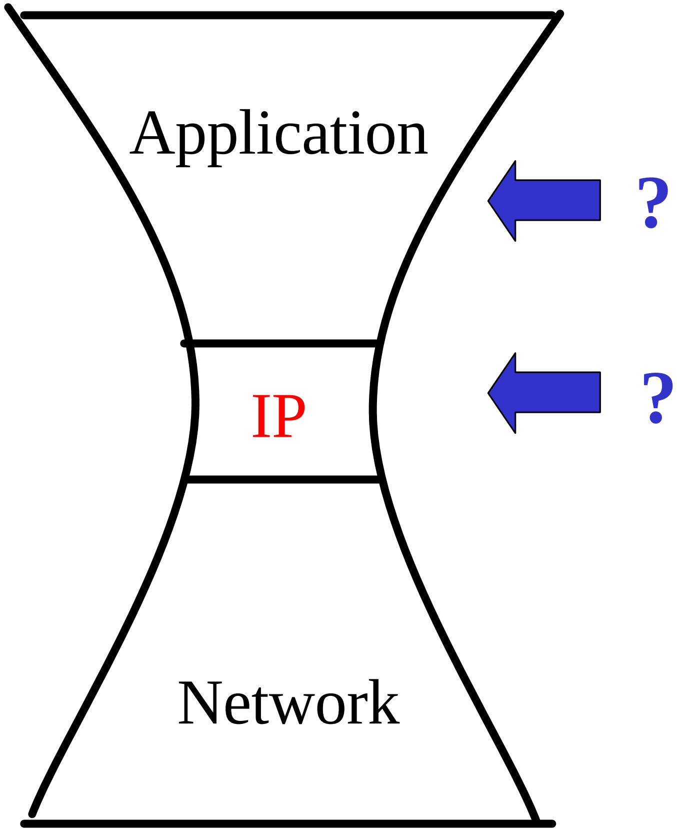
Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture

Yang-hua Chu, Sanjay Rao,
Srini Seshan and Hui Zhang

Carnegie Mellon University

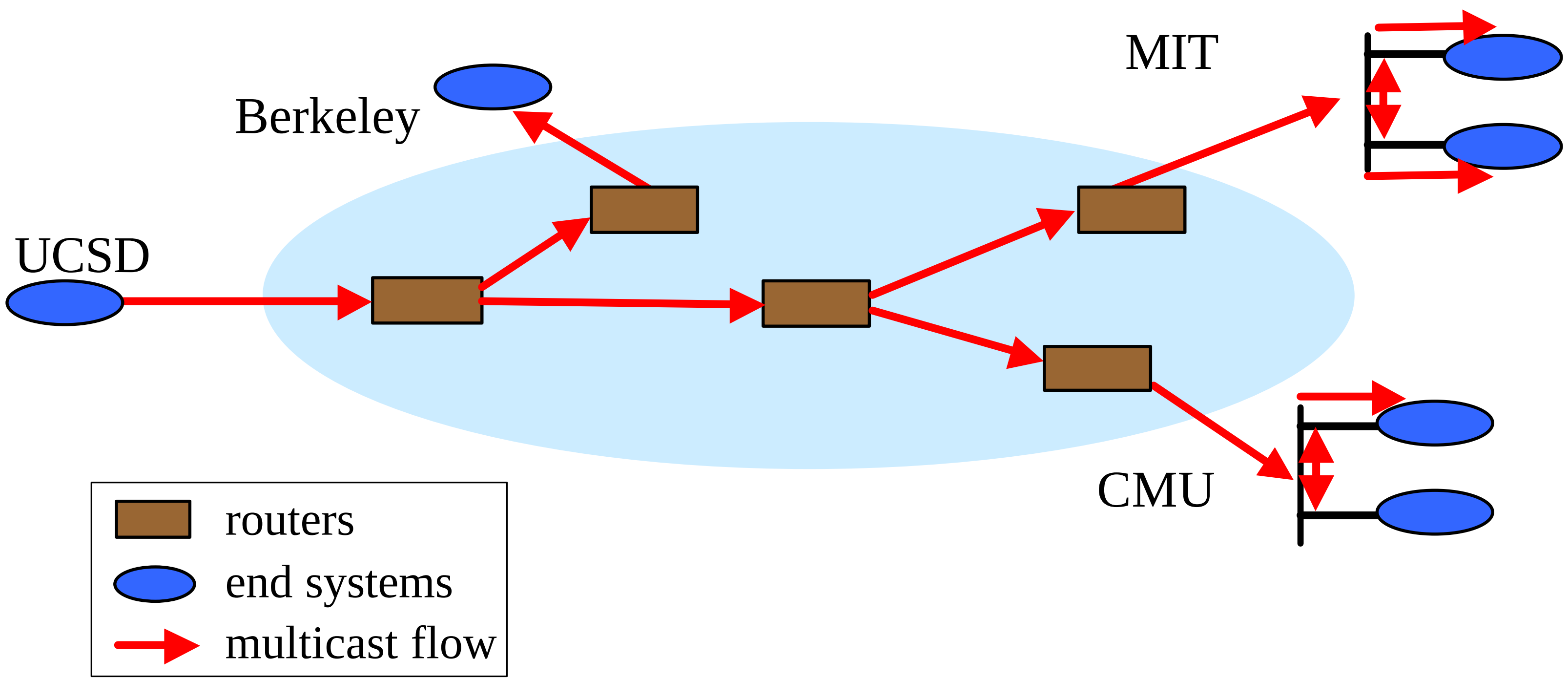
