Resilient Peer-to-Peer Streaming

Venkat Padmanabhan
Microsoft Research

March 2003

Collaborators and Contributors

- Joint work with
  - Phil Chou
  - Helen Wang

- Acknowledgements
  - Kay Sripanidkulchai (former intern from CMU)
  - Steve Zabinsky

Outline

- Motivation and Challenges
- CoopNet approach to resilience:
  - Path diversity: multiple distribution trees
  - Data redundancy: multiple description coding
- Performance evaluation
- Layered MDC & Congestion Control
- Related work
- Summary and ongoing work

Motivation

- Problem: support "live" streaming to a potentially large and highly dynamic population
- Motivating scenario: flash crowds
  - often due to an event of widespread interest…
  - … but not always (e.g., Webcast of a birthday party)
  - can affect relatively obscure sites (e.g., www.cricket.org)
    - site becomes unreachable precisely when it is popular!
- Streaming server can quickly be overwhelmed
  - network bandwidth is the bottleneck

Solution Alternatives

- IP multicast:
  - works well in islands (e.g., corporate intranets)
  - hindered by limited availability at the inter-domain level
- Infrastructure-based CDNs (e.g., Akamai, RBN)
  - well-engineered network $\Rightarrow$ good performance
  - but may be too expensive, even for the big sites
    - (e.g., CNN [LeFebvre 2002])
  - uninteresting for CDN to support small sites

- Goal: solve the flash crowd problem without requiring new infrastructure!

Cooperative Networking (CoopNet)

- Peer-to-peer streaming
  - clients serve content to other clients
- Not a new idea
  - much research on application-level multicast (ALMI, ESM, Scattercast)
  - some start-ups too (Allcast, vTrails)
- Main advantage: self-scaling
  - aggregate system bandwidth grows with demand
- Main disadvantage: hard to provide "guarantees"
  - P2P is not a replacement for infrastructure-based CDNs
  - but how can we improve the resilience of P2P streaming?

Challenges

- Unreliable peers
  - peers are far from being dedicated servers
  - disconnections, crashes, reboots, etc.
- Constrained and asymmetric bandwidth
  - last hop is often the bottleneck in "real-world" peers
  - median broadband bandwidth: 900 Kbps/212 Kbps (PeerMetric study: Lakshminarayanan & Padmanabhan)
  - congestion due to competing applications
- Reluctant users
  - some ISPs charge based on usage
- Others issues:
  - NATs: IETF STUN offers hope
  - Security: content integrity, privacy, DRM

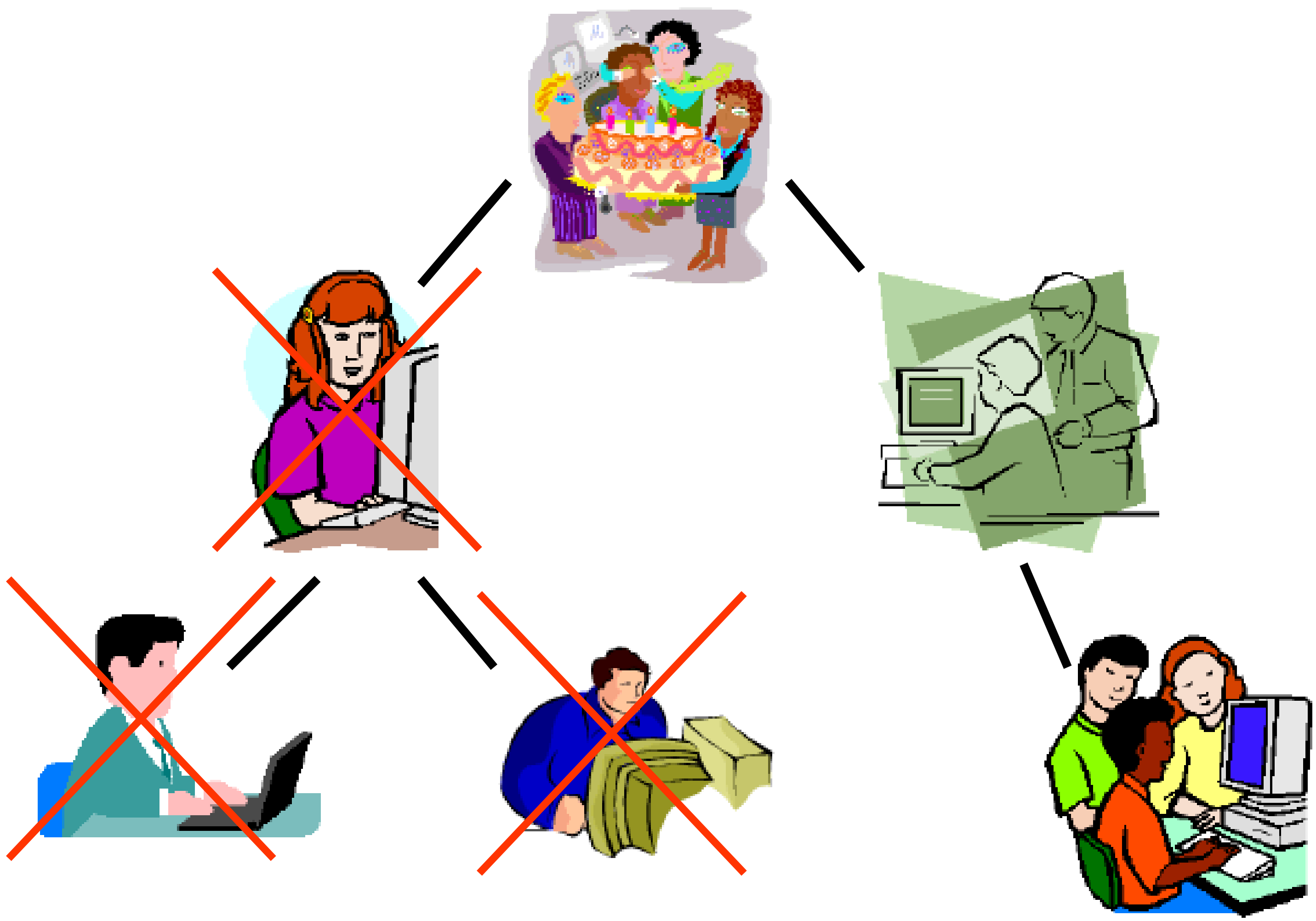
CoopNet Design Choices

- Place minimal demands on the peers
  - peer participates and forwards traffic only for as long as it is interested in the content
  - peer contributes only as much upstream bandwidth as it consumes downstream
  - natural incentive structure
    - enforcement is a hard problem!
- Resilience through redundancy
  - redundancy in network paths
  - redundancy in data

Outline

- Motivation and Challenges
- CoopNet approach to resilience:
  - Path diversity: multiple distribution trees
  - Data redundancy: multiple description coding
- Performance evaluation
- Layered MDC & Congestion Control
- Related work
- Summary and ongoing work

