PURDUE
UNIVERSITY

PROMISE: Peer-to-Peer Media Streaming Using CollectCast

Mohamed Hefeeda[1]

Joint work with

Ahsan Habib[2], Boyan Botev[1], Dongyan Xu[1], Bharat Bhargava[1]

[1]Department of Computer Sciences, Purdue University

[2]School of Information and Management Systems, UC Berkeley

Support: NSF

Motivations

- **Peer-to-Peer (P2P) systems gained much attention in recent years**
  - **File sharing, CFS, distributed processing, streaming**
- **Peers characterized as** **[Saroiu, *et al.* 02]**
  - Highly diverse
  - Dynamic
  - Have limited capacity, reliability
- **Problem**
  - How to select and coordinate multiple peers to render the best possible quality streaming?

Motivations (cont'd)

- **Previous work either**
  - **Assume one sender, e.g.,** **[Tran, *et al.* 03] [Bawa, *et al.* 02]**
    - Ignores peer limited capacity
  - **Or, multiple senders but no careful selection, e.g.,** **[Padmanabhan, *et al.* 02] [Nguyen & Zakhor 02]**
    - Ignores peer diversity and network conditions
- **Our Solution**
  - **CollectCast**
  - **PROMISE**

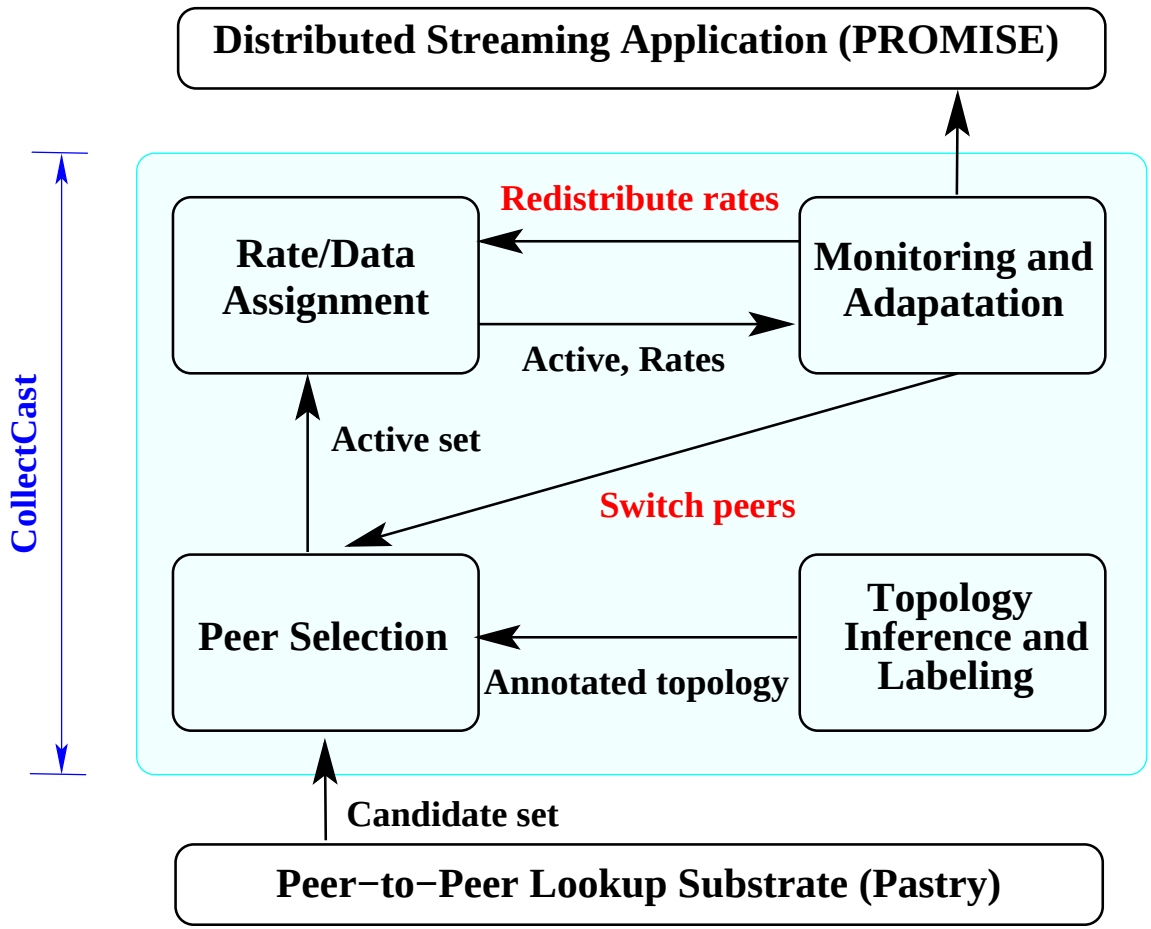
Outline

- Overview of CollectCast
- Peer model
- Peer selection
- Topology inference and labeling
- Simulations
- PROMISE and experiments on PlanetLab
- Conclusion and future work

