P2Cast: Peer-to-Peer Patching Scheme for VoD Service

Yang Guo, Kyoungwon Suh, Jim Kurose, and Don Towsley

yguo,kwsuh,kurose,towsley@cs.umass.edu

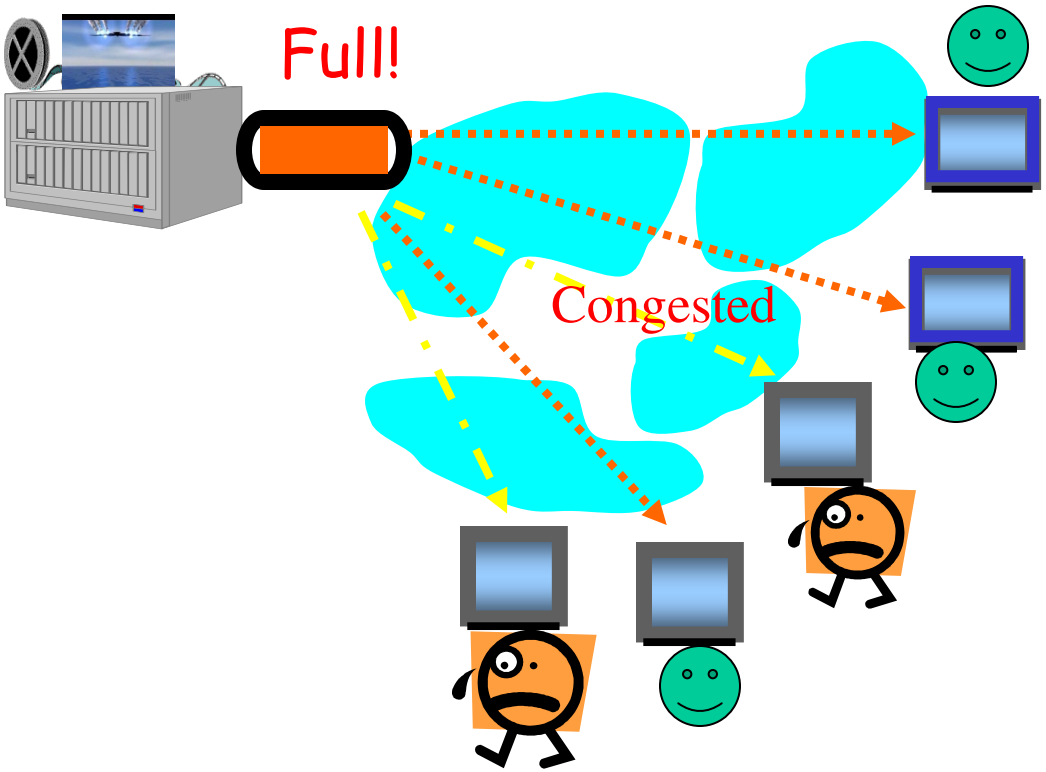
presented by Kyoungwon Suh

University of Massachusetts
Department of Computer Science

Outline

- Motivation
- Previous Work
- P2Cast: Peer-to-Peer Patching
- Overlay construction
- Providing continuous playback
- Performance evaluation
- Summary and future work

Challenges of VoD Service



- High-bandwidth, long duration
- Large number of clients, asynchronous requests
- Client-server service model doesn't scale

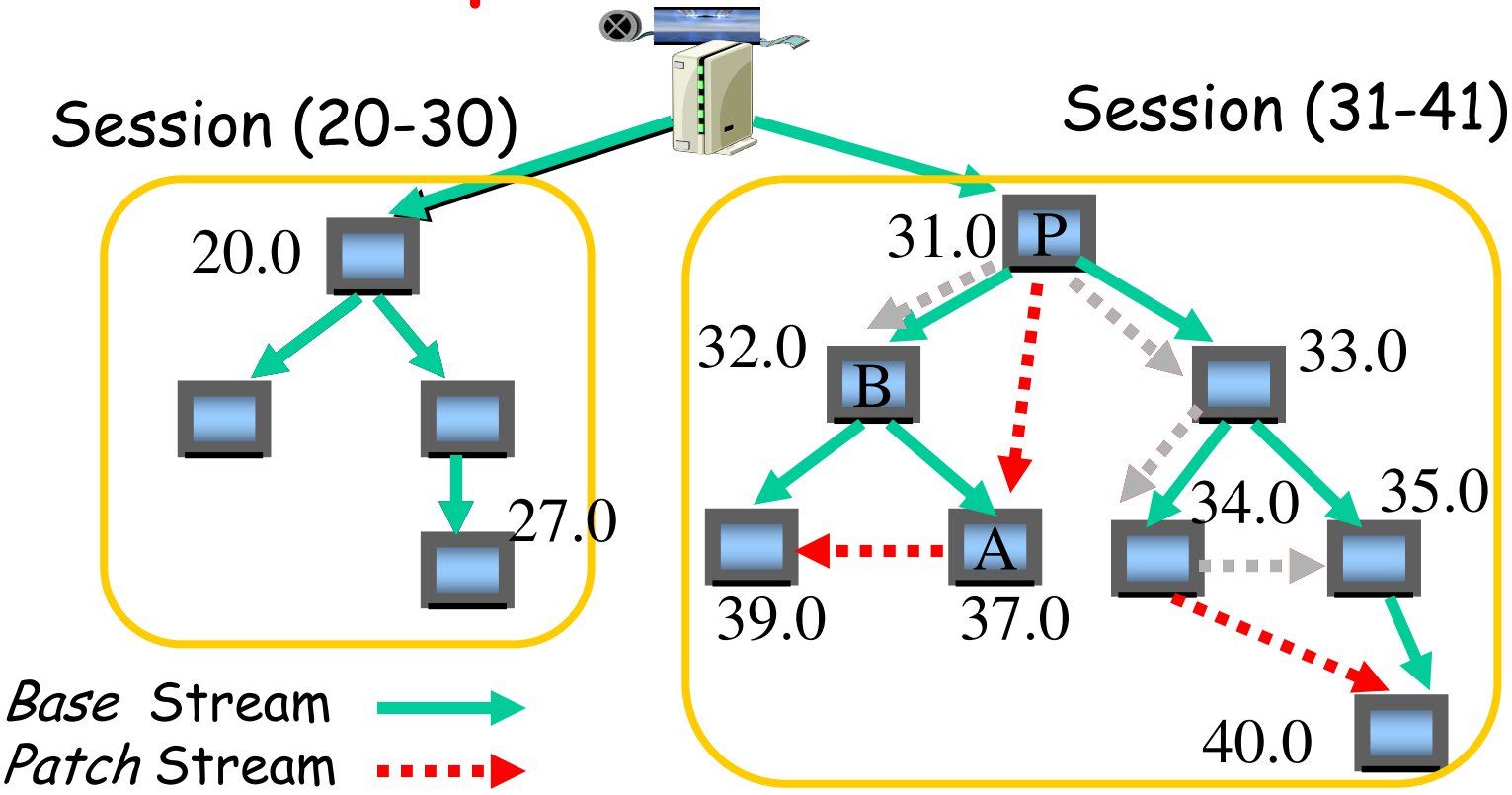
Previous Work on VoD Service

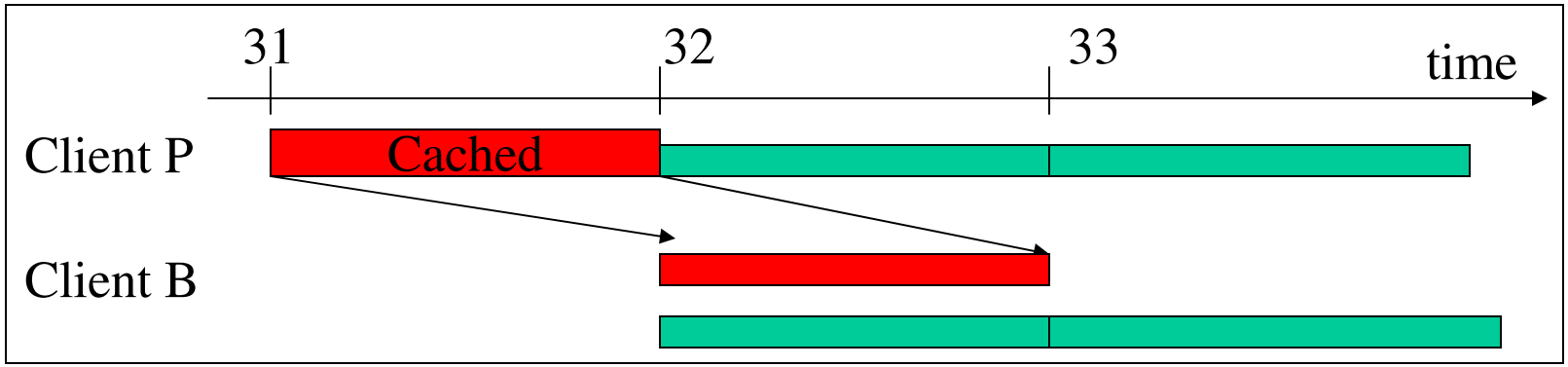
- IP multicast-based approach
  - Patching, Periodic Broadcast, Stream Merging
  - Problem: IP multicast has not been deployed widely
- Proxy Caching / CDN
  - Problems:
    - Infrastructure and maintenance cost
    - Still not scalable
- Peer-to-peer networks
  - Live streaming: SplitStream, PeerCast
  - P2P VoD: CoopNet, DirectStream

P2Cast: Peer-to-Peer Patching Scheme

- Requirements:
  - Unicast connection only
- Idea:
  - Participating clients cooperate with each other
    - Forward stream to other clients
    - Cache and serve the initial part of the video to other clients
- Solution:
  - Clients arriving close in time form a ***session***
    - Closeness is determined by **Threshold**
  - Establishing overlay among peers
    - Stream can be shared by all clients
  - Missing *patch* can be retrieved from early-arrived clients