Traditional Application-level Multicast

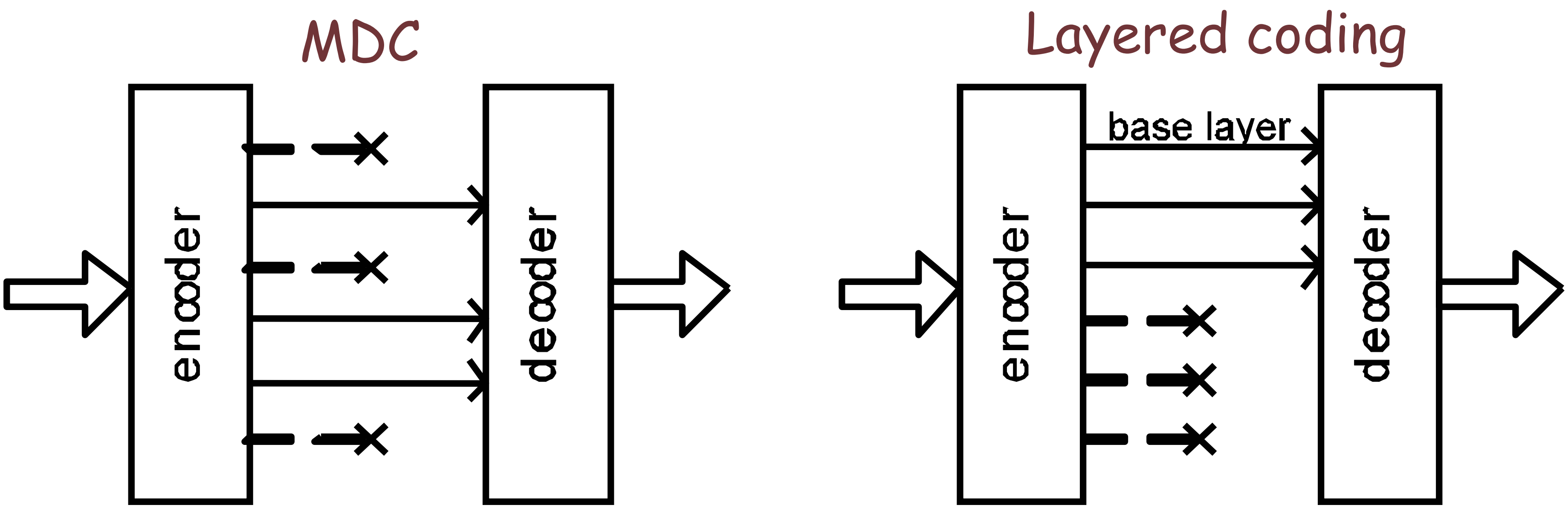


Vulnerable to node departures and failures

CoopNet Approach to Resilience

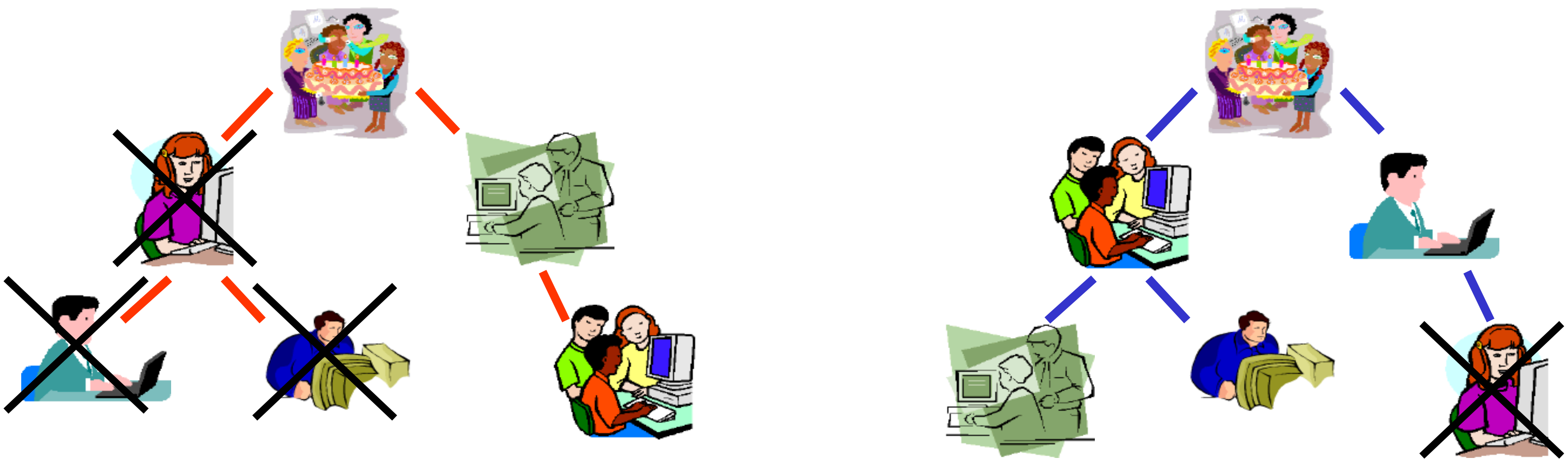
- Add redundancy in data…
  - multiple description coding (MDC)
- … and in network paths
  - multiple, diverse distribution trees

Multiple Description Coding



- Unlike layered coding, there isn't an ordering of the descriptions
- Every subset of descriptions must be decodable
- So better suited for today's best-effort Internet
- Modest penalty relative to layered coding

Multiple, Diverse Distribution Trees



Tree diversity provides robustness to node failures.

Outline

- Motivation and Challenges
- CoopNet approach to resilience:
  - Path diversity: multiple distribution trees
  - Data redundancy: multiple description coding
- Performance evaluation
- Layered MDC & Congestion Control
- Related work
- Summary and ongoing work

Tree Management Goals

- Traditional ALM goals
  - efficiency
    - make tree structure match the underlying network topology
    - mimic IP multicast?
    - optimize over time
  - scalability
    - avoid hot spots by distributing the load
  - speed
    - quick joins and leaves
- But how appropriate are these for CoopNet?
  - unreliable peers, high churn rate
  - failures likely due to peers nodes or their last-mile
  - resilience is the key issue

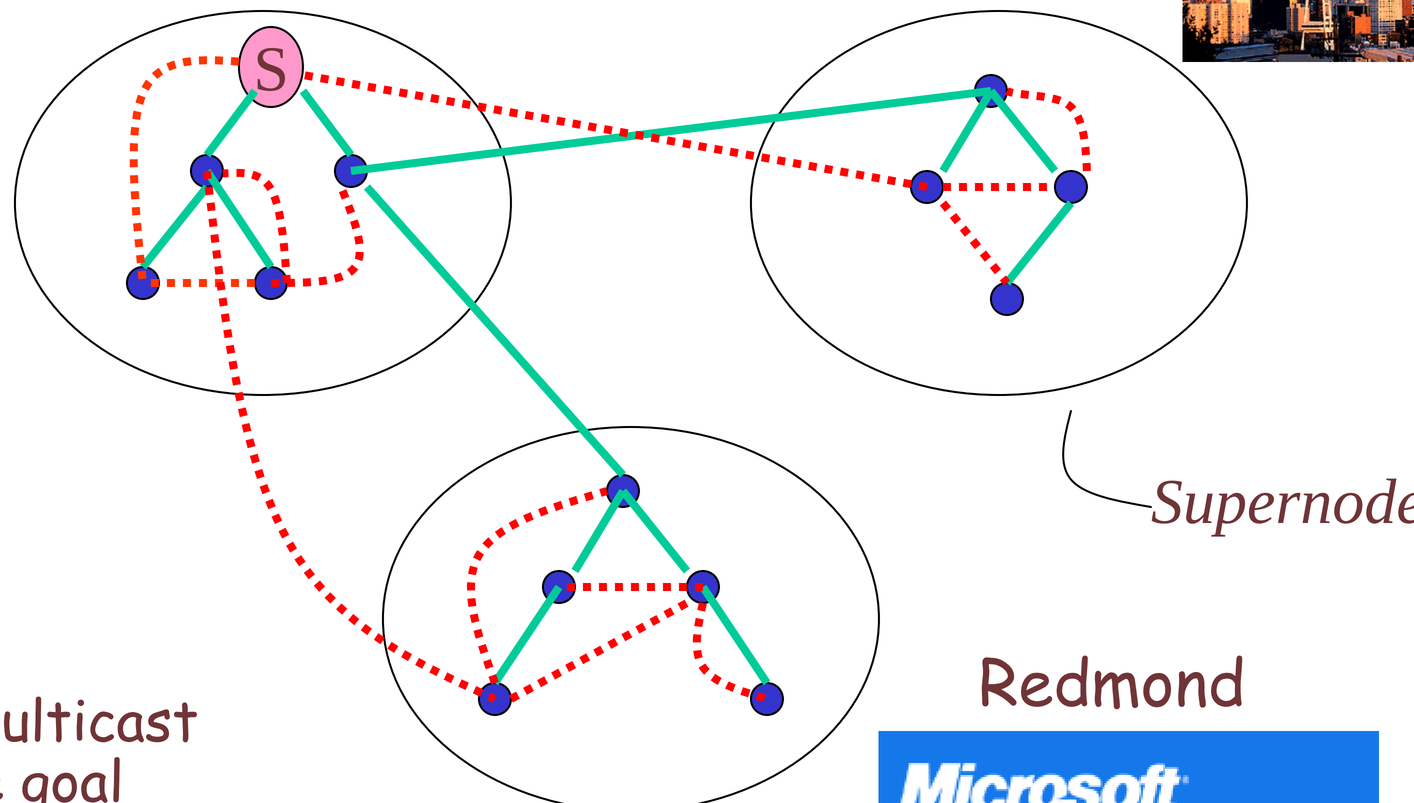
Tree Management Goals (contd.)