CollectCast

- **CollectCast is a new P2P service**
  - Middleware layer between a P2P lookup substrate and applications
  - Collects data from multiple senders
- **Functions**
  - Infer and label topology
  - Select best sending peers for each session
  - Aggregate and coordinate contributions from peers
  - Adapt to peer failures and network conditions

CollectCast (cont'd)



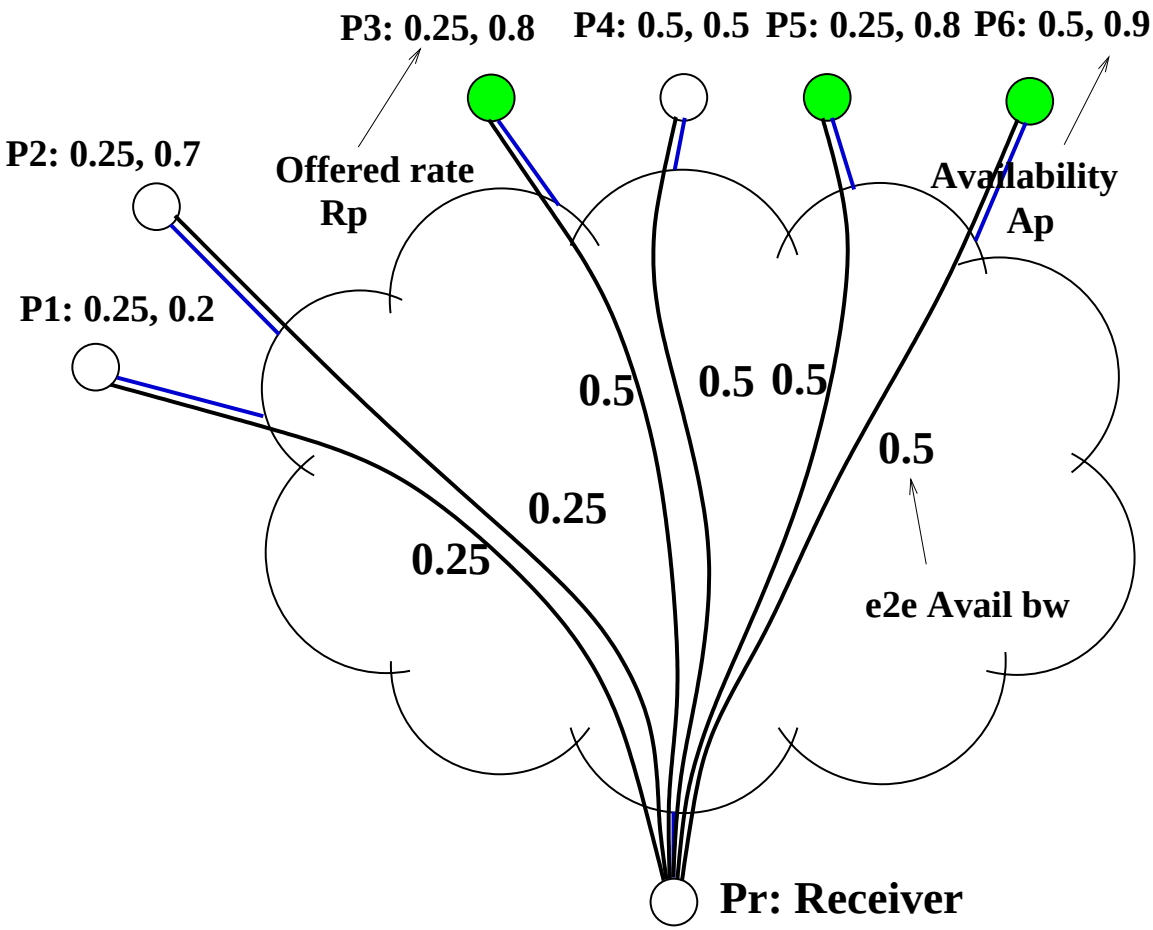
Peer Model

- **Peers are…**
  - **Heterogeneous, limited in capacity, failure-prone**
- **Peer model**
  - **Offered rate** $R_p < R_0$
    - Maximum rate peer *p* can (or is willing) to contribute
    - Captures heterogeneity and limited capacity
  - **Availability** $A_p(t)$
    - The fraction of time peer *p* is available for streaming
    - Captures reliability
    - A collection of random variables (stochastic process)