A Snapshot of P2Cast



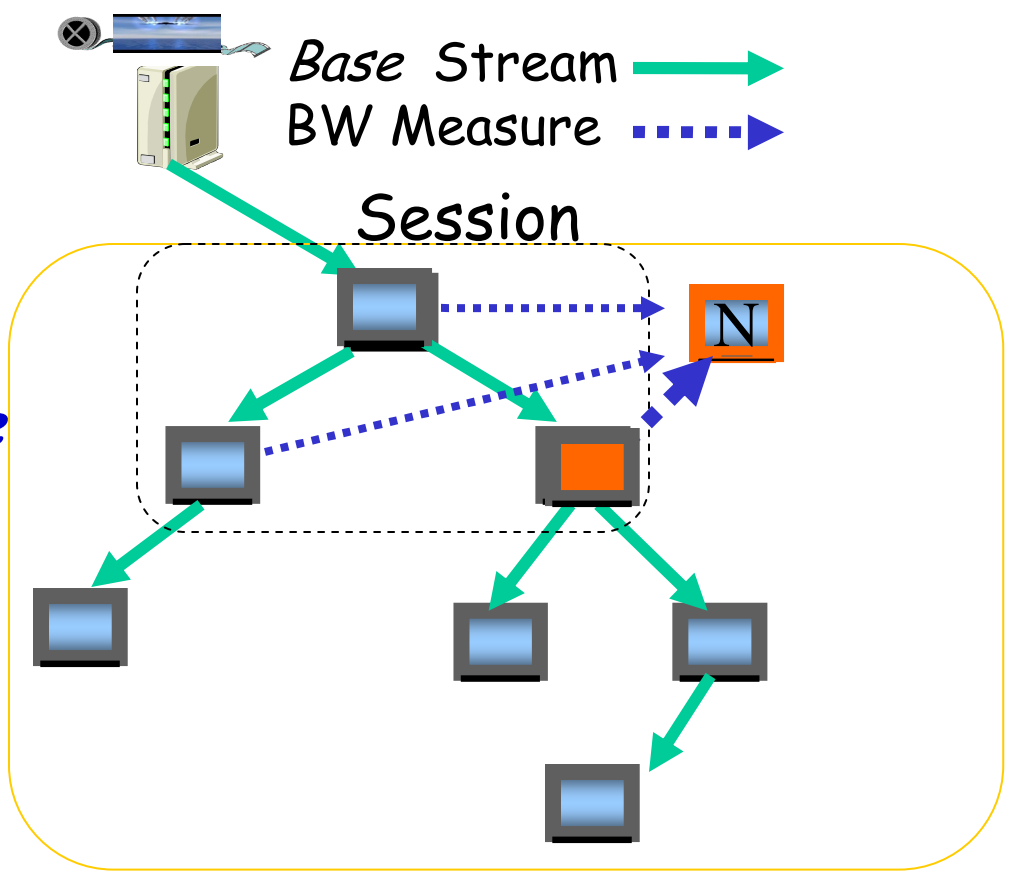


Challenge for P2Cast

- Constructing application overlay appropriate for streaming
  - Supports sufficient Bandwidth to transmit the stream
  - Our solution: Best Fit (BF) Algorithm

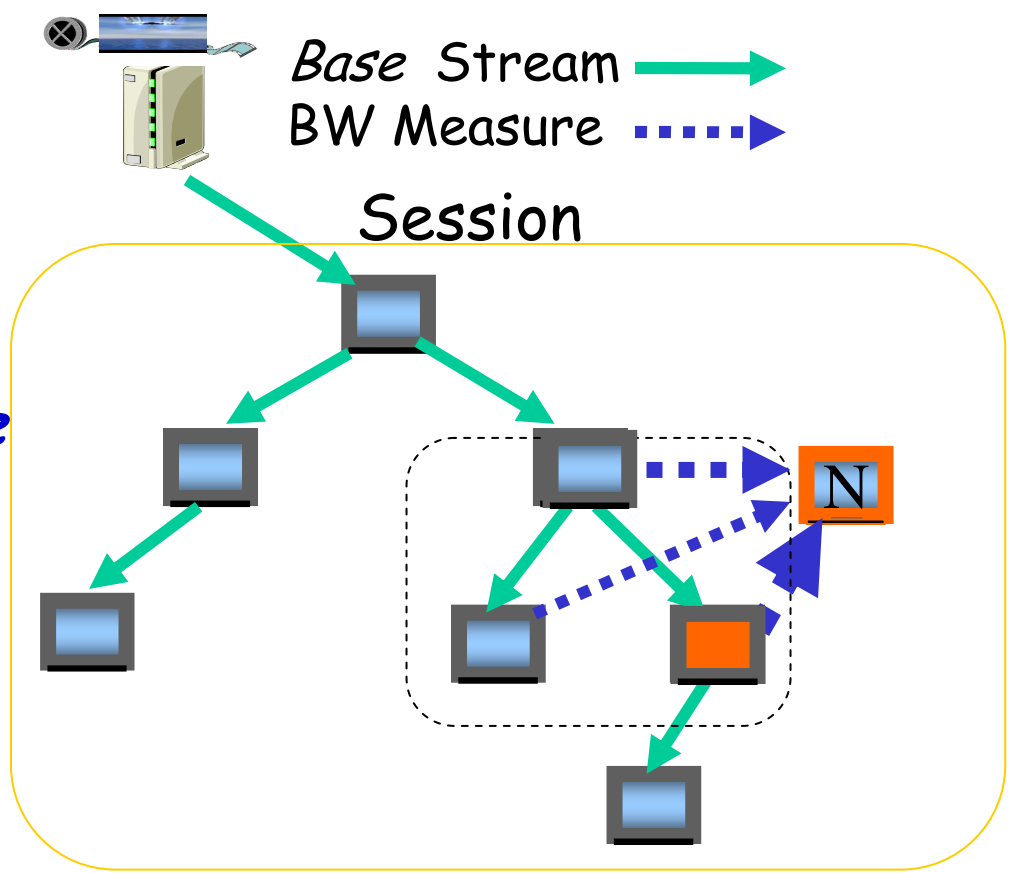
"Best-fit (BF)" Overlay Construction

- Design Principles:
  - Bandwidth (BW)-first principle
  - Local information only principle
- Idea: Try to use Maximum ***Available BW***
  - Measure BW in the unit of playback rate



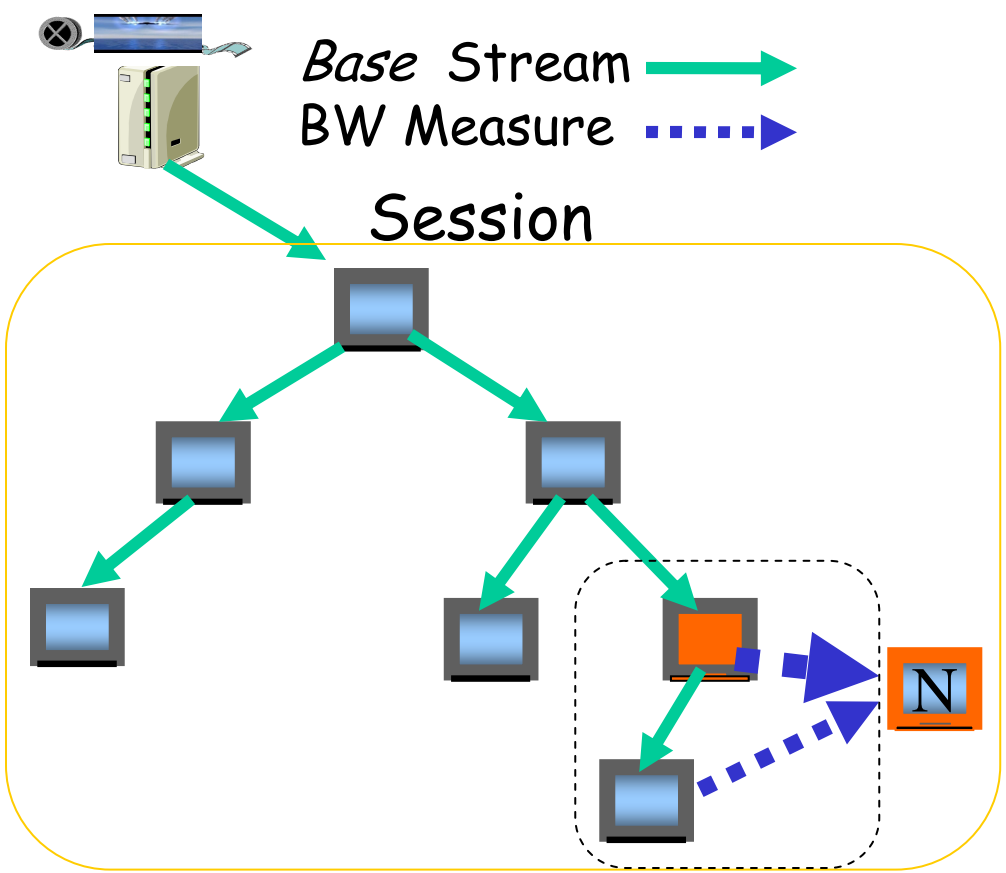
"Best-fit (BF)" Overlay Construction

- Design Principles:
  - Bandwidth (BW)-first principle
  - Local information only principle
- Idea: Try to use Maximum ***Available BW***
  - Measure BW in the unit of playback rate