- Additional goals for CoopNet:
  - shortness
    - fewer ancestors $\Rightarrow$ less prone to failure
  - diversity
    - different ancestors in each tree $\Rightarrow$ robustness
- Some of the goals may be mutually conflicting
  - shortness vs. efficiency
  - diversity vs. efficiency
  - speed vs. scalability
- Our goal is resilience
  - so we focus on shortness, diversity, and speed
  - we sacrifice a little on self-scaling
  - efficiency is a secondary goal

Shortness, Diversity & Efficiency

New York

Seattle



Redmond

Mimicking IP multicast
is not the goal

CoopNet Approach

Centralized protocol anchored at the server (akin to the Napster architecture)

- Nodes inform the server when they join and leave
  - they indicate available bandwidth, delay coordinates
- Server maintains the trees
- Nodes monitor loss rate on each tree and seek new parent(s) when it gets too high
  - single mechanism to handle packet loss and ungraceful leaves

Pros and Cons

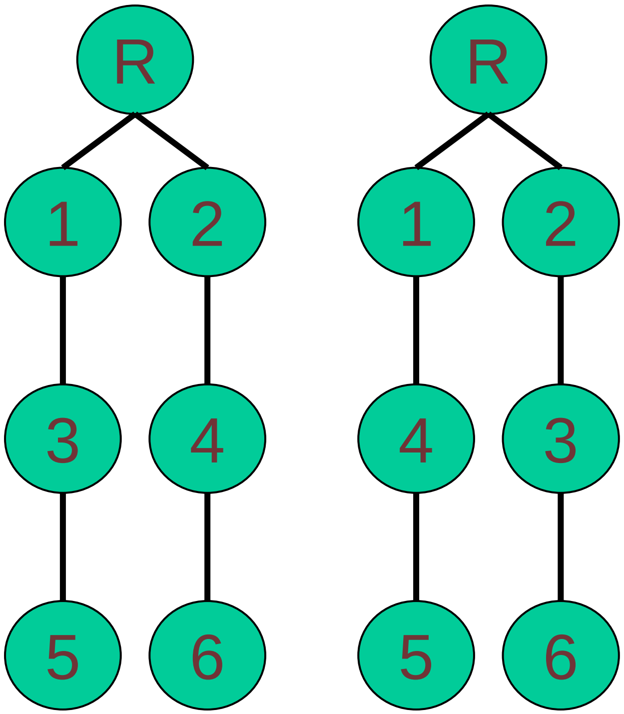
- Advantages:
  - availability of resourceful server simplifies protocol
  - quick joins/leaves: 1-2 network round-trips
- Disadvantages:
  - single point of failure
    - but server is source of data anyway
  - not self-scaling
    - but still self-scaling with respect to bandwidth
    - tree manager can keep up with ~100 joins/leaves per second on a 1.7 GHz P4 box (untuned implementation)
    - tree manager can be scaled up using a server cluster
      - CPU is the bottleneck

Randomized Tree Construction

Simple motivation: randomize to achieve diversity!

- Join processing:
  - server searches through each tree to find the highest *k* levels with room
    - need to balance shortness and diversity
    - *k* is usually small (1 or 2)
  - it randomly picks a parent from among these nodes
  - informs parents & new node
- Leave processing:
  - find new parent for each orphan node
  - orphan's subtree migrates with it
- Reported in our NOSSDAV '02 paper

Why is this suboptimal?

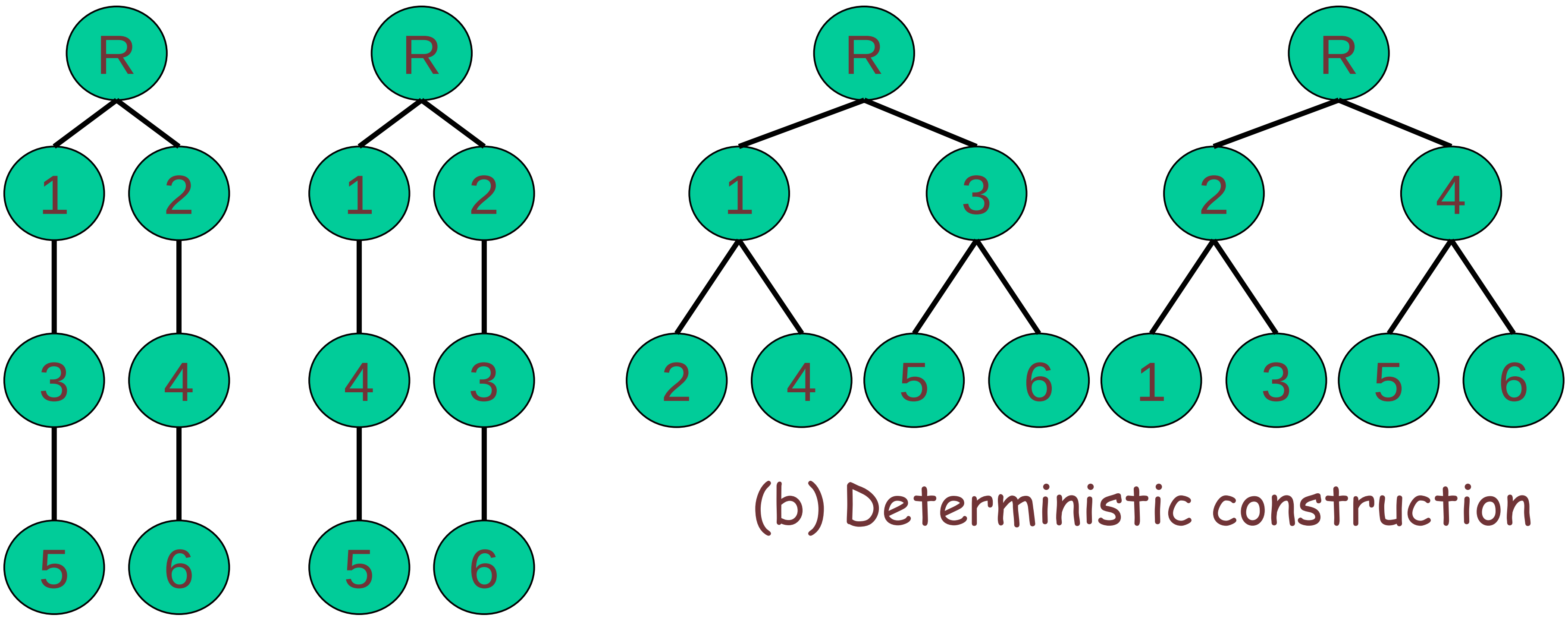


- We ask nodes to contribute only as much bandwidth as they consume
- So T trees $\Rightarrow$ each node can support at most T children in total
- Q: how should a node's out-degree be distributed?
- Randomized tree construction tends to distribute the out-degree randomly
- This results in deep trees that not very bushy