Supporting Multicast on the Internet



Internet architecture

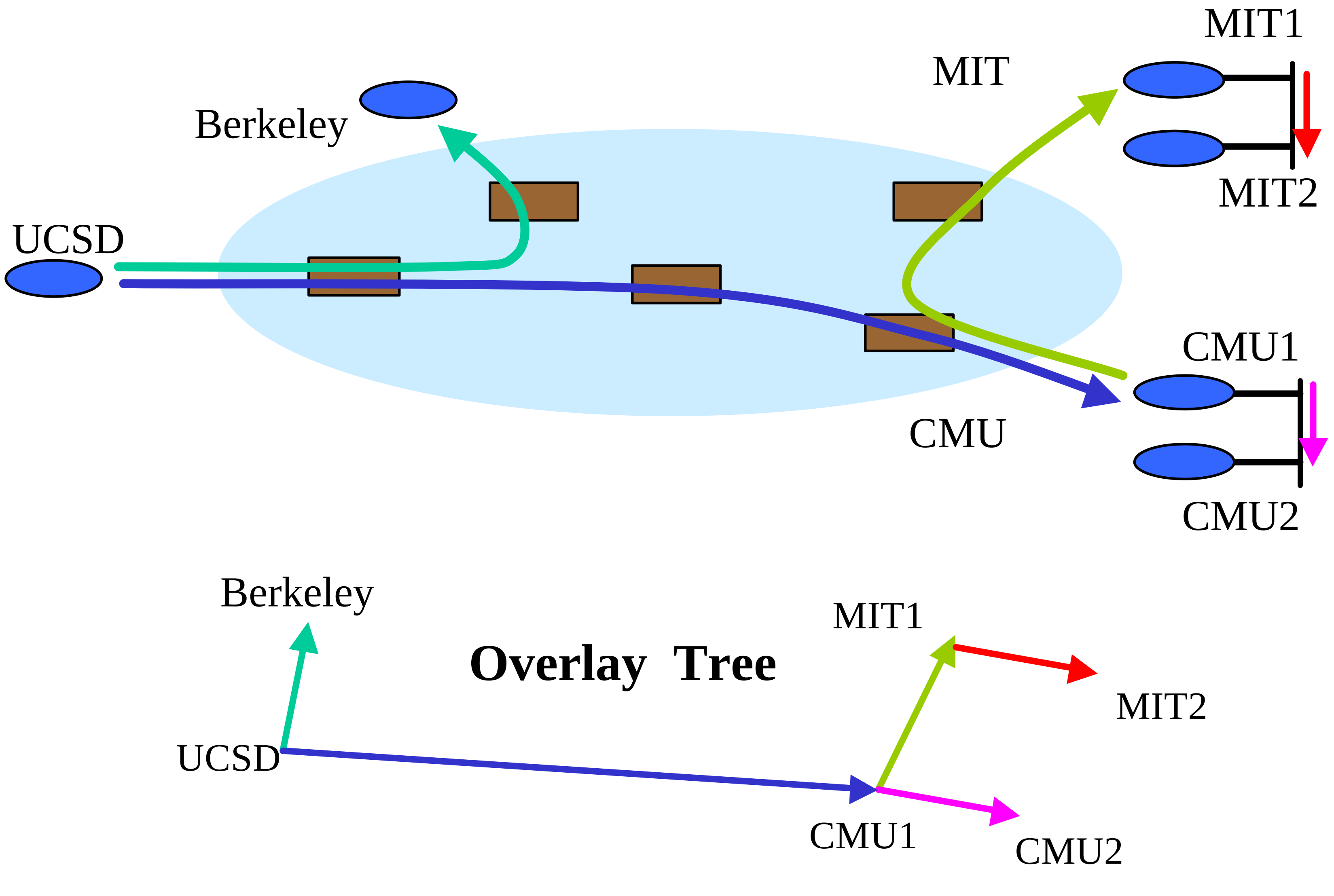
At which layer should multicast be implemented?