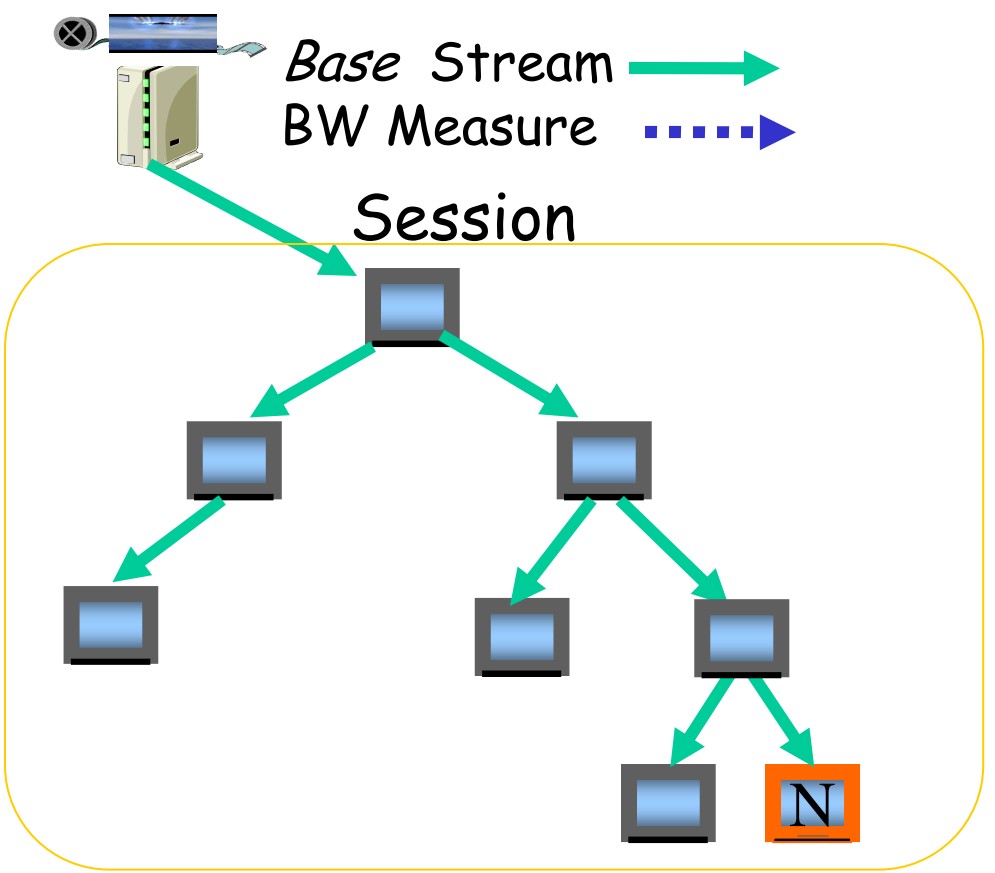
"Best-fit (BF)" Overlay Construction

- Design Principles:
  - Bandwidth (BW)-first principle
  - Local information only principle
- Idea: Try to use Maximim Available BW
  - Measure BW in the unit of playback rate



"Best-fit (BF)" Overlay Construction

- Design Principles:
  - Bandwidth (BW)-first principle
  - Local information only principle
- Idea: Try to use Maximize Available BW
  - Measure discretized avail. bandwidth rounded to playback rate

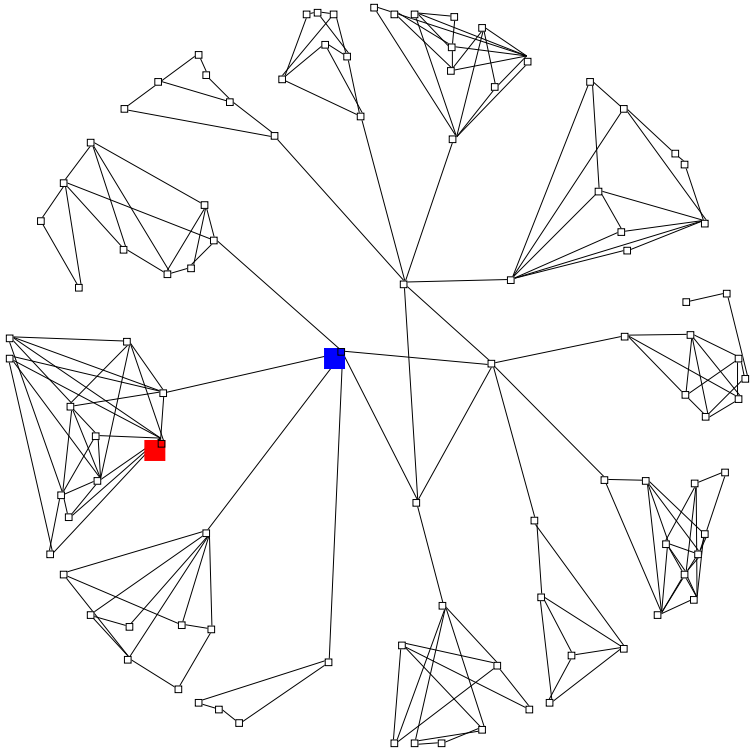


Two variations of BF: BF-Delay and BF-Delay-approx

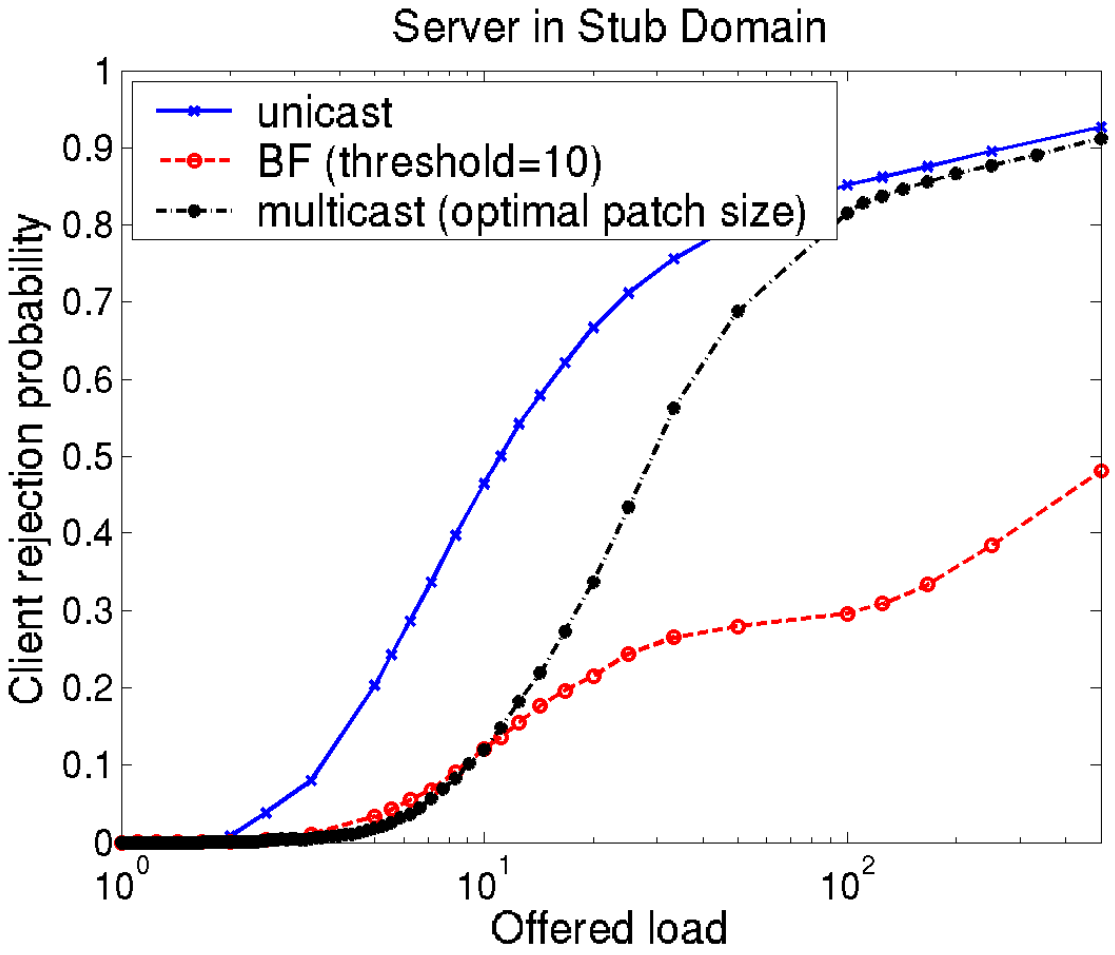
- BF-delay
  - Network delay information is used to break tie
    - Choose a closer client
    - Improve the *overall scalability*
- BF-delay-approx
  - Choose a peer with *sufficient* BW with delay info
    - Check Avail BW can sustain playback rate
    - Reduce measurement overhead
    - *Decrease joining delay*

Performance Evaluation

- Simulation settings:
  - Transit-Stub network topology generated by GT-ITM (100 nodes)
  - Node: Abstraction of a local network
  - One Transit Network, 12 stub domains
  - 95% confidence interval is within 5% of the point estimate