Deterministic Tree Construction

- Motivated by SplitStream work [Castro '03]
  - a node need be an interior node in just one tree
  - their motivation: bound outgoing bandwidth requirement
  - our motivation: shortness!
- Fertile nodes and sterile nodes
  - every node is fertile in one and only one tree
  - deterministically pick fertile tree for a node
  - deterministically pick parent at the highest level with room
  - may need to “migrate” fertile nodes between trees
- Diversity
  - set of ancestors are guaranteed to be disjoint
  - unclear how much it helps when multiple failures are likely

Randomized vs. Deterministic Construction



(b) Deterministic construction

(a) Randomized construction

Outline

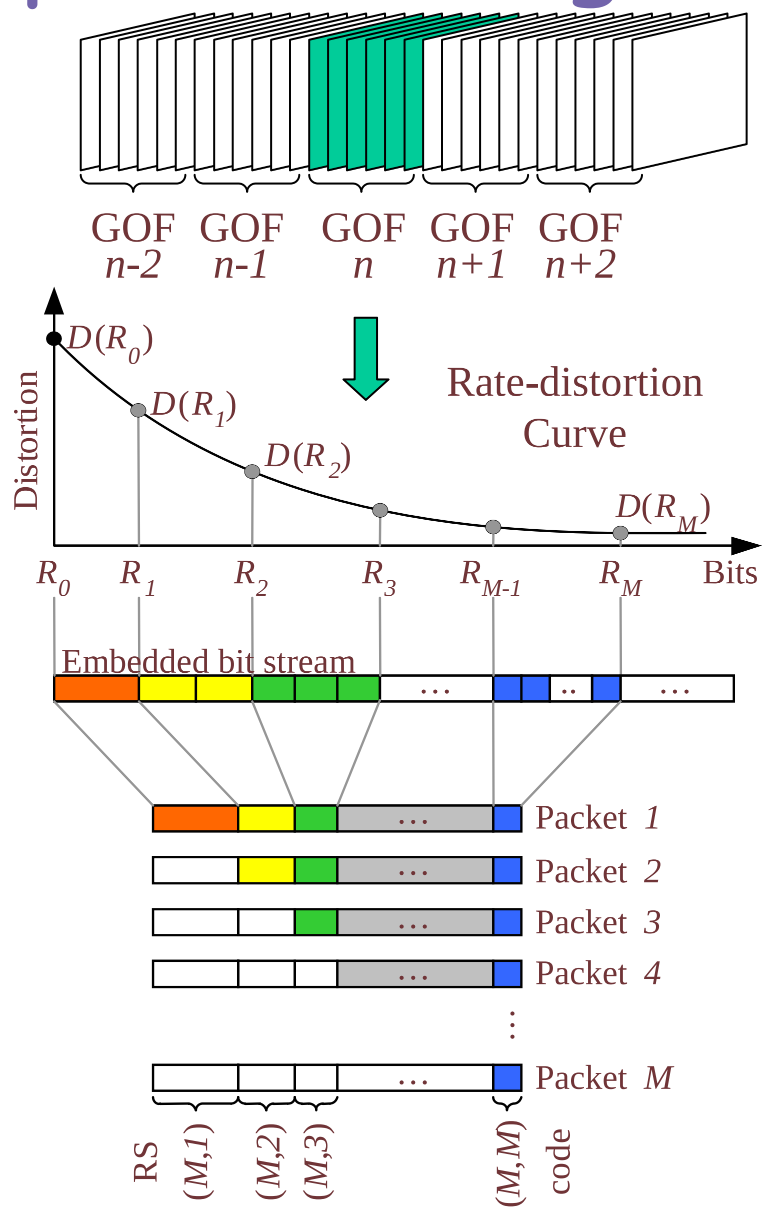
- Motivation and Challenges
- CoopNet approach to resilience:
  - Path diversity: multiple distribution trees
  - Data redundancy: multiple description coding
- Performance evaluation
- Layered MDC & Congestion Control
- Related work
- Summary and ongoing work

Multiple Description Coding

- Key point: independent descriptions
  - no ordering of the descriptions
  - any subset should be decodable
- Old idea dating back to the 1970s
  - e.g., "voice splitting" work at Bell Labs
- A simple MDC scheme for video
  - every M^{th} frame forms a description
  - makes inter-frame coding less efficient
- Can do better
  - e.g., Puri & Ramchandran '99, Mohr '00

Multiple Description Coding

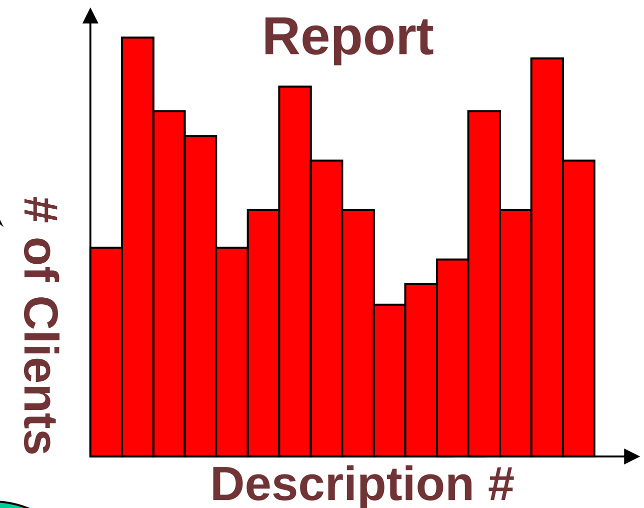
- MDC using FEC
  - Puri & Ramchandran '99
- Combine:
  - layered coding
  - Reed-Solomon coding
  - priority encoded transmission
  - optimized bit allocation
- Easy to generate if the input stream is layered
- M = R*G/P
- Adapt rate-points based on loss distribution



Scalable Feedback

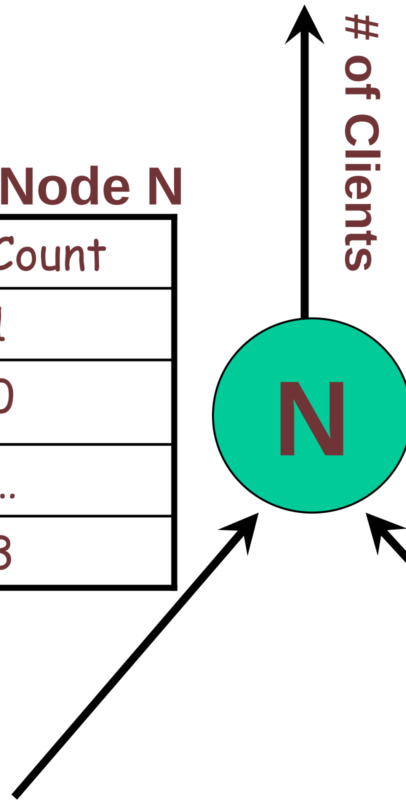
- Optimize rate points based on loss distribution
  - source needs to know p(m) distribution
  - individual reports from each node might overwhelm the source
- Scalable feedback
  - a small number of trees are designated to carry feedback
  - each node maintains a local h(m) histogram
  - the node adds up histograms received from its children…
  - …and periodically passes on the composite histogram for the subtree to its parent
  - the root (source) then computes p(m) for the entire group