Peer Selection

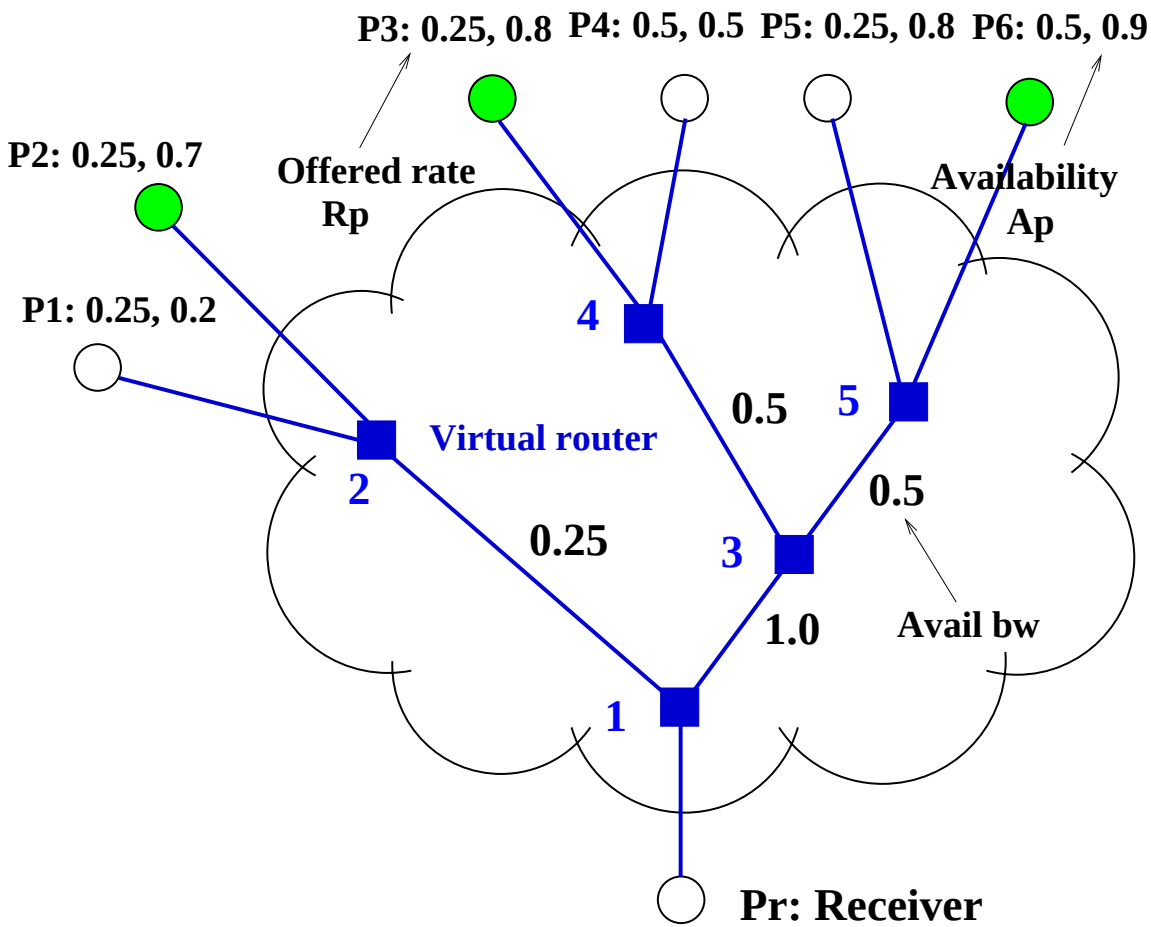
- Given a set of candidate peers, select sending peers
- Three approaches
  - Random Selection
  - End-to-End Selection
  - Topology-Aware Selection (used in CollectCast)

Peer Selection: End-to-End



- Considers: R_p, $A_p(t)$ and e2e available bandwidth and loss rate
- Ignores: Shared path segments

Peer Selection: Topology-Aware



- Considers: R_p, $A_p(t)$, e2e available bandwidth and loss rate, and Shared path segments

Topology-Aware Selection (cont'd)

- **Goodness Topology**
  - **Directed graph that interconnects candidate peers and receiving peer**
  - **Edge ≡ one or more links with no branching points (we call it path *segment*)**
  - **Each segment is labeled with a quality or *goodness* metric**
  - **Built in two steps**
    - Network tomography techniques are used to infer and label topology with loss rate and available bandwidth
    - Transform network metrics to a combined logical goodness metric

