., VoIP
  - Converter unit
  - Multichannel mixer unit
  - RemoteIO unit
  - etc. (not used in PAS4i)

AU Converter (3rd)

- Provides sample rate, bit depth, and bit format (linear to fixed-point) conversions
- Conversion is done only between PCM (Audio Converter Services for conversion with other compressed formats)



input bus 0

PCM

Input

AU Converter

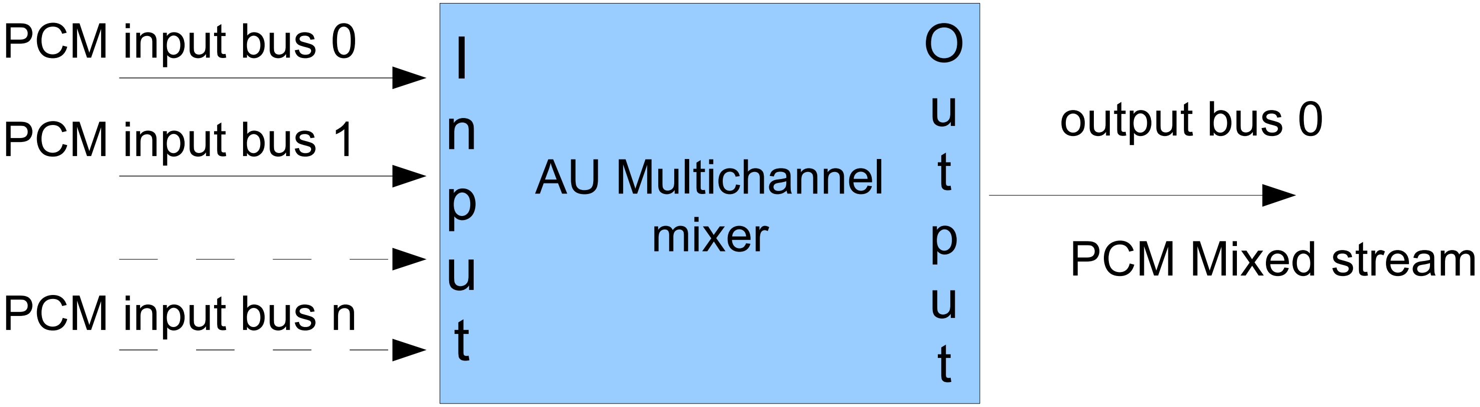
Output

output bus 0

PCM*

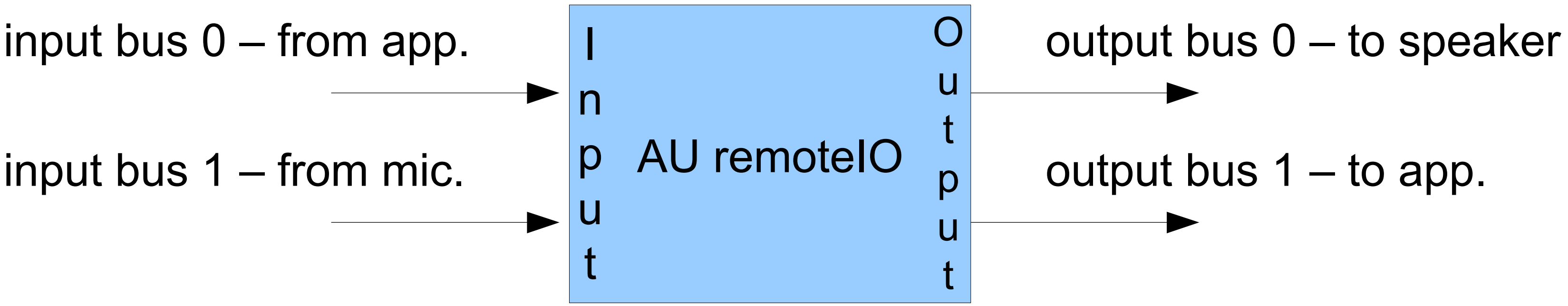
AU Multichannel mixer (3rd)

- Used to mix multiple audio streams to a single stream
- Only works with PCM



AU remoteIO (3rd)

- Allows to connect to inputs (from microphone or from application data) and outputs (to speaker or to application)
- Only works with PCM

