Peer-to-Peer Media Streaming
Focus On:

- How Media is Delivered?
- NOT on locating content or the bootstrapping process
P2P media streaming architecture

- Mesh-Pull
- Tree-Push
  - No infrastructure support
  - With partial support from infrastructure nodes
- CDNs: (not classified under P2P)
  - Full support from infrastructure nodes
Mesh-Pull Architecture

- Strong similarity to BitTorrent
  - Exchange “buffer maps” and retrieve missing chunks

- Significant difference w.r.t BitTorrent
  - Every chunk has a playback deadline
    - Video chunk scheduling algorithm indispensable
  - Peer selection algorithms:
    - Uses gossip based peer search algorithms

- Many deployed P2P IPTV services use this architecture
  Examples: PPLive, SopCast, CoolStreaming, TVAnts etc
Mesh-Pull: Overview

Mesh
Mesh-Pull Architecture contd..

Pros
- Robustness: suitable for high churn in p2p environment
- Simple: a important selling point

Cons
- High initial start-up time
- Video switching delays
- Is the stress to the underlying network higher than Tree-Push architecture ???
# Peer-to-Peer Streaming Systems

<table>
<thead>
<tr>
<th>System</th>
<th>Push/pull</th>
<th>Tree/Mesh</th>
<th>Buffer</th>
<th>Playout Delay</th>
<th>Startup Delay</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPLive</td>
<td>Pull</td>
<td>Mesh</td>
<td>2 min</td>
<td>1 min</td>
<td>20 s-2 min</td>
<td>Rate: 300 – 350 kbit/s, Res. 320 x 240 pixel</td>
</tr>
<tr>
<td>Coolstreaming</td>
<td>Pull</td>
<td>Mesh</td>
<td>2 min</td>
<td>1 min</td>
<td>1 min</td>
<td>1 min</td>
</tr>
<tr>
<td>Anysee</td>
<td>Push</td>
<td>Hybrid</td>
<td>40 s</td>
<td>20-30 s</td>
<td>20 s</td>
<td>20 s</td>
</tr>
<tr>
<td>SopCast</td>
<td>Pull</td>
<td>Mesh</td>
<td>1 min</td>
<td>1 min</td>
<td>1-5 min</td>
<td>1-5 min</td>
</tr>
</tbody>
</table>

- **More**
  - CoopNet
  - PALS
  - PROMISE
  - SPLIT Stream
  - Bullet
Tree-Pull Architecture

▶ Important components
- Distribution tree construction (ex: based on RTT)
- Distribution tree maintenance
  - Pro-Active node re-positioning
- Distribution tree repair (in case of churn)

▶ Operations related to distribution tree can be
- Centralized (single entity performs the task) OR
- Distributed
  ▪ At any instant, more than one entity is trying to make decisions that optimize the distribution
Tree-Pull Architecture contd..

- P2P media distribution solutions based on Tree-Pull
  - (A) No infrastructure nodes
    - purely based on end users
  - (B) With partial support from infrastructure nodes
(A) No Infrastructure nodes

Any Deployment Experience: ESM (End System Multicast)
http://esm.cs.cmu.edu/
(B) Partial support from Infrastructure nodes

Tree operations:
Interconnecting infrastructure node might use different tree algorithms from the algorithms used for build tree within normal users.
Tree-Push Architecture contd..

- **Pros**
  - Start-up time is small
  - Playback time lag among peers is less
  - Tree construction mechanism, can optimize the tree so as to reduce stress on the underlying network

- **Cons**
  - Stability
    - Loss of a node affects all other nodes below its hierarchy.
Stability Issue in Tree-Push

- One proposed Solution: Using multiple trees
  - Media stream is split into 'n' independent streams
  - Each stream is independently decodable
  - Construct distribution tree for every stream

So, instability due to churn can be minimized
# Comparison: Tree vs. Mesh

<table>
<thead>
<tr>
<th></th>
<th>Push</th>
<th>Pull</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Overlay</strong></td>
<td>Maintains multiple transmission trees</td>
<td>Defines partnership mesh and for the whole streaming session</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Schedules block of packets</td>
</tr>
<tr>
<td><strong>Sign of infeasibility</strong></td>
<td>Reconnection failure</td>
<td>Infeasible transmission schedule</td>
</tr>
<tr>
<td><strong>Delay control</strong></td>
<td>Tree structure</td>
<td>Parent selection and scheduling</td>
</tr>
<tr>
<td><strong>Loss control</strong></td>
<td>Redundancy and retransmission</td>
<td>Redundancy, scheduling, including retransmission and network coding</td>
</tr>
<tr>
<td><strong>Bandwidth utilization</strong></td>
<td>Tree construction and maintenance</td>
<td>Scheduling</td>
</tr>
<tr>
<td><strong>Performance optimization</strong></td>
<td>Tree maintenance</td>
<td>Scheduling and parent reselection</td>
</tr>
<tr>
<td><strong>Resilience to churn</strong></td>
<td>Tree construction and loss control</td>
<td>Mesh maintenance, scheduling, and loss control</td>
</tr>
<tr>
<td><strong>Control cost</strong></td>
<td>Tree maintenance</td>
<td>Mesh maintenance and packet pulling</td>
</tr>
<tr>
<td><strong>Trades resilience for</strong></td>
<td>Redundancy and control</td>
<td>Delay and control</td>
</tr>
</tbody>
</table>
Summary..

- Many real world P2P streaming solutions use Mesh-Pull Tree-Pull architecture have largely been in research stage
- For large scale P2P streaming: Many open questions
  - Improving efficiency of the distribution mechanisms
    - Both in network and application perspectives
      - Avoid carrying duplicate traffic in the link
    - Scalability to large number of users
    - Self-Organizing to adapt changing node dynamics (churn)
    - Heterogeneity in user's contributing bandwidth
      - Asymmetric DSL lines
    - Copyrights: DRM issues
    - Firewalls, NAT
CDN based approach

- Not classified under P2P
- Use replication of data by large scale deployment of infrastructure nodes
- Real World Example:
  Akamai Content Distribution Network (CDN)
CDN based approach contd..
Using CDNs to connect IP Multicast clouds
References & Further reading

- IPTV over P2P Streaming Networks: the Mesh-pull Approach, By Xiaojun Hei, Yong Liu and Keith W. Ross
- A Measurement Study of a Large-Scale P2P IPTV System
  Xiaojun Hei, Chao Liangz, Jian Liangy, Yong Liuz and Keith W. Ross
- Opportunities and Challenges of Peer-to-Peer Internet Video Broadcast
  Jiangchuan Liu, Sanjay G. Rao, Bo Li, and Hui Zhang
- The Feasibility of Supporting Large-Scale Live Streaming Applications with Dynamic Application End-Points
  Kunwadee Sripinidkulchai, Aditya Ganjam, Bruce Maggs, and Hui Zhang
- Construction of an Efficient Overlay Multicast Infrastructure for Real-time Applications
  Suman Banerjee, Christopher Kommareddy, Koushik Kar, Bobby Bhattacharjee, Samir Khuller