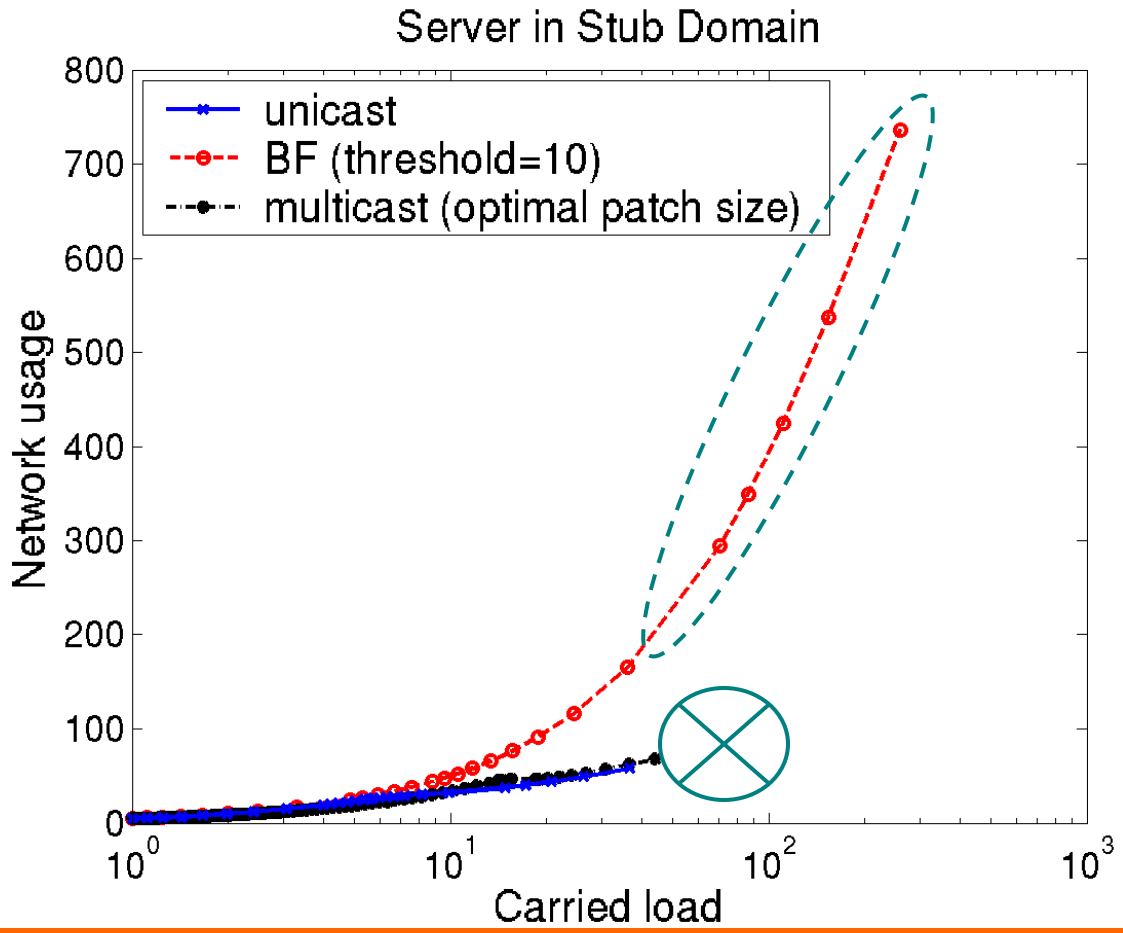


Client Rejection Probability



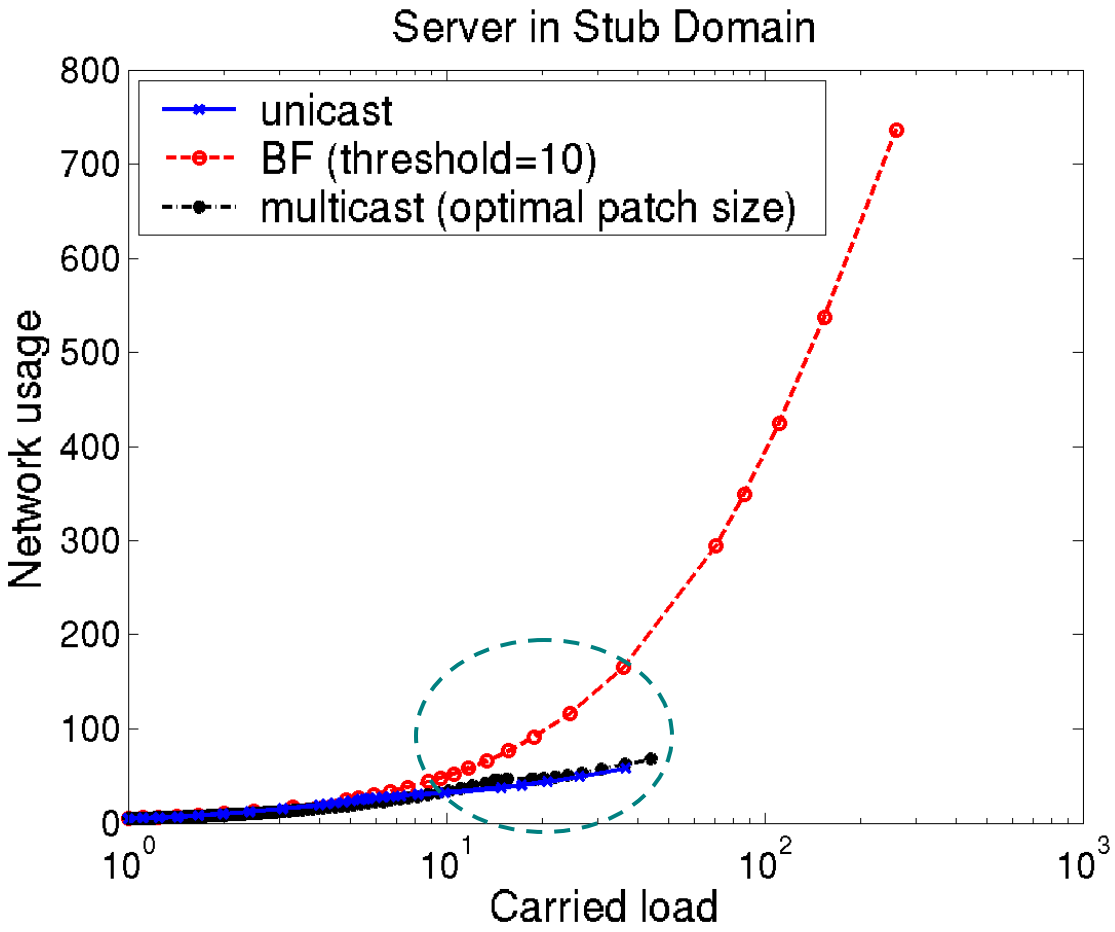
Rejection Probability in P2Cast using BF algorithm is smallest

Network usage



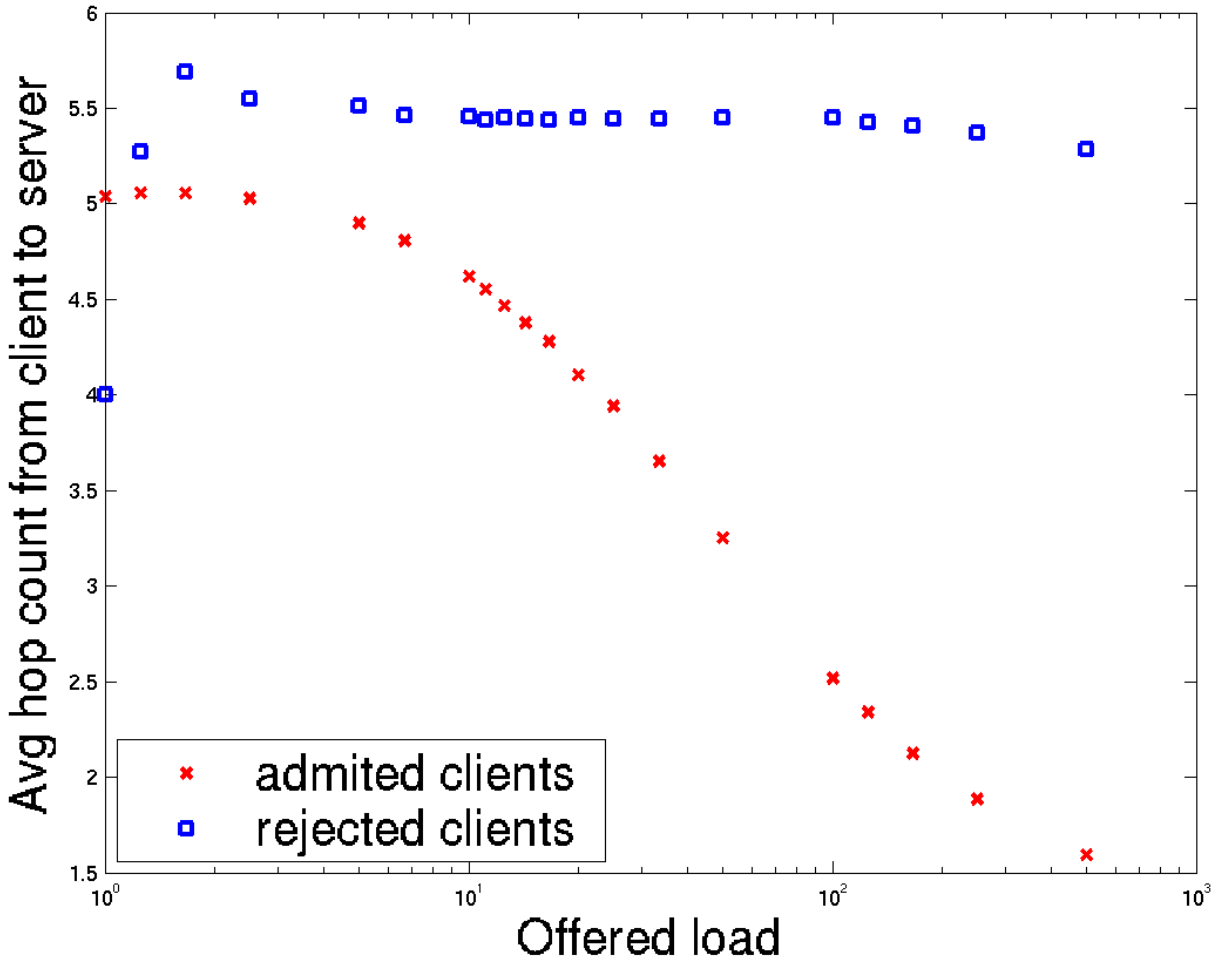
Network resources can be fully utilized in P2Cast with BF algorithm