IP Multicast



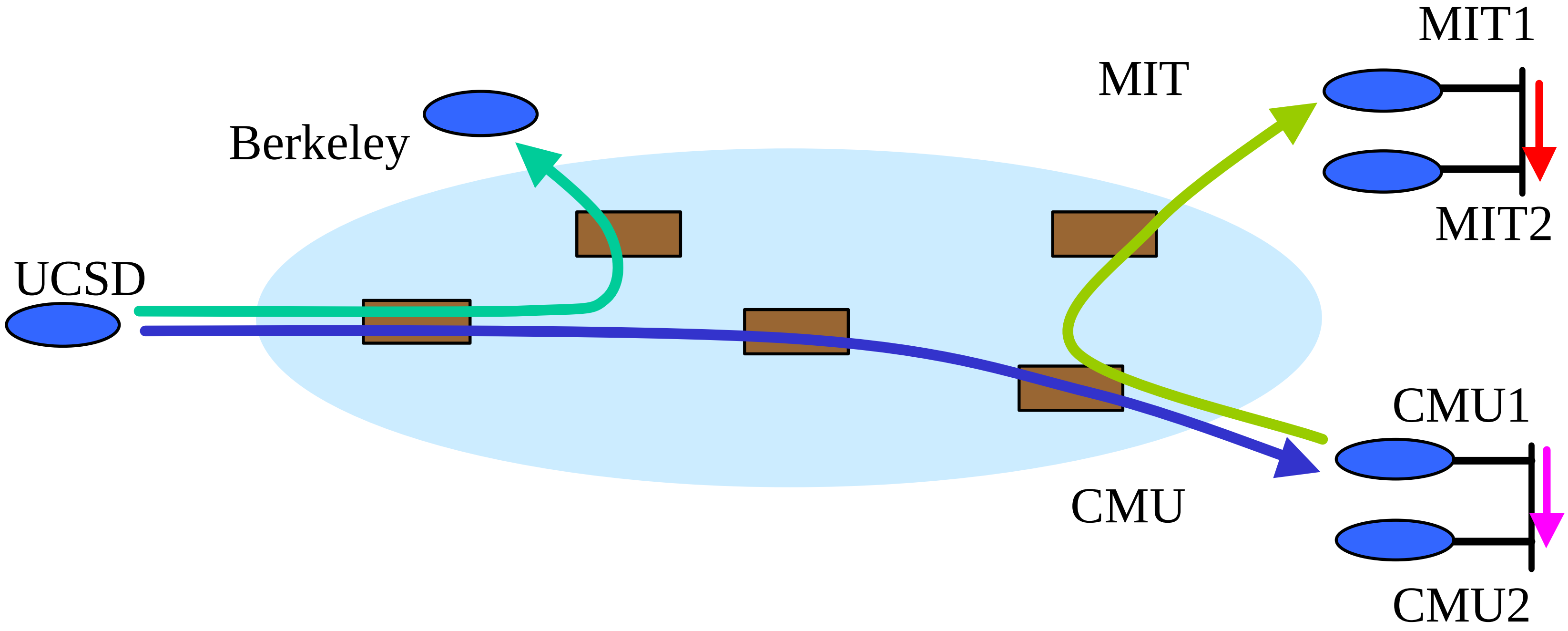
- Highly efficient
- Good delay

End System Multicast



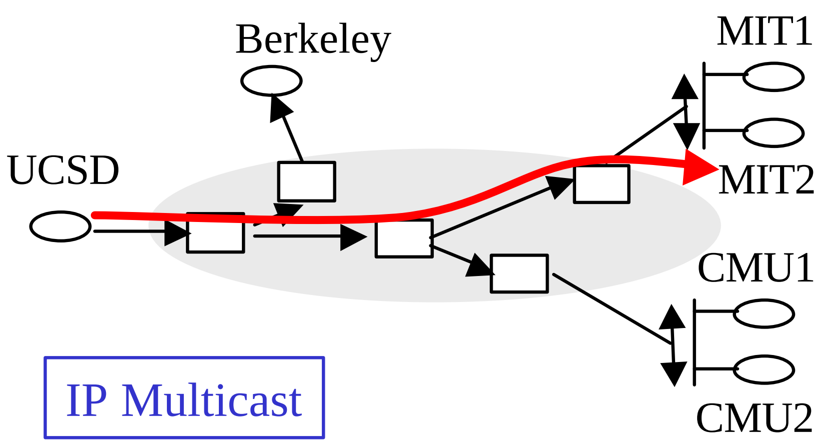
Potential Benefits over IP Multicast

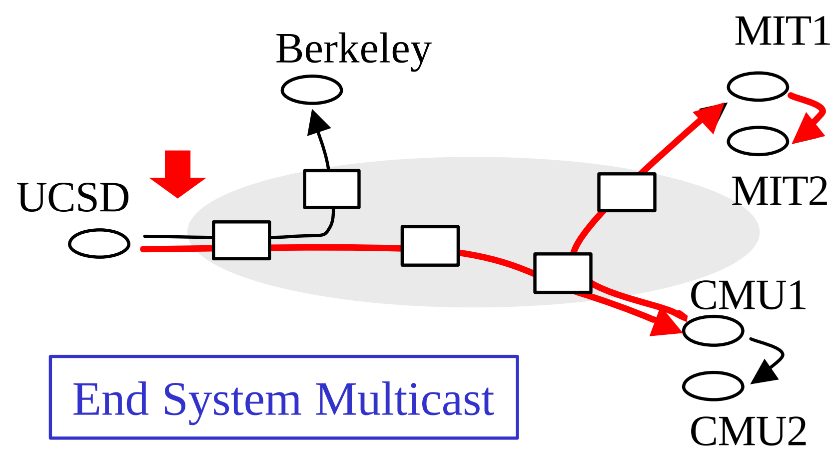
- Quick deployment
- All multicast state in end systems
- Computation at forwarding points simplifies support for higher level functionality



Concerns with End System Multicast

- Challenge to construct efficient overlay trees
- Performance concerns compared to IP Multicast
  - Increase in delay
  - Bandwidth waste (packet duplication)





Past Work

- Self-organizing protocols
  - Yoid (ACIRI), Narada (CMU), Scattercast (Berkeley), Overcast (CISCO), Bayeux (Berkeley), …
  - Construct overlay trees in distributed fashion
  - Self-improve with more network info
- Performance results showed promise, but…
  - Evaluation conducted in simulation
  - Did not consider impact of network dynamics on overlay performance

Focus of This Paper

- Can End System Multicast support real-world applications on the Internet?
  - Study in context of conferencing applications
  - Show performance acceptable even in a dynamic and heterogeneous Internet environment