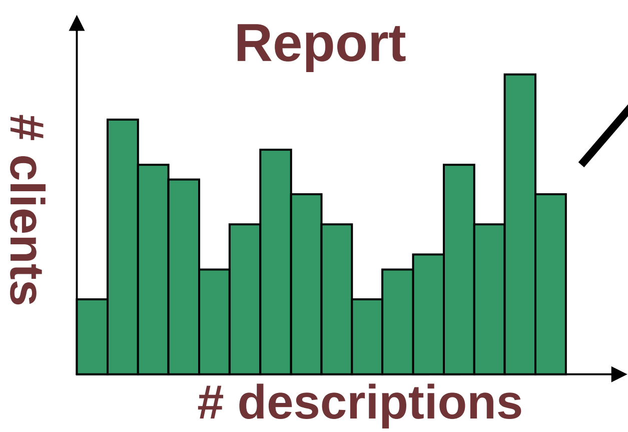
Scalable Feedback

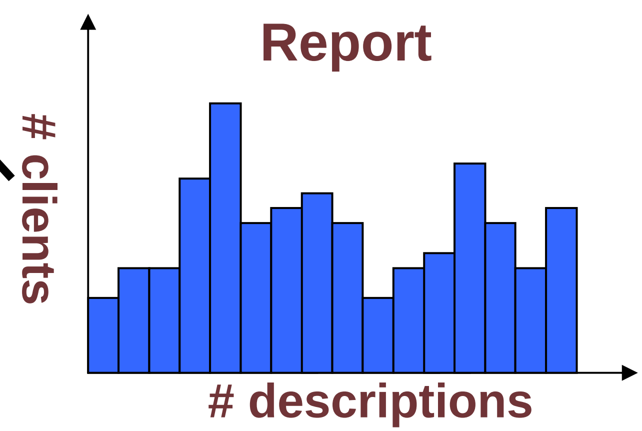


Record at Node N

# Desc	Count
0	1
1	0
…	…
16	3







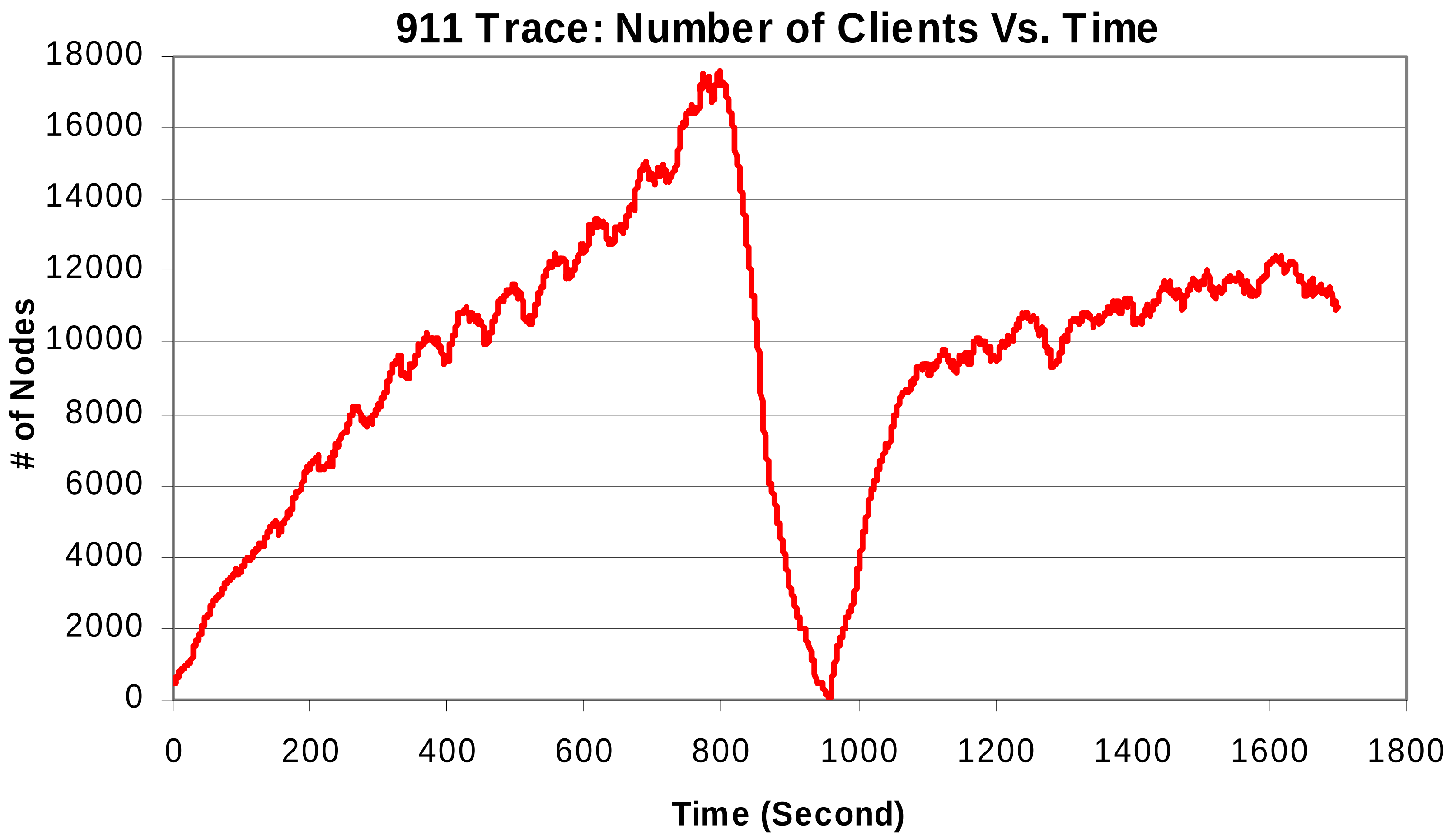
Outline

- Motivation and Challenges
- CoopNet approach to resilience:
  - Path diversity: multiple distribution trees
  - Data redundancy: multiple description coding
- Performance evaluation
- Layered MDC & Congestion Control
- Related work
- Summary and ongoing work

Flash Crowd Traces

- MSNBC streaming logs from Sep 11, 2001
  - join time and session duration
  - assumption: session termination ⇒ node stops participating
- Live streaming: 100 Kbps Windows Media Stream
  - up to ~18,000 simultaneous clients
  - ~180 joins/leaves per second on average
  - peak rate of ~1000 per second
  - ~70% of clients tuned in for less than a minute
    - high churn possibly because of flash crowd congestion

Flash Crowd Dynamics



Simulation Parameters

Server bandwidth: 20 Mbps
Peer bandwidth: 160 Kbps
Stream bandwidth: 160 Kbps
Packet size: 1250 bytes
GOF duration: 1 second
descriptions: 16
trees: 1, 2, 4, 8, 16
Repair interval: 1, 5, 10 seconds

Video Data

Akiyo

Foreman

Stefan

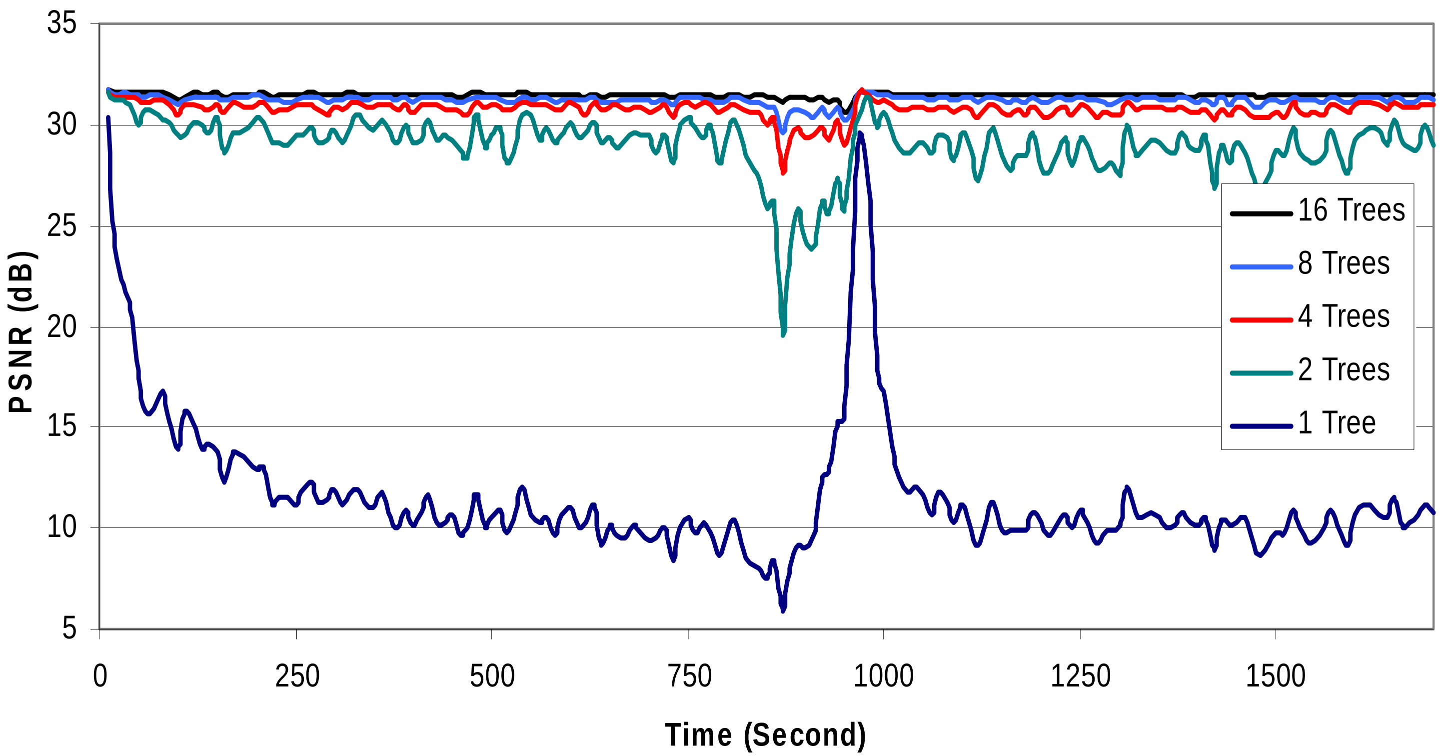
- We don't have the actual MSNBC video content
- Standard MPEG test sequences (10 seconds each)
- QCIF (176x144), 10 frames per second