Network usage

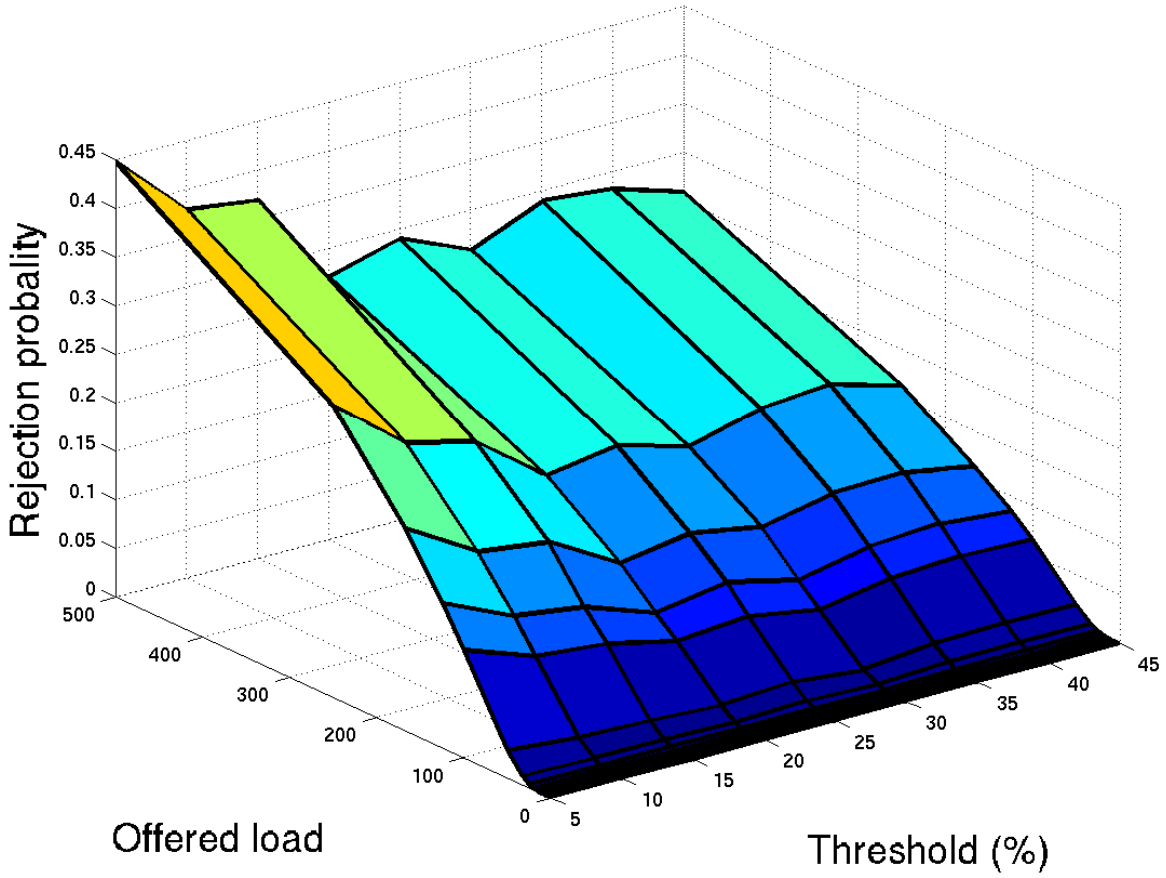


Why does P2Cast with BF algorithm have higher network usage than Unicast?

Hop Count for Unicast



Threshold impact on the performance of P2Cast

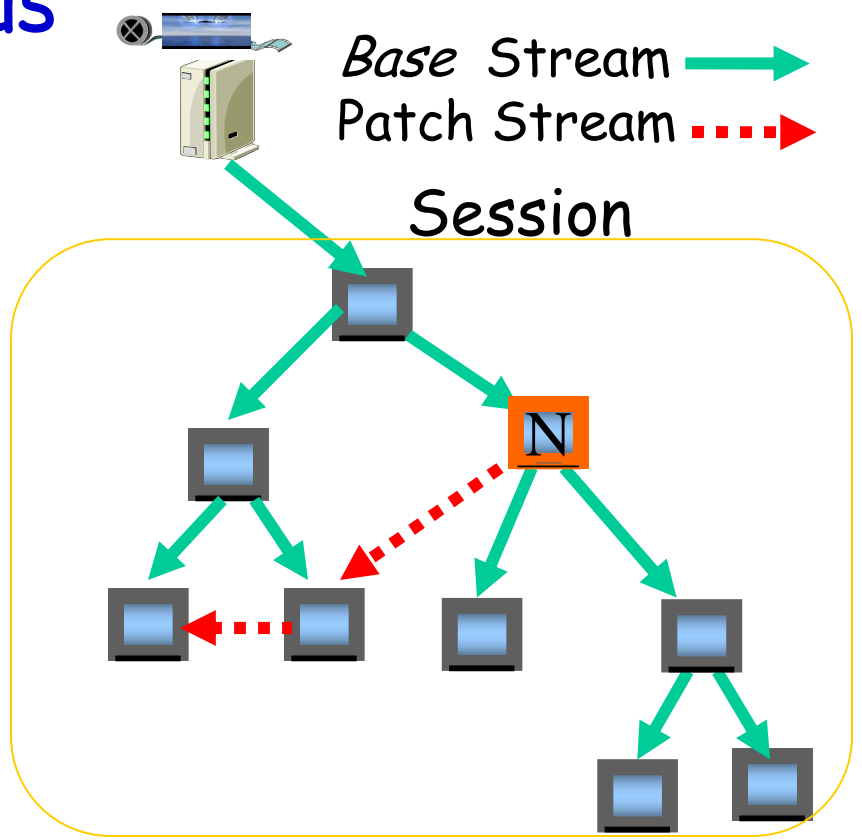


Larger Threshold ➔ Better Scalability

Small threshold is sufficient

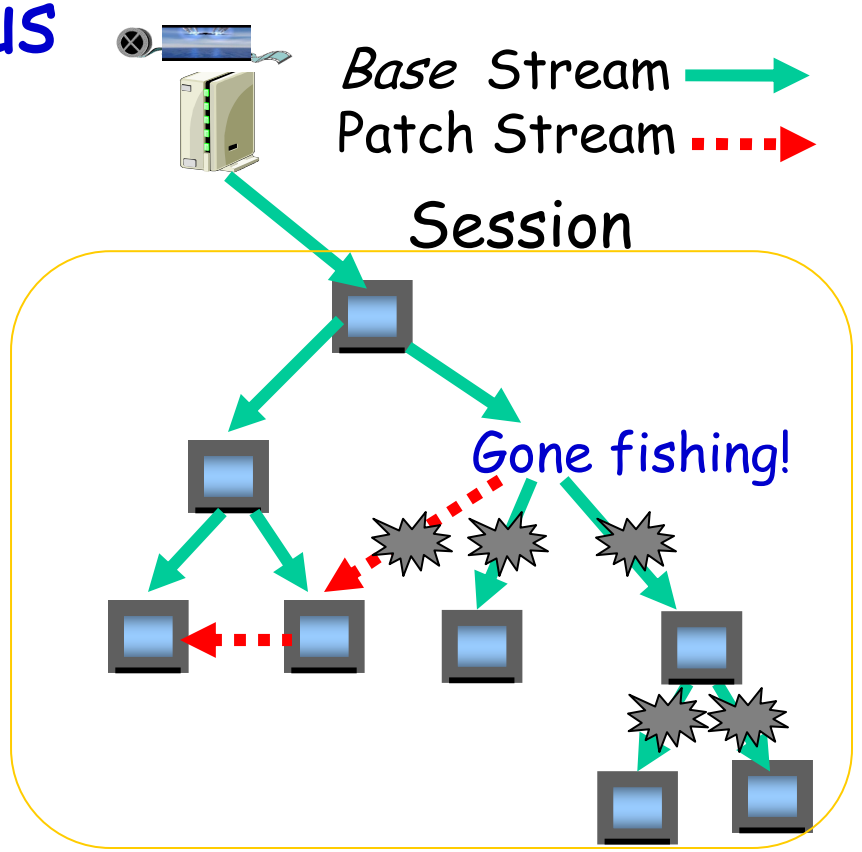
Challenge: providing continuous playback with departing clients

- Providing continuous stream playback without glitch
  - Resilient to disruption from early departing clients



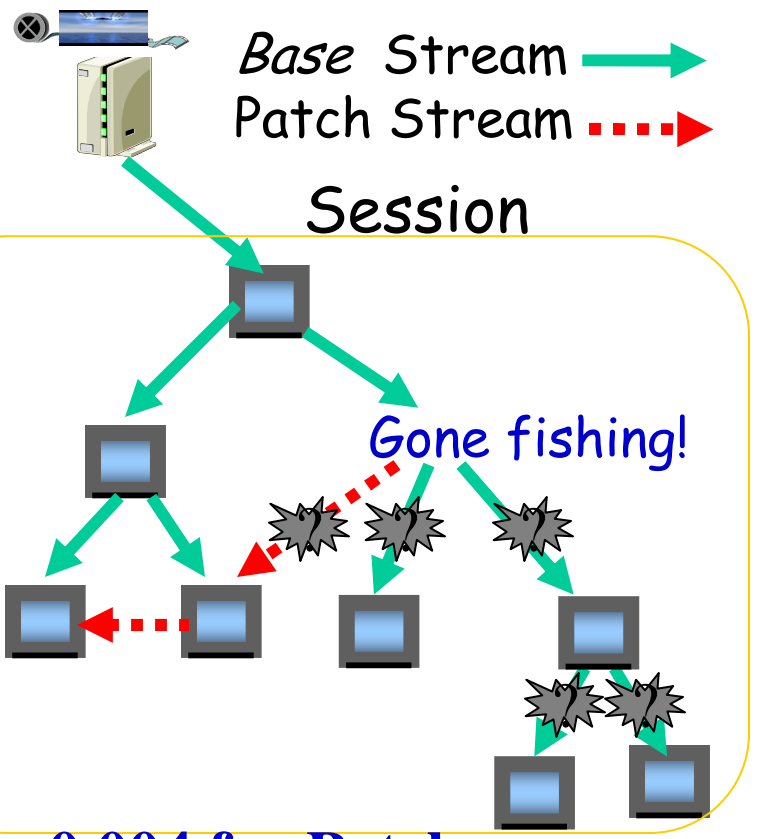
Challenge: providing continuous playback with departing clients

