CollectCast: A Tomography-Based Network Service for P2P Streaming

Mohamed Hefeeda, Ahsan Habib, Dongyan Xu, Bharat Bhargava
Computer Sciences Department, Purdue University

1. Motivation

- Target environment (e.g., P2P)
  - Multiple-to-one streaming
  - Heterogeneous, failure-prone suppliers
  - Dynamic network conditions
- Challenge
  - Achieve and maintain full-quality
- Our Solution
  - CollectCast: based on tomography

2. CollectCast

- Infer approximate network conditions (avail bw, loss, topology)
- Select best peers from a candidate set
- Adaptive assignment of rate and data to suppliers
- Seamless supplier switching to maintain full quality

3. Inference

- Adapt tomography techniques, e.g.,
  - Not interested in "exact" avail bw, rather, can a path support aggregate rate from sullying peers?
  - Probe with real (movie) data!
  - Peers are weak: coordinate probing from multiple peers
- Result
  - Topology annotated with segment-wise loss and avail bw

4. Suppliers Selection

- Find suppliers (P^{actv}) that:

Maximizes $E\left[\sum_{p\in P^{actv}} G_p R_p\right]$

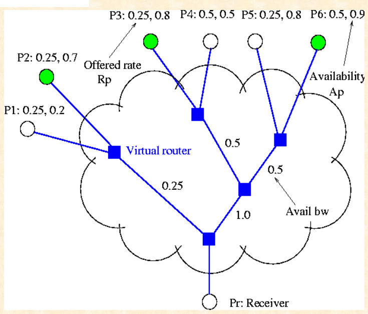
Subject to $\alpha_l R_0 \le \sum_{p\in P^{actv}} R_p \le \alpha_u R_0$

- $G_p \equiv$ How good peer p is for this session:

$$G_p = A_p \prod_{i\to j\in p\Rightarrow r} g_{i\to j} = A_p \prod_{i\to j\in p\Rightarrow r} w_{i\to j}^{(p)} x_{i\to j}$$

x: depends on loss rate

w: weight based on avail bw and level of sharing



P2, P3, P6 likely provide best quality

5. Rate/Data Assignment

- Assign rate/data to suppliers with adaptive FEC
  - Pre-encode segments, FEC(α_u)
  - Send at αR_0 to tolerate *current* aggregate loss rate
  - Typical: $1 \le \alpha \le \alpha_u = 1.25$

$$\alpha = 1 + L_\Sigma = 1 + \frac{\sum_{p\in P^{actv}} l_{p\Rightarrow r} R_p}{\sum_{p\in P^{actv}} R_p}$$

Assigned Rate

$$\hat{R}_p = \frac{\alpha R_0}{\sum_{x\in P^{actv}} R_x} R_p$$

Assigned Data

$$D_p = \left\lceil \frac{\Delta}{(2-\alpha)} \frac{\hat{R}_p}{\alpha R_0} \right\rceil$$

6. Adaptation

- Peer failure/degradation ➔ switch suppliers
  - Update topology, labels
  - Solve the maximization problem
  - Note: keep the good peers that you already have!
- Network fluctuations
  - Adjust α (loss tolerance level)
    - ✦ Reduce redundancy if network is fine
    - ✦ Increase, otherwise
  - If new α is greater than what current peers can support, add/replace peer(s)

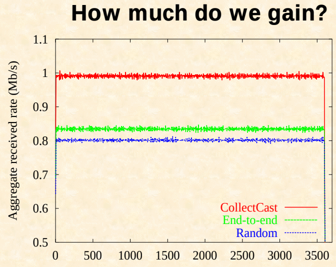
7. Overhead

- Communication overhead
  - We use real data for probing ➔ little overhead!
  - Larger receiver buffer, though (order of Mbytes)
- Processing overhead
  - To run the estimation procedures and construct the topology
  - Not a big concern (order of milliseconds)
- Frequency of update
  - Internet path properties (loss, bw, delay) exhibit a relative constancy, at least in order of minutes [Zhang *et al.*, IMW'01]

8. Evaluation: Sample Results

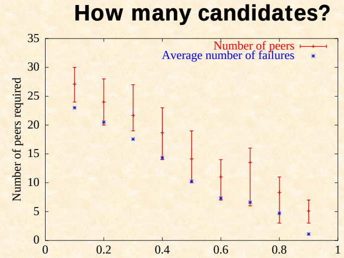
- Setup
  - Large topology, Markov losses, random avail bw
  - Peers fail
  - Select peers using
    - ✦ CollectCast (tomography)
    - ✦ E2E (no notion of shared segments)
    - ✦ Random
  - Measure aggregate received rate

How much do we gain?



Careful selection pays off!

How many candidates?



Candidate set size depends on reliability

9. Application

- PROMISE—P2P Streaming Using CollectCast
  - Integrated Pastry, CollectCast
  - To appear in ACM Multimedia Conference, November 2003
- More Info at
  - ww.cs.purdue.edu/~mhefeeda
- Support
  - NSF grant ANI-0219110