- First detailed Internet evaluation to show the feasibility of End System Multicast

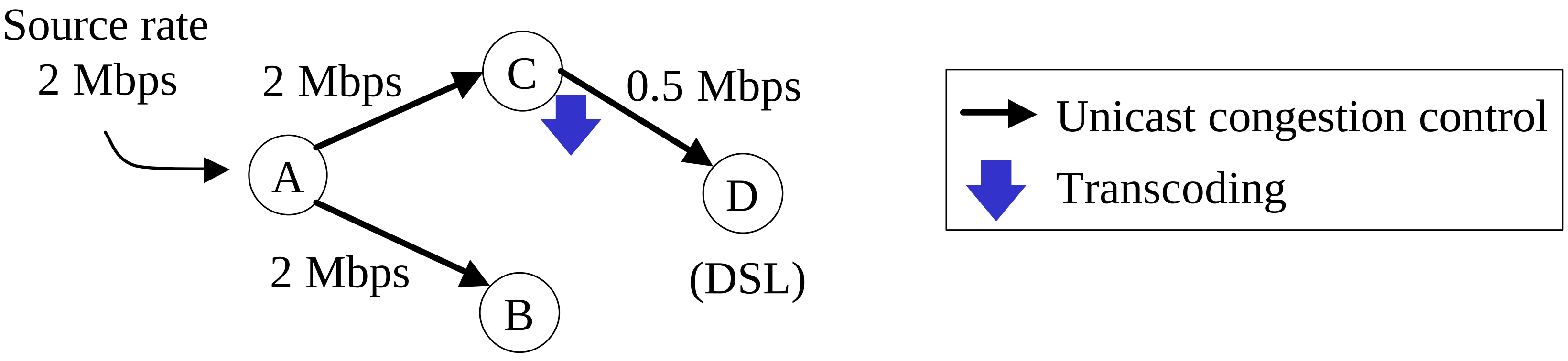
Why Conferencing?

- Important and well-studied
  – Early goal and use of multicast (vic, vat)
- Stringent performance requirements
  – High bandwidth, low latency
- Representative of interactive apps
  – E.g., distance learning, on-line games

Roadmap

- Enhancing self-organizing protocols for conferencing applications
- Evaluation methodology
- Results from Internet experiments

Supporting Conferencing in ESM (End System Multicast)



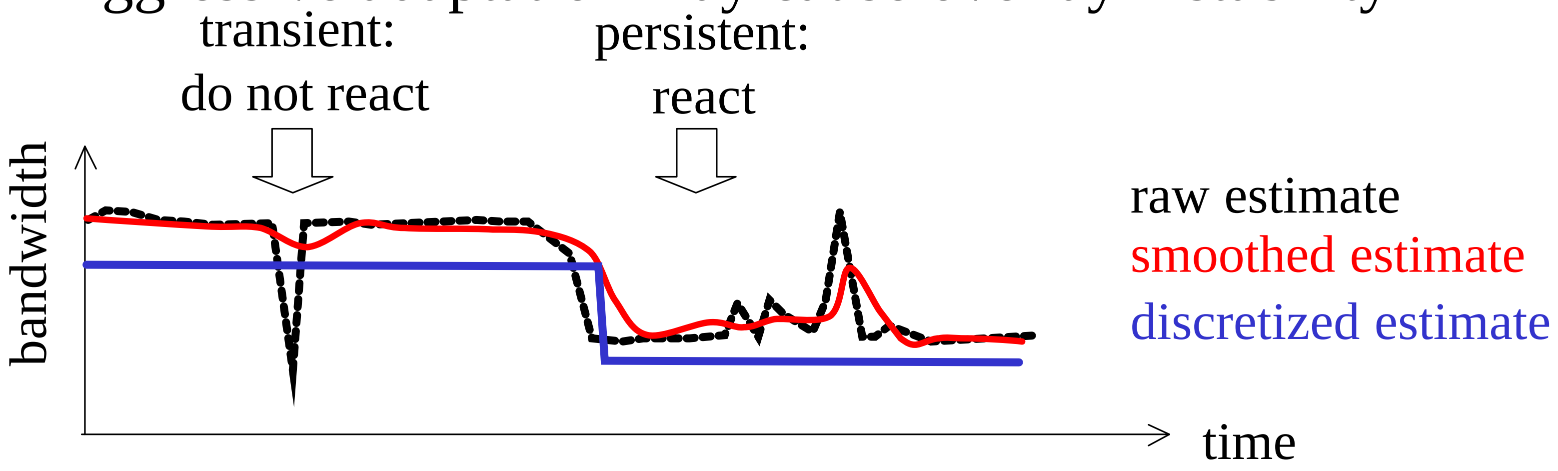
- Framework
  - Unicast congestion control on each overlay link
  - Adapt to the data rate using transcoding
- Objective
  - High bandwidth and low latency to all receivers along the overlay