- Providing continuous stream playback without glitch
  - Resilient to disruption from early departing clients



Challenge: providing continuous playback with departing clients

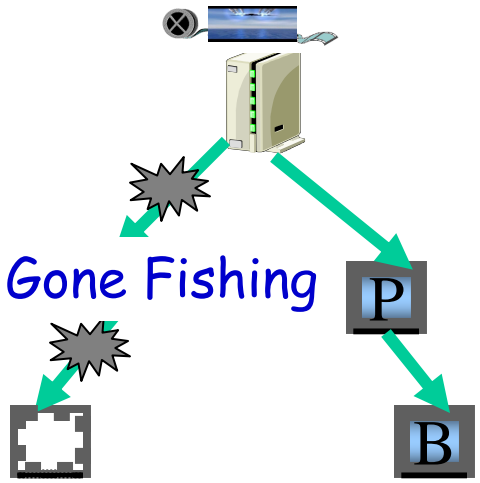
- Providing continuous stream playback without glitch
  - Resilient to disruption from early departing clients

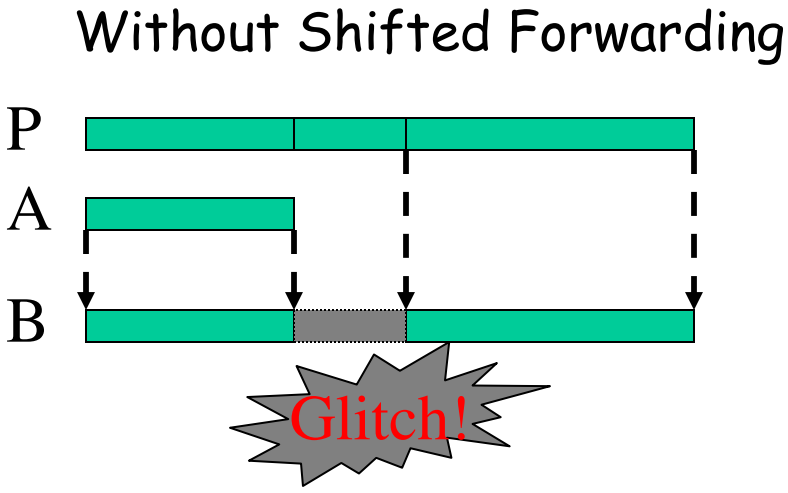


How Often? **0.004 for Patch**
0.0017 for Base

Providing Continuous Playback

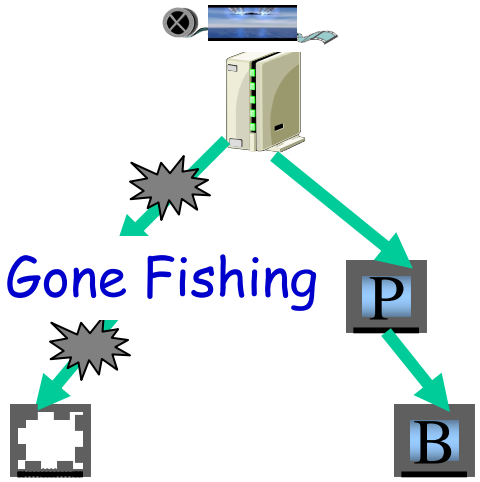
- **Limit session size**
- Buffering patch stream
- Shifted forwarding for base stream
  - Use of interval caching

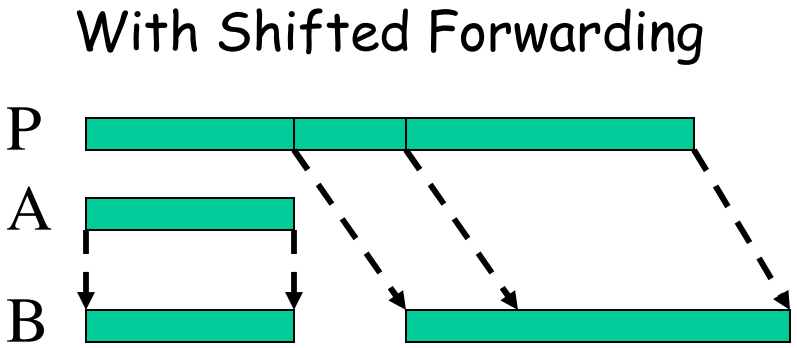




Providing Continuous Playback

- **Limit session size**
- Delay the start of playback for patch stream
- Shifted forwarding for base stream





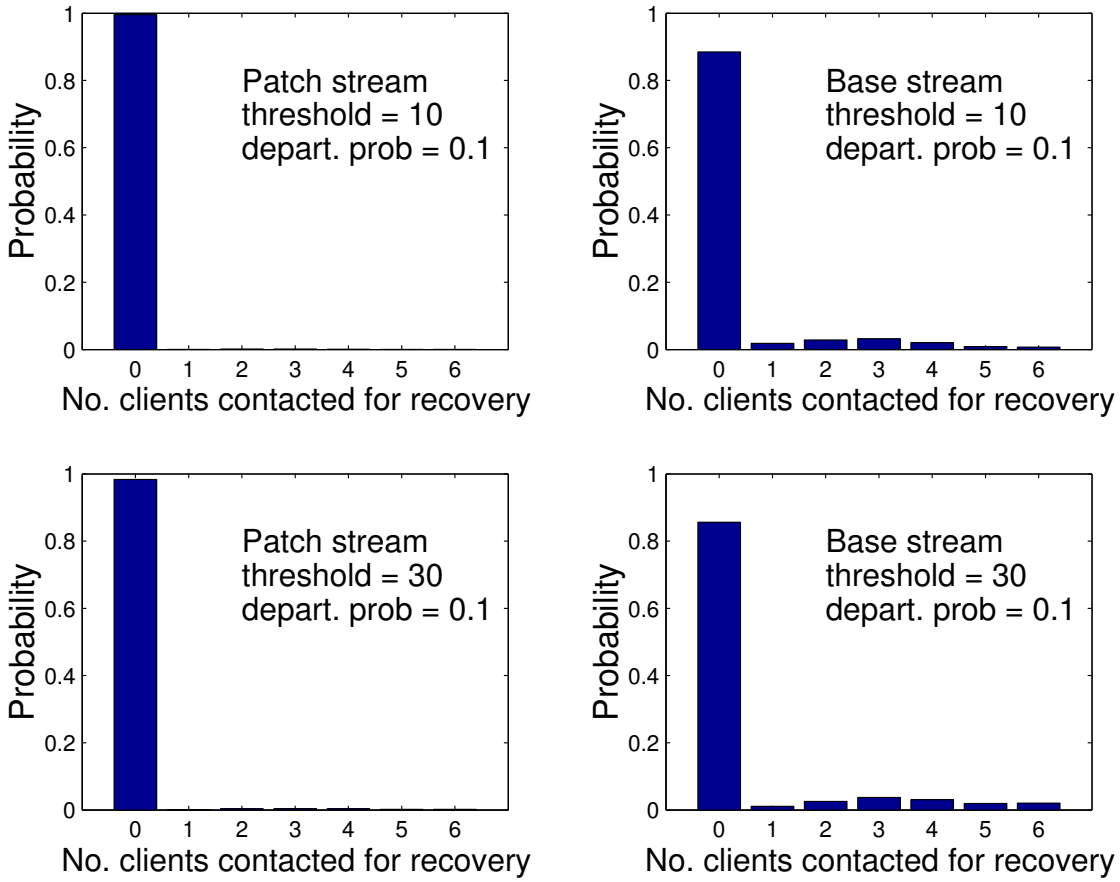
Summary and Future work

- Proposed P2P patching scheme
  - Constructing application overlay appropriate for streaming
  - Providing continuous stream playback
- P2Cast outperforms:
  - unicast-based scheme and
  - native IP multicast-based patch scheme
- Experiment of P2Cast on the Internet