Topology-Aware Selection (cont'd)

- **Assume we have an inferred topology with loss rate and available bandwidth (later, we discuss how to get that)**
- **We define segment goodness as:**

$$_{i\to j} = w_{i\to j} \quad _{i\to j}$$

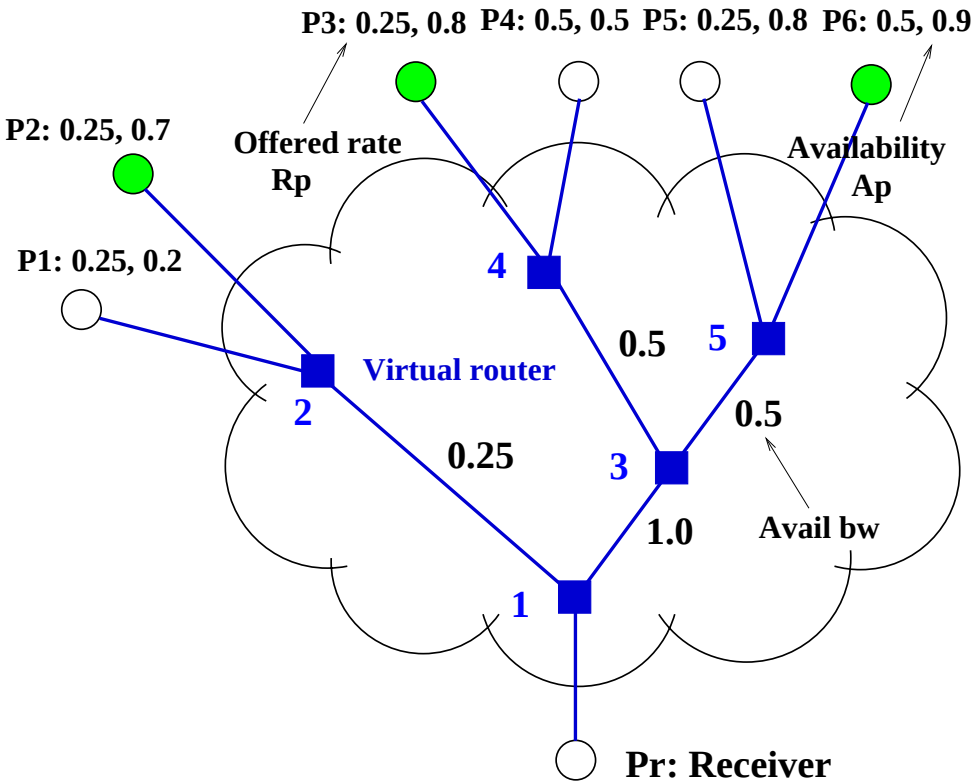
- ***w*: *weight* based on available bandwidth and level of sharing**
- **x: binary random variable that depends on loss rate:**

$$_{i\to j} = \begin{cases} 1, & \text{if a packet is } \mathbf{not} \text{ lost on } i \to j \\ 0, & \text{otherwise} \end{cases}$$

Topology-Aware Selection (cont'd)

- **Segment weight is a *per-peer* metric**

$$w_{i\to j}^{(p)} = \min\left(1,\ \max\left(0,\ \frac{b_{i\to j} - \sum_{s\in S,\ i\to j\in s\Rightarrow r} R_s}{R_p}\right)\right)$$



- **Example**
  - Consider segment 5->3
  - P6 ➔ w = 1
  - P5 ➔ w = 0

Topology-Aware Selection (cont'd)

- **Peer goodness: How good this peer is for the session**

$$_{p} = \; _{p} \prod_{i \to j \in p \Rightarrow r} {}_{i \to j} = \; _{p} \prod_{i \to j \in p \Rightarrow r} w_{i \to j}^{(p)} \; {}_{i \to j}$$

- **Active Peer Selection Problem:**

Given the goodness topology, find the set of active peers that **maximizes** ***the expected aggregate rate at the receiver, provided that the receiver in-bound bandwidth is not exceeded***

Topology-Aware Selection (cont'd)

- **Mathematically, find P^{actv} that**

$$\text{Maximizes} \quad E\left[\sum_{p \in P^{actv}} {}_pR_p\right]$$

$$\text{Subject to} \quad R_l \leq \sum_{p \in P^{actv}} R_p \leq R_u$$

- **Given this formulation, a simple iterative algorithm finds the best active set**