Questions

- Benefits of multiple, diverse trees
- Randomized vs. deterministic tree construction
- Variation across the 3 video clips
- MDC vs. pure FEC
- Redundancy introduced by MDC
- Impact of repair time
- Impact of network packet loss

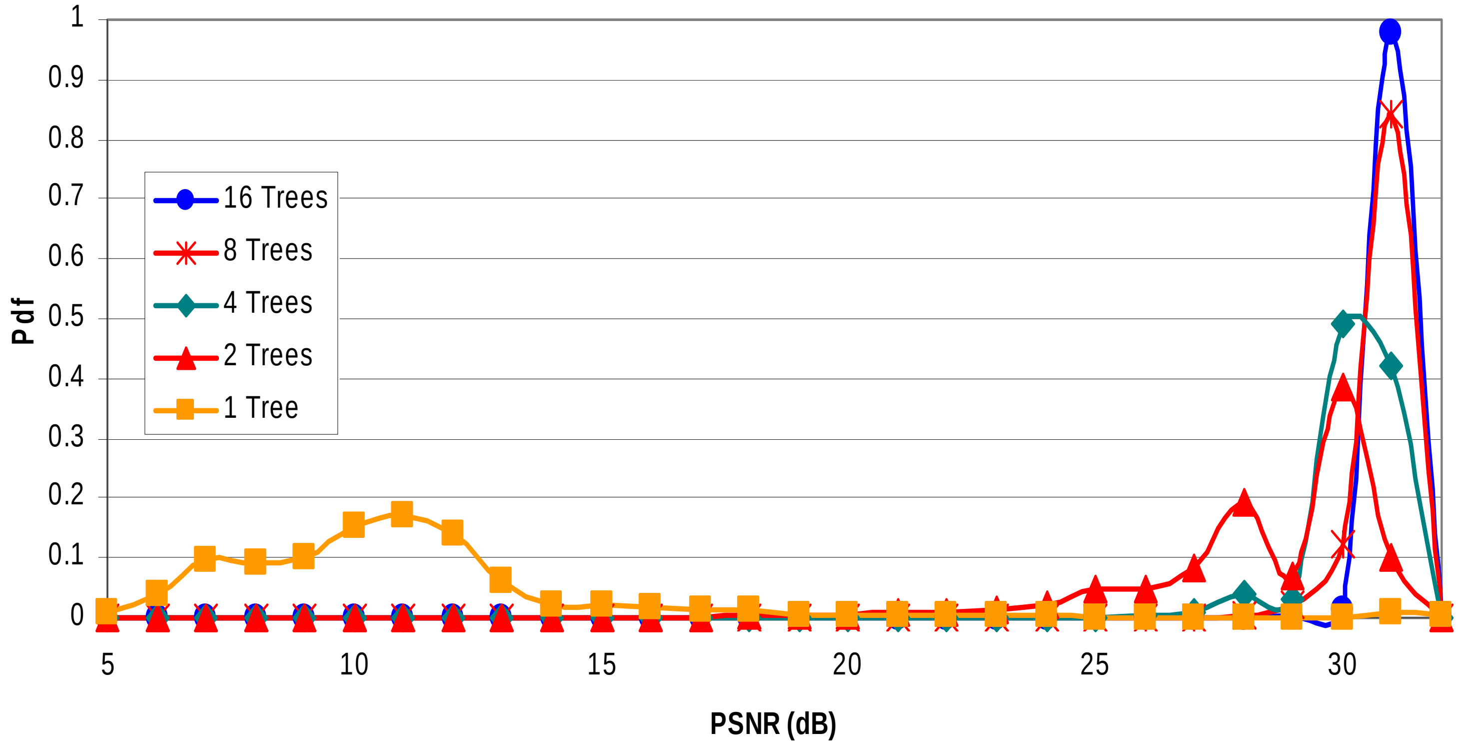
- What does it look like?