The End

Backup Slides

Probability of no. candidate clients contacted for **recovery**

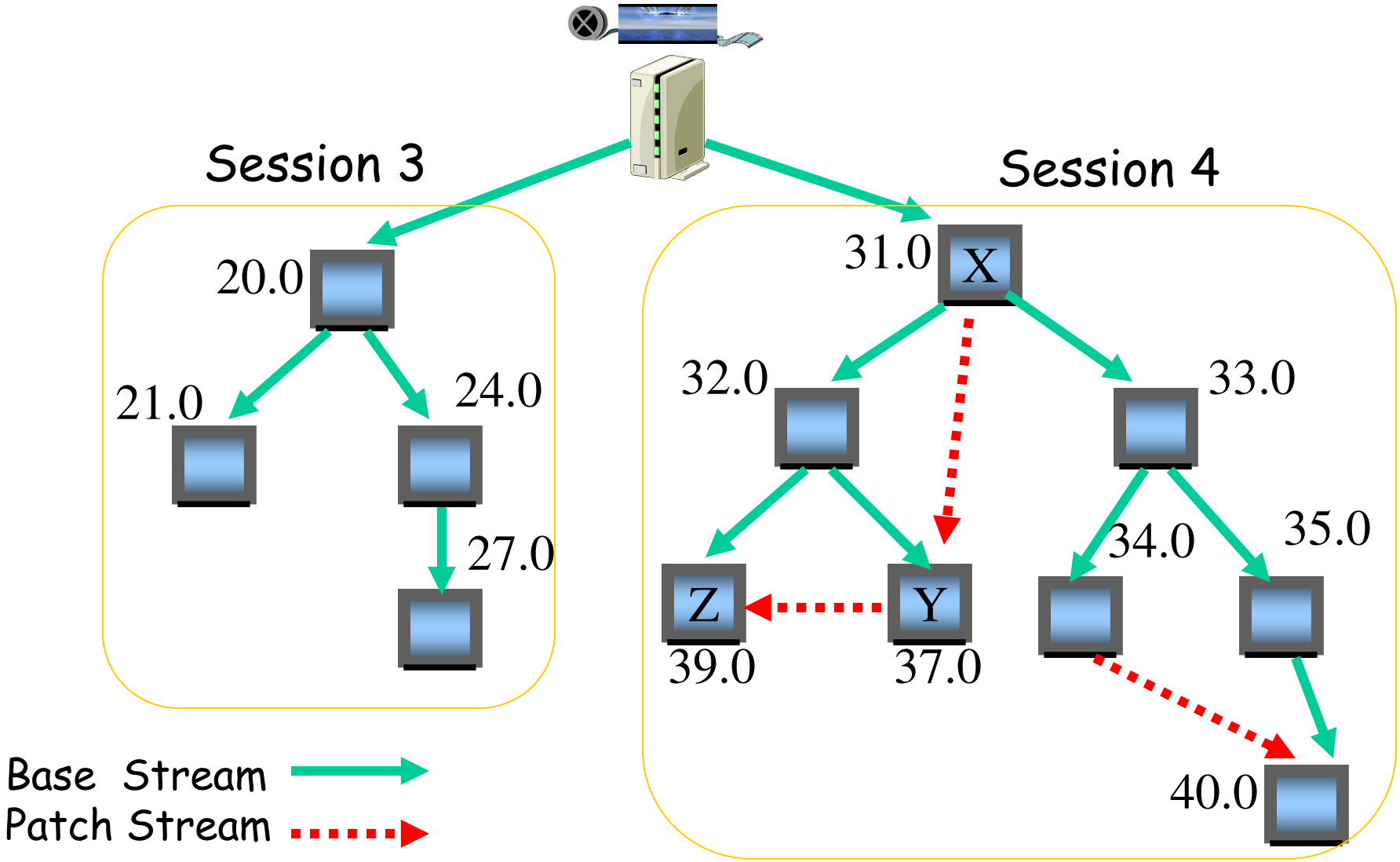


- Smaller Threshold shows better resilience to disruption
- Patch stream is relatively immune to disruption

Expected Questions

1. Why not P2P PB or P2P streaming Merging?
2. Overhead of Avail-BW measurement?
3. Why not Narada + Patching Scheme?
4. Spatial Locality Information
   - Can be extended with Clustering mechanism given locality information

A Snapshot of P2Cast at time 40

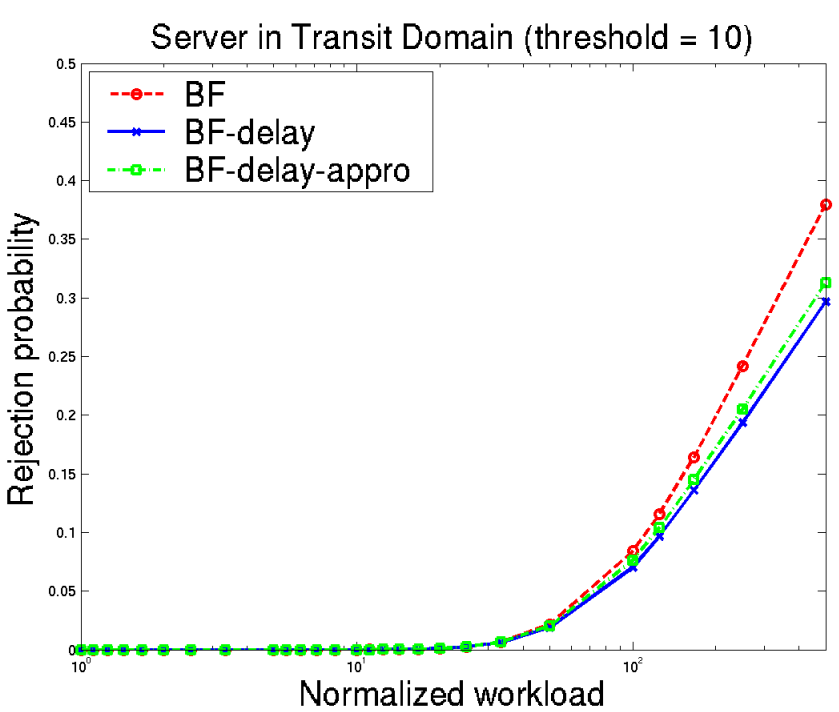


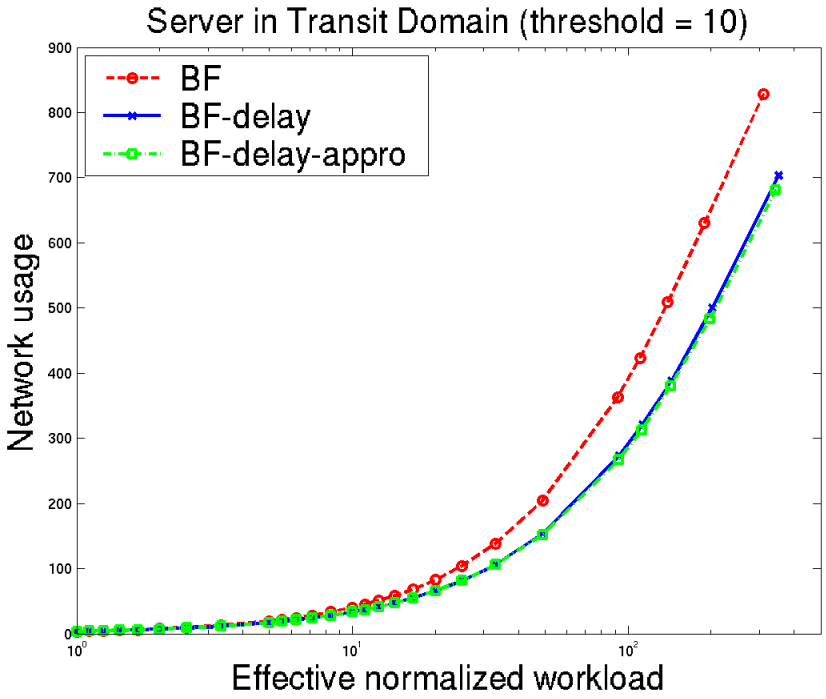
Peers in P2Cast

- Role of Peers:
  - Forward base stream to other clients
  - Cache and serve the initial part of the video to other clients
- Benefit:

  Our Experimental Results show that
  - P2Cast scales much better than traditional client-server unicast service
  - If clients can cache > **10%** of stream's initial part, it outperforms IP multicast-based patching scheme

Comparison of BF, BF-delay, BF-Delay Approx





Exploiting "Delay" information make BF-delay and BF-delay approx perform better than BF

"Best-fit (BF)" Overlay Construction algorithm

- Design Principles:
  - Bandwidth-first principle
  - Local information only principle
- Algorithm:

Step 1: Client N contacts a candidate parent P, starting with the server

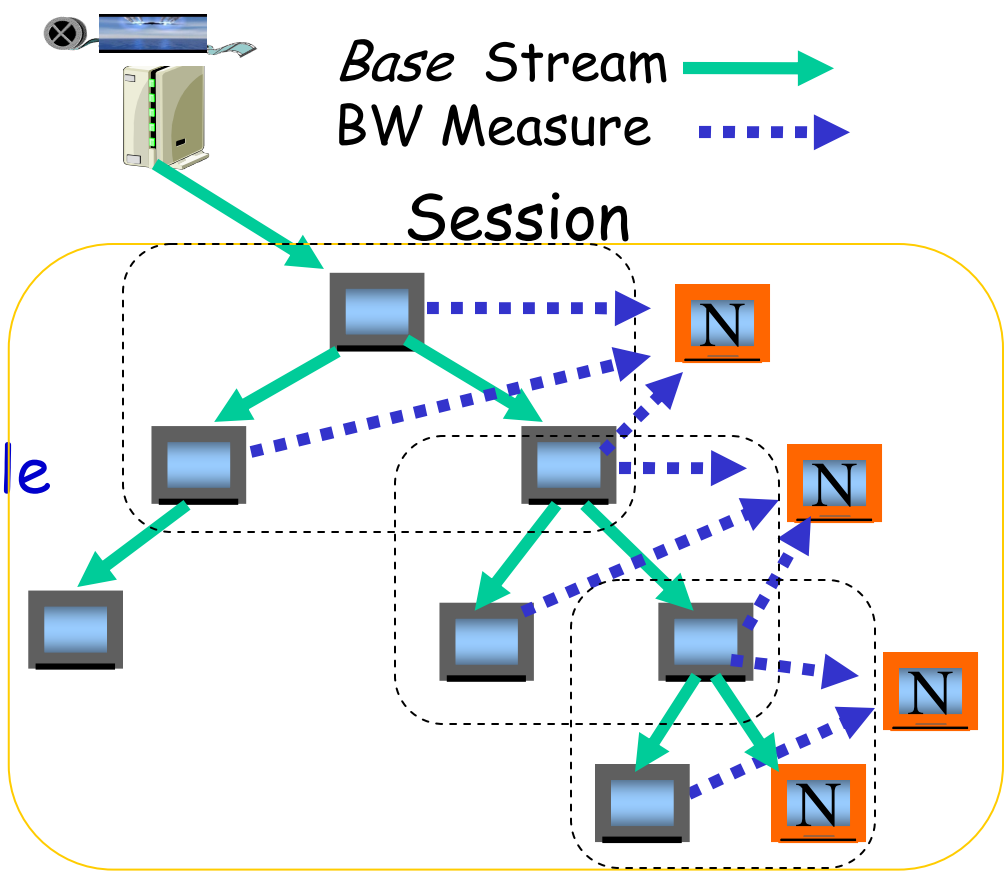
Step 2: Measure bandwidth from P to N, B(P,N) and B(C_i, N) where $C_i \in$ Children(P)