Topology Inference

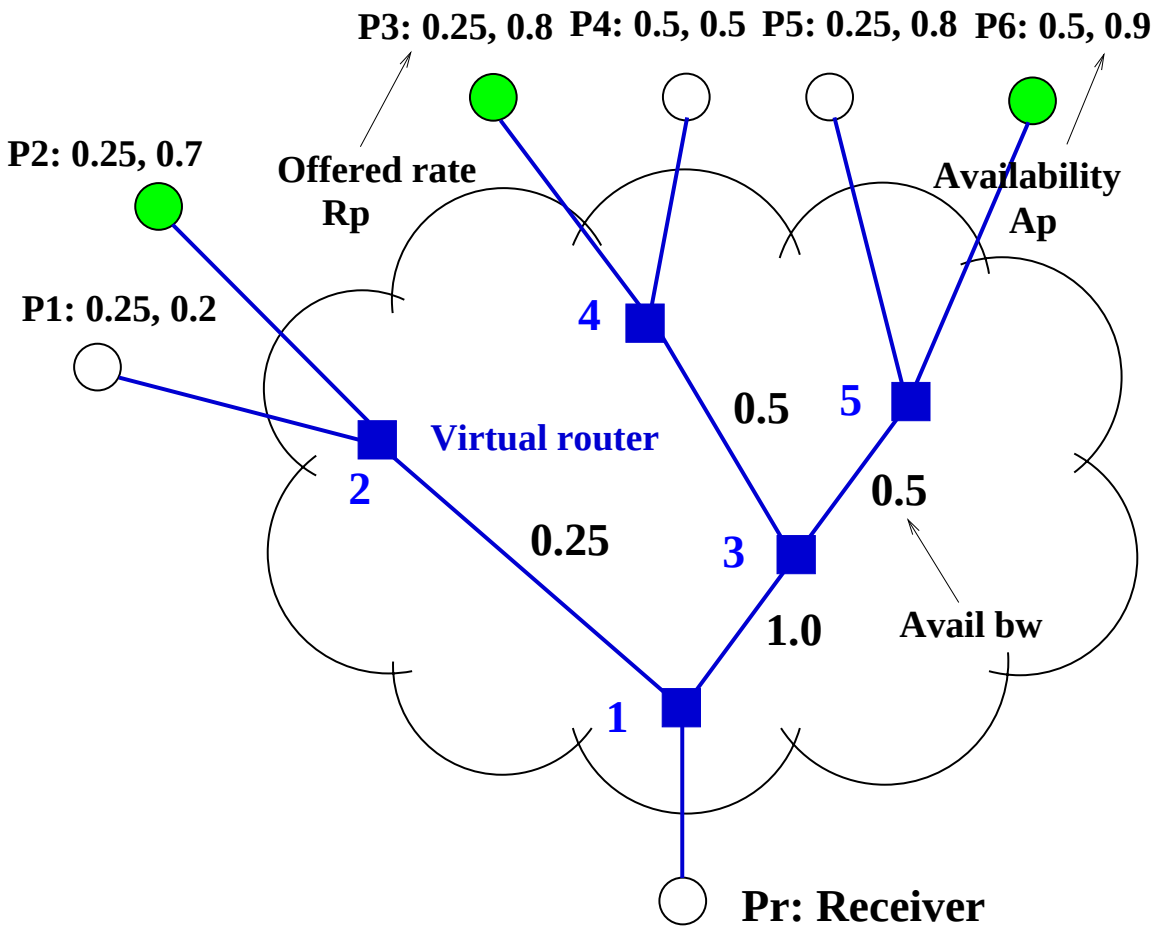
- **Network Tomography**
  - **Infer *internal* network characteristics from e2e probing**
    [Coates, *et al.*, 02], [Bestavros, *et al.* 02], [Harfoush, *et al.* 03]
  - **Premise in literature**
    - Applications may achieve significant performance gain
    - Few applications make use of it
    - Why? Techniques are generic and quite expensive
  - **Our contribution**
    - Adapt some of them to problem in hand
    - Show a concrete example for the potential benefits
  - **CollectCast is orthogonal to inference techniques**
    - Few years later ➔ better techniques
    - CollectCast is ready!