Impact of Number of Trees

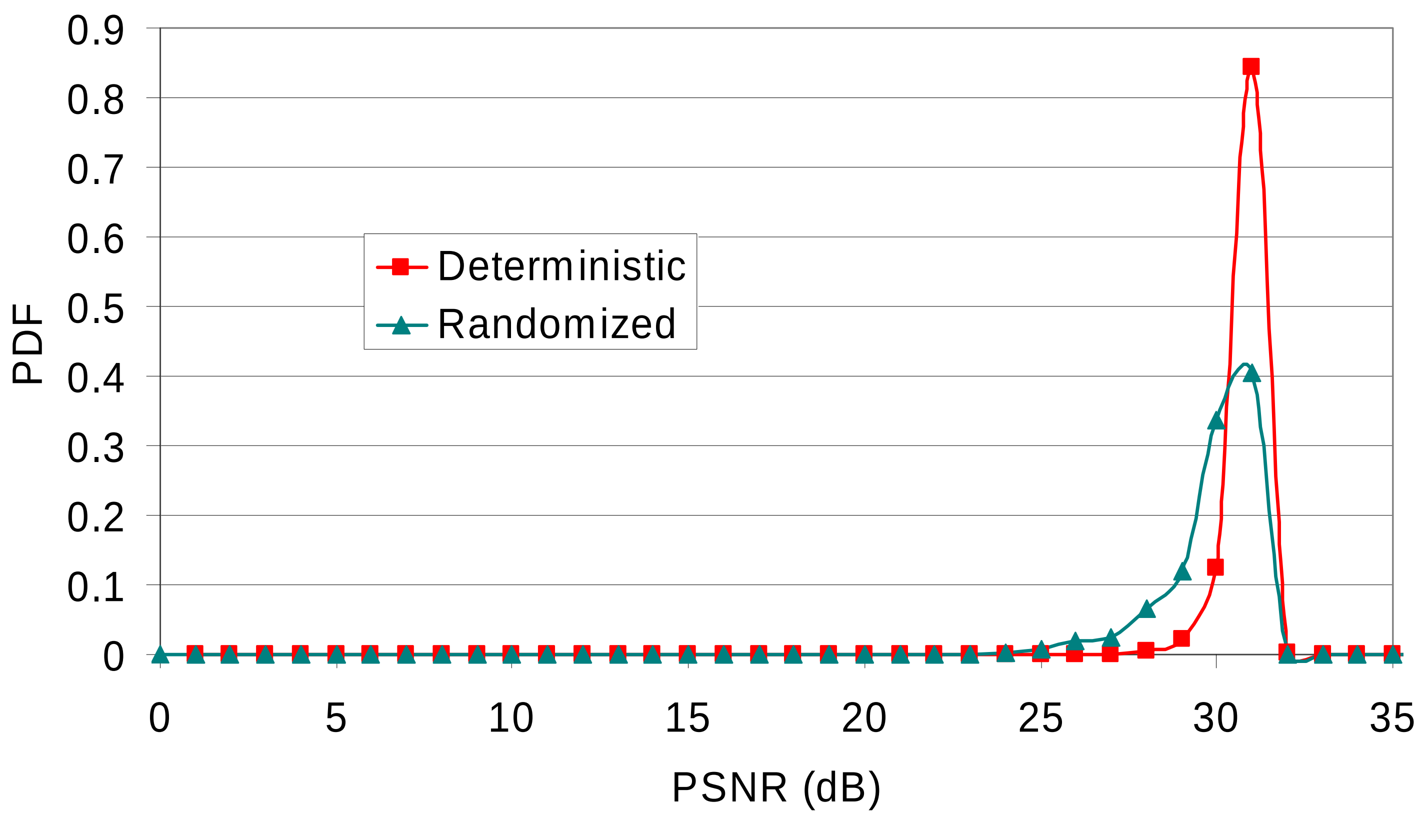


Multiple, diverse trees help significantly.
Much of the benefit is achieved with 8 trees.

Impact of Number of Trees

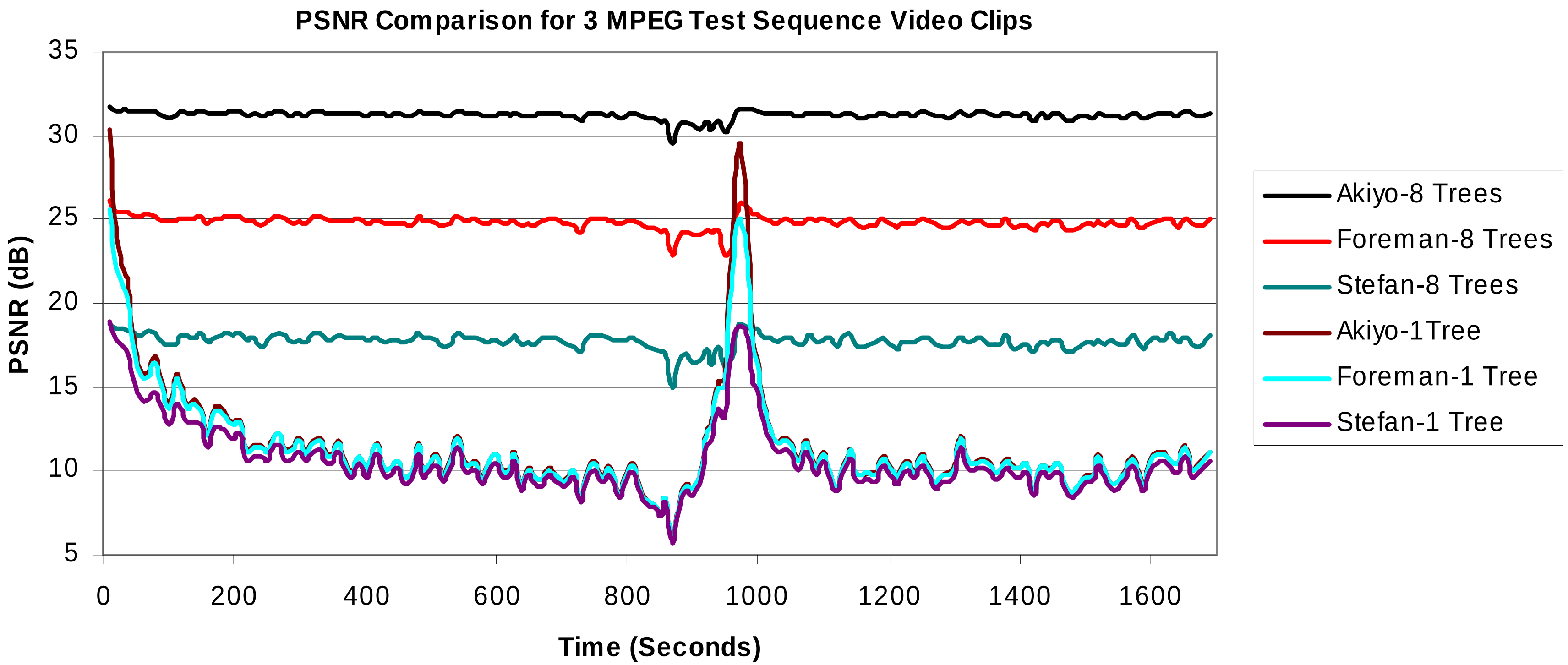


Randomized vs. Deterministic Tree Construction



Deterministic algorithm results in shorter trees that are less prone to disruption

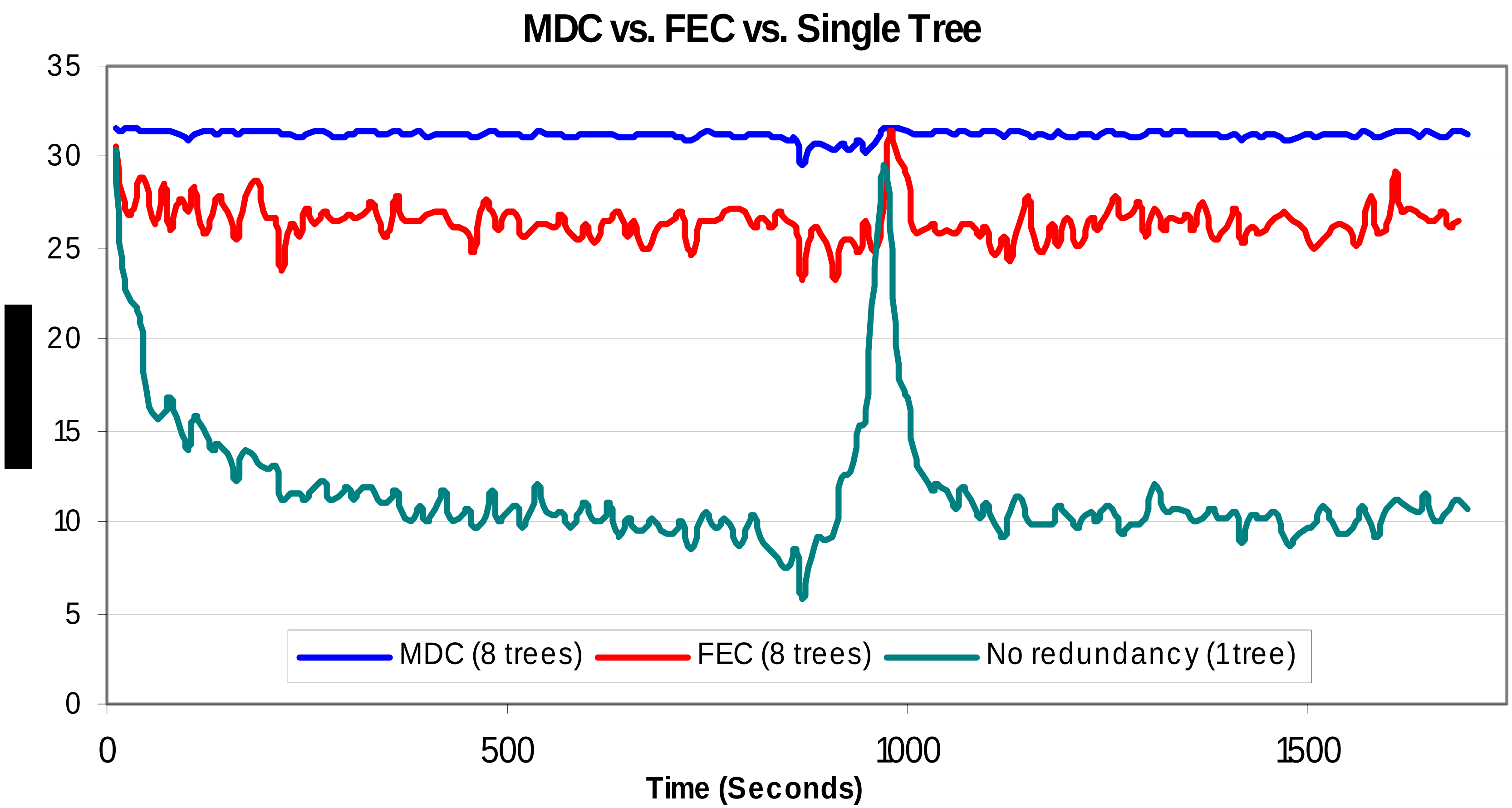
Comparison of Video Clips



Clips with high motion suffer worse quality.