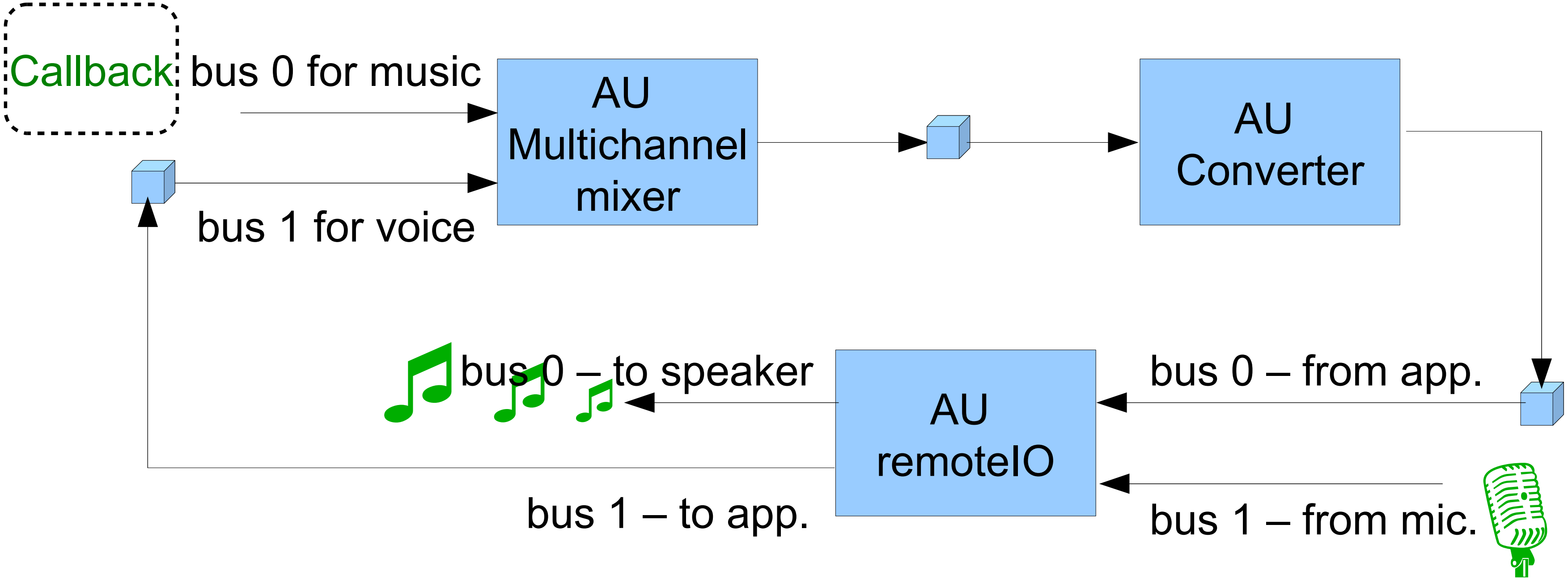


Audio Processing Graph Services (4th)

- Provides interfaces for representing a set of audio units, connections between their inputs and outputs, and callbacks used to provide inputs

Audio Processing Graph Services (4th)

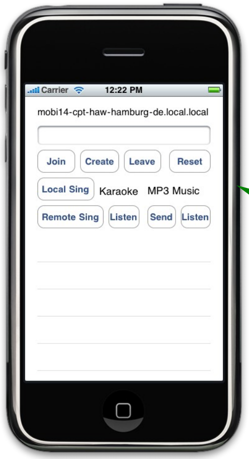
- Graph connection for audio processing



iPhone audio Codec

- Hardware-assisted codec
  - For decoding MP3, ACC, ALAC
  - Offline rendering: AudioQueueOfflineRender
  - High performance: fast and save battery
  - Only play a single format
- Software codec (Audio Converter Services)
  - Support a wide range of audio data format
  - Support simultaneous playback of multiple sounds