Topology Inference (cont'd)

- **Measuring available bandwidth**
  - **Basic technique** **[Jain & Dovrolis 02]**
    - End-to-end path available bandwidth (not segment-wise)
    - Idea: one-way delay differences of a periodic packet stream is a good indication for the available bandwidth
  - **Our approach**
    - Not interested in the "exact" bandwidth, rather
    - Can a path accommodate the aggregate rate from peers?
    - One peer may not be able to send at R_0, *coordinate* multiple of them to do the task. It's a P2P world!!
    - Conservatively mark all segments with the min avail bw
    - Send *real data* (from the movie) as probes!
    - Trade-off unneeded accuracy with much less overhead

Topology Inference: Example

- Let us estimate avail bw metric on segment 5→3



Topology Inference: Loss Rates

- **We already have them e2e**
  - **During avail bw measurements, record lost packets**
  - **We know data packets that are supposed to be sent**
- **Segment-wise loss rates**
  - **Passive network tomography** **[Padmanabhan, *et al.* 03]**
  - **Think of it as a *system identification* problem**
  - **Use ideas from image processing (restoration) field**
    - Bayesian inference using Gibbs sampling
      - Assume initial distribution
      - Use measured data and initial distribution to compute *posterior* distribution
      - Iterate

Topology Inference: Overhead

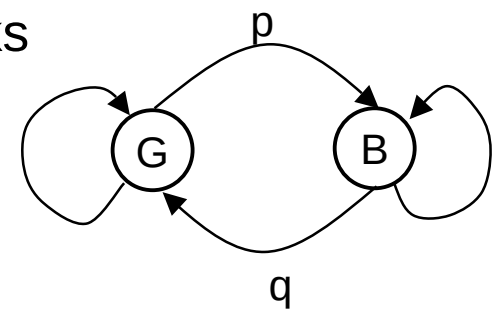
- **Communication overhead**
  - **We use real data for probing ➔**
  - **Little communication overhead!**
  - **Receiver needs larger buffer, though (order of Mbytes)**
  - **Longer start up time (still order of seconds)**
- **Processing overhead**
  - **To run estimation procedures and construct topology**
  - **Not a big concern (order of milliseconds)**
    - Small topologies (10 – 25 nodes)
    - Fast processors
- **Frequency of update**
  - **Internet path properties (loss, bw, delay) exhibit relative constancy, at least in order of minutes [Zhang, *et al.* 01]**

Simulations

- Compare selection techniques in terms of
  - The aggregated received rate, and
  - The aggregated loss rate
  - With and without peer failures
- Impact of peer availability on size of candidate set
- Size of active set
- Load on peers

Simulation: Setup

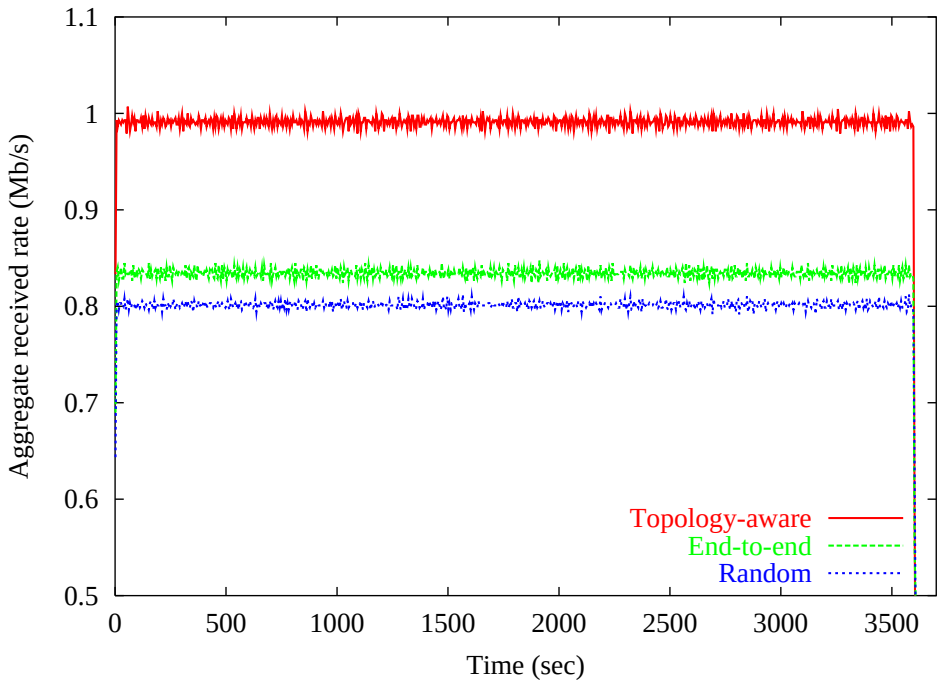
- **Topology**
  - **On average 600 routers and 1,000 peers**
  - **Hierarchical (Internet-like)**
- **Cross traffic**
  - **We approximate its effects through**
    - Attaching stochastic loss model to links
      - Two-state Markov chain
      - Captures temporal dependence in packet losses [Yajnik *et al.*, 99]
    - Randomly vary link bandwidth
      - Uniform in [$0.25R_0$, $1.5R_0$]