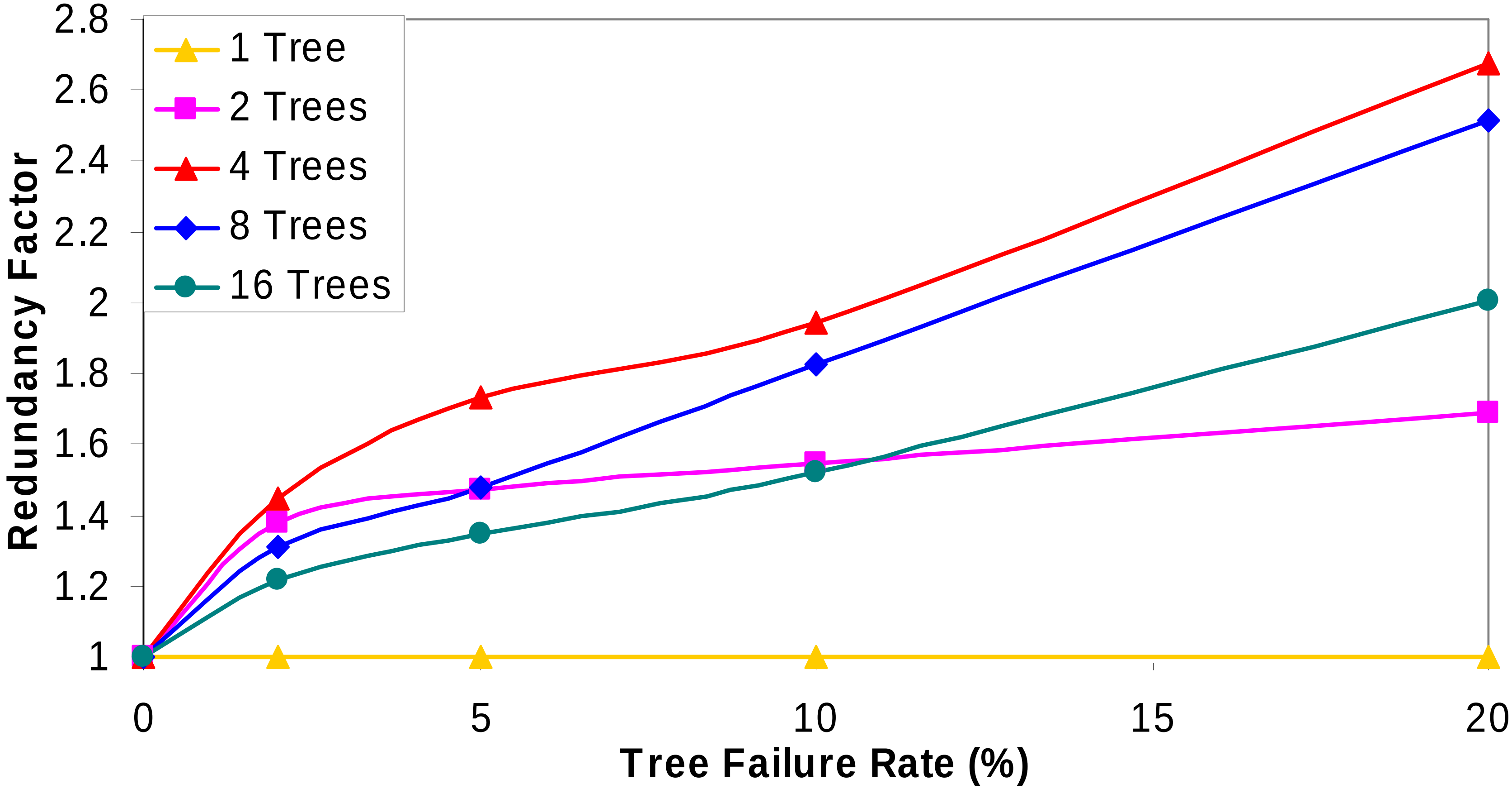
But CoopNet helps in all cases.

MDC vs. Pure FEC



MDC is better able to adapt to a wide spatial distribution in packet loss than pure FEC.

Redundancy vs. Tree Failure Rate



Amount of redundancy decreases with more trees because loss of many descriptions becomes less likely

What it looks like

Single-tree Distribution

CoopNet Distribution with FEC (8 trees)

CoopNet Distribution with MDC (8 trees)

Outline

- Motivation and Challenges
- CoopNet approach to resilience:
  - Path diversity: multiple distribution trees
  - Data redundancy: multiple description coding
- Performance evaluation
- Layered MDC & Congestion Control
- Related work
- Summary and ongoing work

Heterogeneity & Congestion Control

- Motivated by RLM [McCanne '96]
- Layered MDC
  - base layer descriptions and enhancement layer descriptions
  - forthcoming paper at Packet Video 2003
- Congestion response depends on location of problem
- Key questions:
  - how to tell where congestion is happening?
  - how to pick children to shed?
  - how to pick parents to shed?
- Tree diversity + layered MDC can help
  - infer location of congestion from loss distribution
  - parent-driven dropping: shed enhancement-layer children
  - child-driven dropping: shed enhancement-layer parent in sterile tree

Related Work

- Application-level multicast
  - ALMI [Pendarakis '01], Narada [Chu '00], Scattercast [Chawathe'00]
    - small-scale, highly optimized
  - Bayeux [Zhuang '01], Scribe [Castro '02]
    - P2P DHT-based
    - nodes may have to forward traffic they are not interested in
    - performance under high rate of node churn?
  - SplitStream [Castro '03]
    - layered on top of Scribe
    - interior node in exactly one tree $\Rightarrow$ bounded bandwidth usage
- Infrastructure-based CDNs
  - Akamai, Real Broadcast Network, Yahoo Platinum
  - well-engineered network but for a price
- P2P CDNs
  - Allcast, vTrails

Related Work (Contd.)

- Coding and multi-path content delivery
  - Digital Fountain [Byers '98]
    - focus on file transfers
    - repeated transmissions not suitable for live streaming
  - Parallel downloads [Byers '02]
    - take advantage of lateral bandwidth
    - focus on speed rather than resilience
  - MDC for on-demand streaming in CDNs [Apostolopoulos '02]
    - what if last-mile to the client is the bottleneck?
  - Integrated source coding & congestion control [Lee '00]

Summary

- P2P streaming is attractive because it has the potential of being self-scaling
- Resilience to peer failures, departures, disconnections is a key concern

- CoopNet approach:
  - minimal demands placed on the peers
  - redundancy for resilience
    - multiple, diverse distribution trees
    - multiple description coding

Ongoing and Future Work

- Layered MDC
- Congestion control framework
- On-demand streaming

- More info: research.microsoft.com/projects/coopnet/
- Includes papers on:
  - case for P2P streaming: NOSSDAV '02
  - layered MDC: Packet Video '03
  - resilient P2P streaming: MSR Tech. Report
  - P2P Web content distribution: IPTPS '02