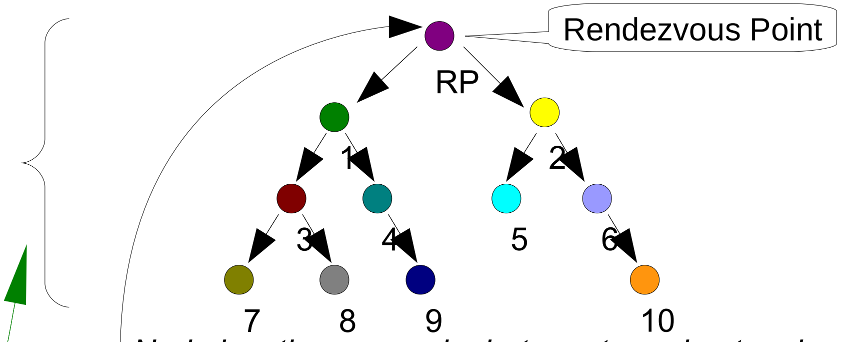
PAS4i - Layers



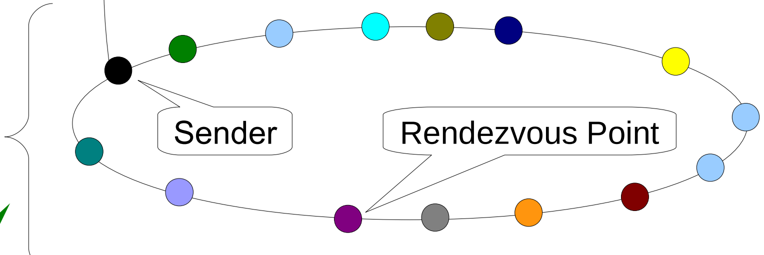
GUI & App. Logic & Audio Processing

Scribe: multicast tree

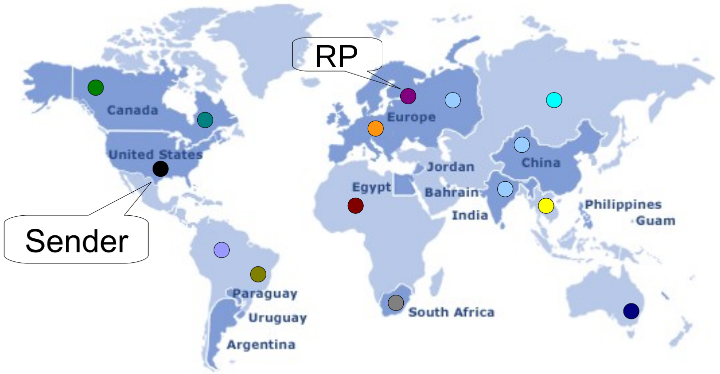
Pastry: overlay network



Node locations on a single-tree streaming topology



Node locations on the overlay P2P network



Node locations on the underlay IP network

PAS4i - Components

- oRTP library for RTP and RTCP
- Chimera – KBR service (similar to Pastry)
- Scribe – Multicast service
- PAS4i – GUI, audio processing (using iPhone Core Audio Unit)

PAS4i - Functionalities

- Routing service (Pastry)
- Join the Pastry overlay (Pastry)
- Create a streaming group (Scribe)
- Join a group (Scribe)
- Leave a group (Scribe)
- Send multicast karaoke/music stream to a group (Scribe)
- Tree maintenance (Scribe – not yet)

PAS4i - Usage

- Send multicast or receive MP3 music stream (Audio File Stream Services + Audio Queue Services)
- Send multicast or receive karaoke background music stream for low PCM (Audio Unit Services: remoteIO, converter)
- Sing karaoke with local or with received background music stream and multicast to another group (Audio Processing Graph Services + Audio Unit Services: remoteIO, converter, multichannel mixer)

PAS4i – Future Implementation

- Tree maintenance function
- Using hardware-assisted and software audio codecs for streaming efficiency
- Voice chat for group communication
- Adding karaoke lyric and meta data (song name, singer, location etc.) together with the music stream
- Feedback from listeners (grading, etc.)

Discussion

- Live streaming vs. Video-on-demand streaming
- Scribe ALM via KBR conflicts heterogeneity issue for streaming systems
  - Non-DHT, key generation takes heterogeneity and locality of nodes in consideration
- Considerations: peer churn, communication overhead, NATs and Firewalls, network coding
- ISP friendly systems
  - IP Multicast in combination with ALM?
- ALM needs more study as well as ...
- PAS4i needs improvement

Questions