Simulations: Setup (cont'd)

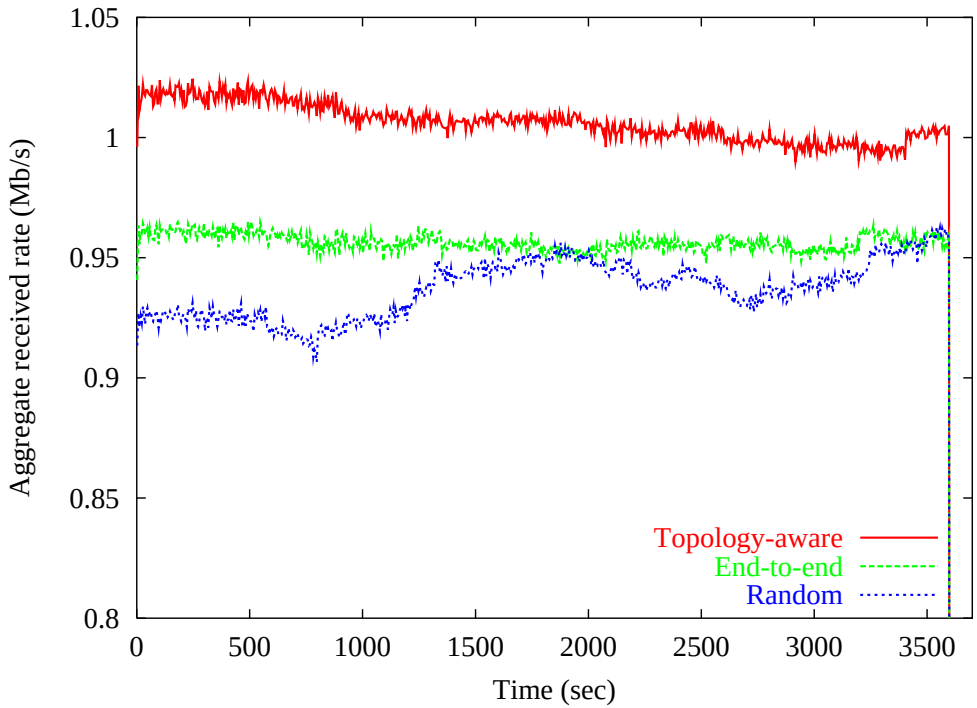
- **Streaming session**
  - Rate R_0 = 1 Mb/s
  - Duration = 60 minutes
  - Loss tolerance level α_u = 1.2
- **Peers**
  - Offered rate: uniform in $[0.125R_0,\ 0.5R_0]$
  - Availability: uniform in [0.1, 0.9]
  - Diverse P2P community
- **Results are averaged over 100 runs with different seeds**

Aggregate Rated: No Failures



- **Careful selection pays off!**

Aggregate Rate: With Peer Failures



- **Good performance, but starts to degrade as we encounter many failures ➔ How large should the candidate set be?**

PROMISE and Experiments on PlanetLab

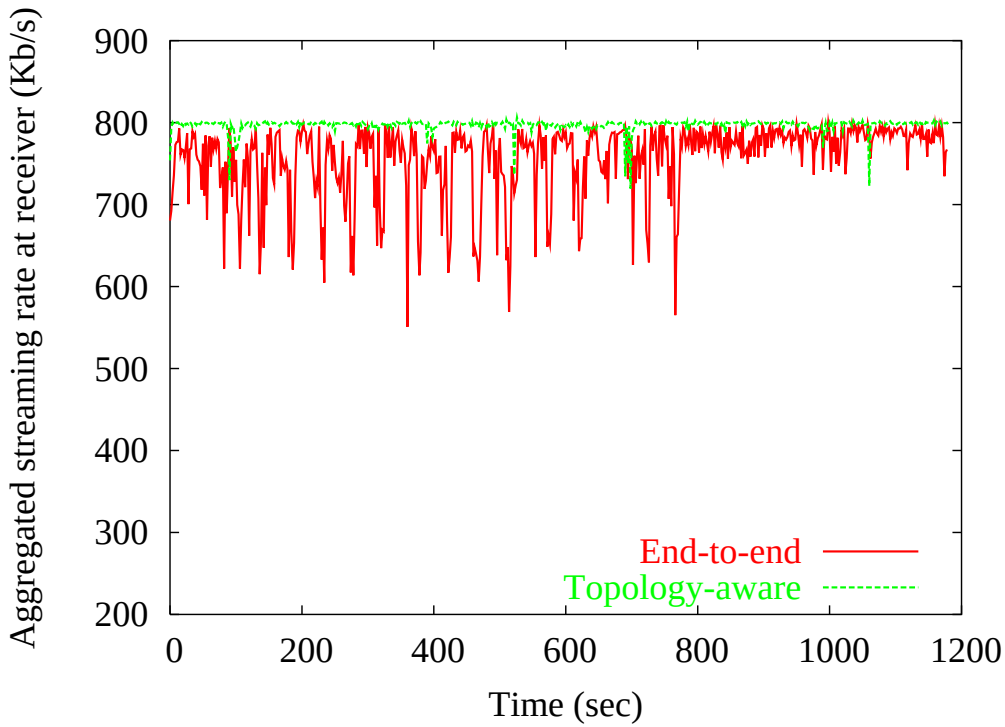
- PROMISE is a P2P media streaming system built on top of CollectCast
- Tested in local and wide area environments
- Extended Pastry to support multiple peer look up