Step 3: C_{max} = argmax$_{C \in \text{Children(P)}}$ B(C,N)

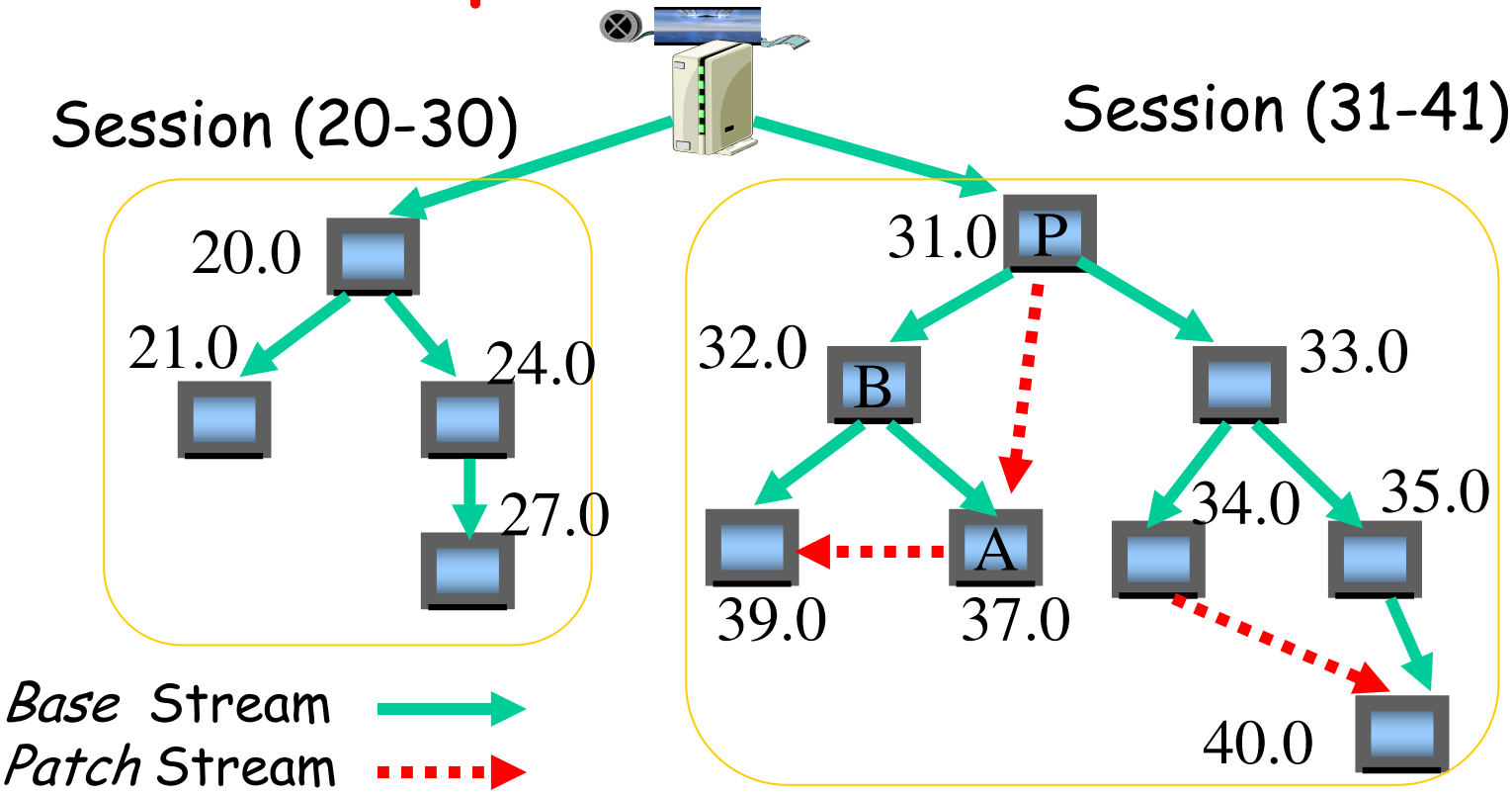
- If B(P,N) > B(C_{max},N), P provides service to N and Stop
- If B(P,N) $\leq$ B(C_{max},N), N is re-directed to C_{max} and starts from Step 1

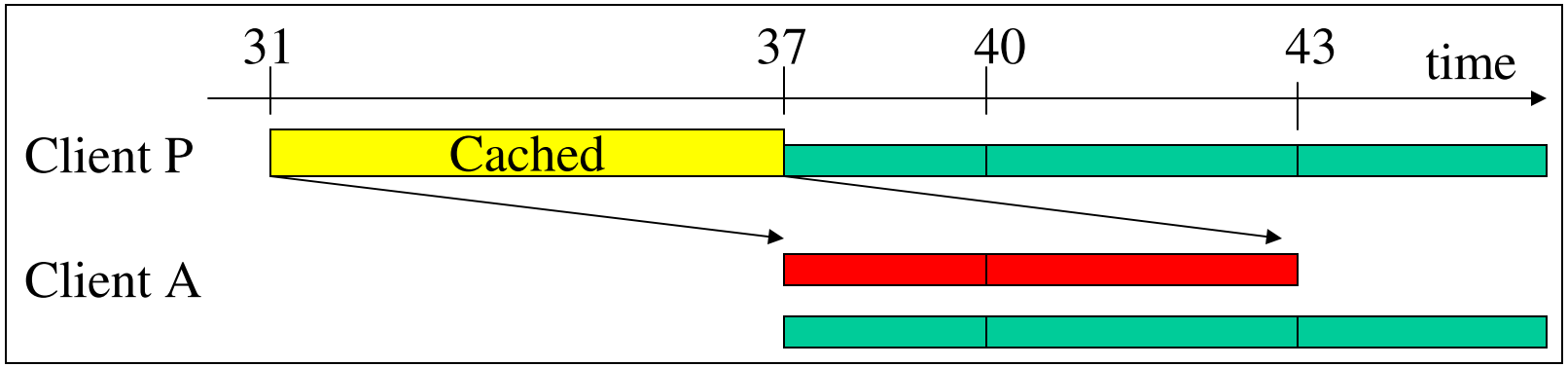
"Best-fit (BF)" Overlay Construction

- Design Principles:
  - Bandwidth (BW)-first principle
  - Local information only principle
- Idea: Try to Maximize Available BW

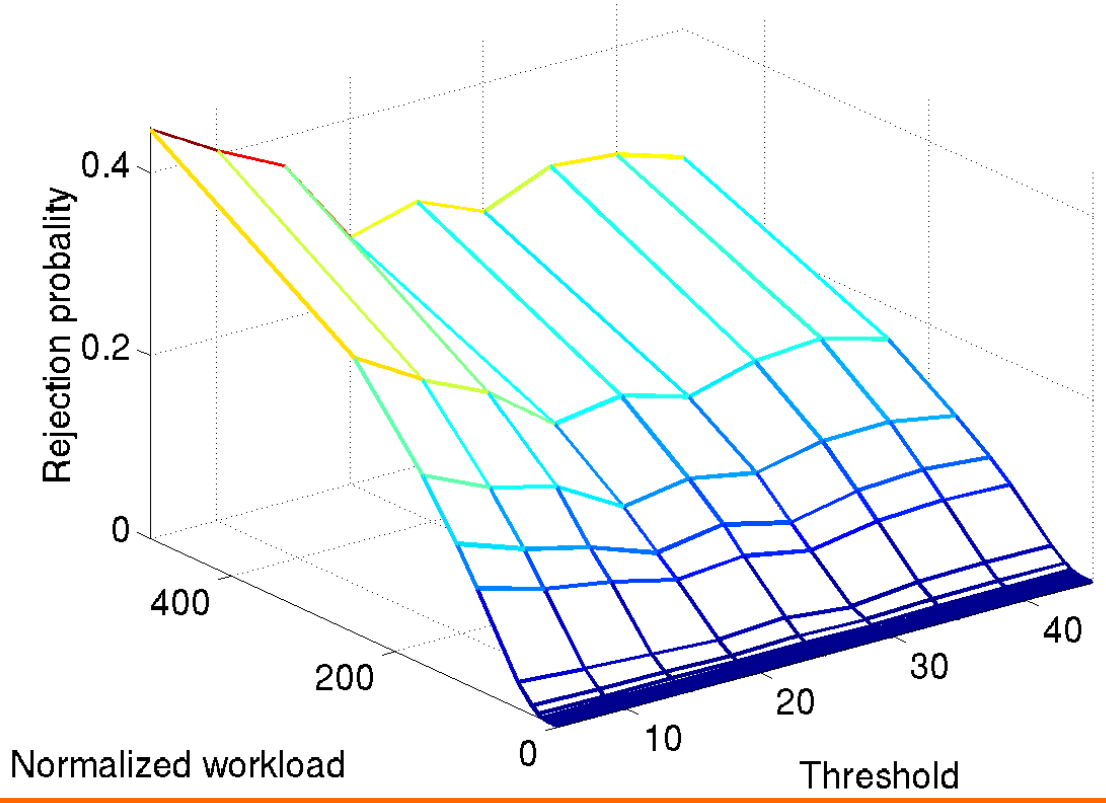


A Snapshot of P2Cast





Threshold impact on the performance of P2Cast



Larger Threshold ➔ Better Scalability

Small threshold is enough

Challenges for P2Cast

2. Providing continuous stream playback without glitch
   - Resilient to disruption from early departing clients
   - Our solution:
     - **Limit session size**
     - Delay the start of playback for patch stream
     - Shifted forwarding for base stream