PlanetLab Experiments*

- PROMISE is installed on 15 nodes
- Use several MPGE-4 movie traces
- Select peers using topology-aware (the one used in CollectCast) and end-to-end
- Evaluate
  - Packet-level performance
  - Frame-level performance and initial buffering
  - Impact of changing system parameters
  - Peer failure and dynamic switching

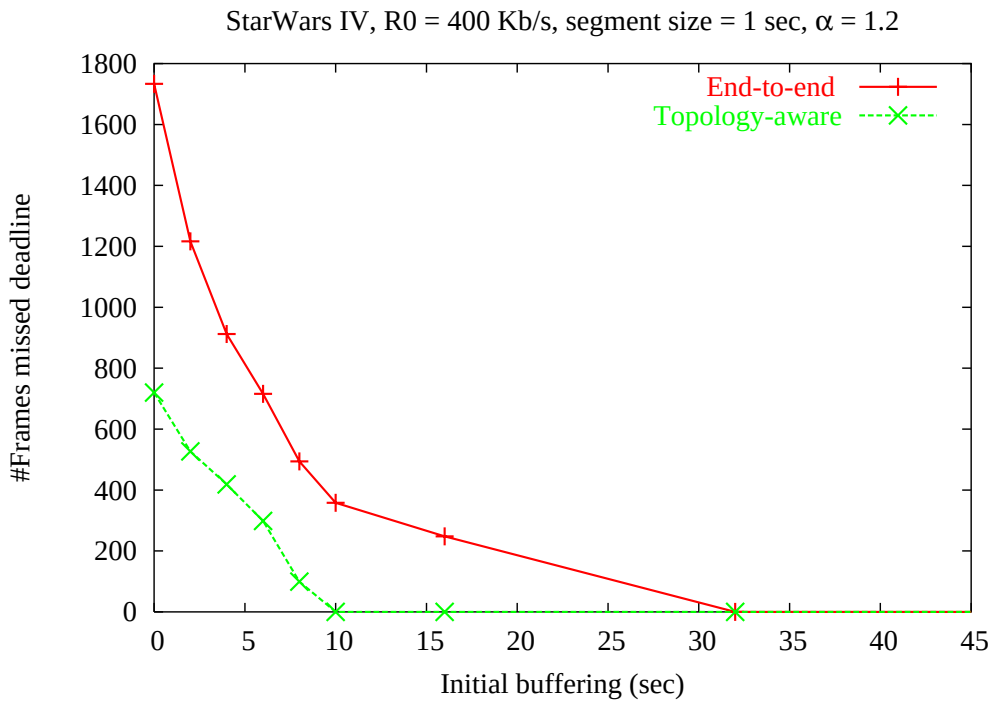
*Most of these results are presented in the extended version of the paper

Packet-Level: Aggregated Rate



- **Smoother aggregated rate achieved by CollectCast**

Frame-Level: #Frames Missed Deadline



- **Much fewer frames miss their deadlines with CollectCast**
- **CollectCast requires, on the average, half of the initial buffering time to ensure all frames meet their deadlines**

Conclusions

- New service for P2P networks (CollectCast)
  - Infer and leverage network performance information in selecting and coordinating peers
- PROMISE is built on top of CollectCast to demonstrate its merits
- Internet Experiments show proof of concept
  - Streaming from multiple, heterogeneous, failure-prone, peers is indeed feasible
- Extend P2P systems beyond file sharing
- Concrete example of network tomography

Future Work

- **Extend CollectCast beyond physical network characteristics**
  - Consider peer trustworthiness/reputation into peer selection
  - Graph labeled with trust metric
  - Would enable security-sensitive applications on top of CollectCast

Thank You!

Questions?

- The extended version of the paper is available at …

http://www.cs.purdue.edu/homes/mhefeeda/promise