Enhancements of Overlay Design

- Two new issues addressed
  - Dynamically adapt to changes in network conditions
  - Optimize overlays for multiple metrics
    - Latency and bandwidth
- Study in the context of the Narada protocol (Sigmetrics 2000)
  - Techniques presented apply to all self-organizing protocols

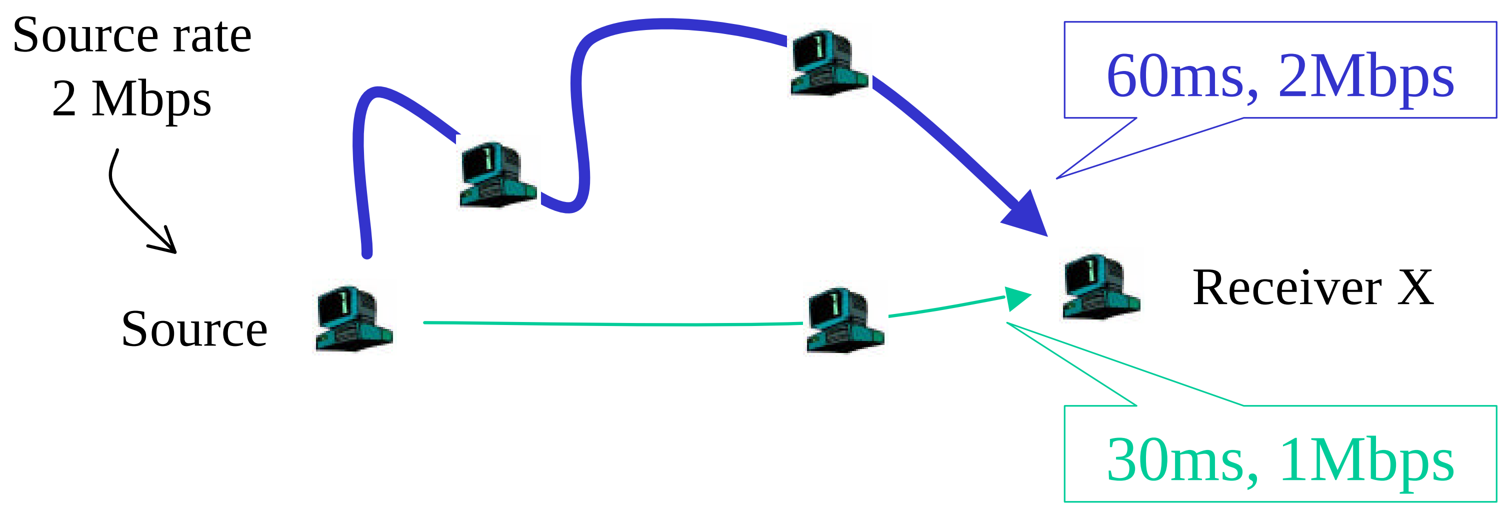
Adapt to Dynamic Metrics

- Adapt overlay trees to changes in network condition
  - Monitor bandwidth and latency of overlay links (note: CAP-probe gives both)
- Link measurements can be noisy
  - Aggressive adaptation may cause overlay instability



- Capture the long term performance of a link
  - Exponential smoothing, Metric discretization

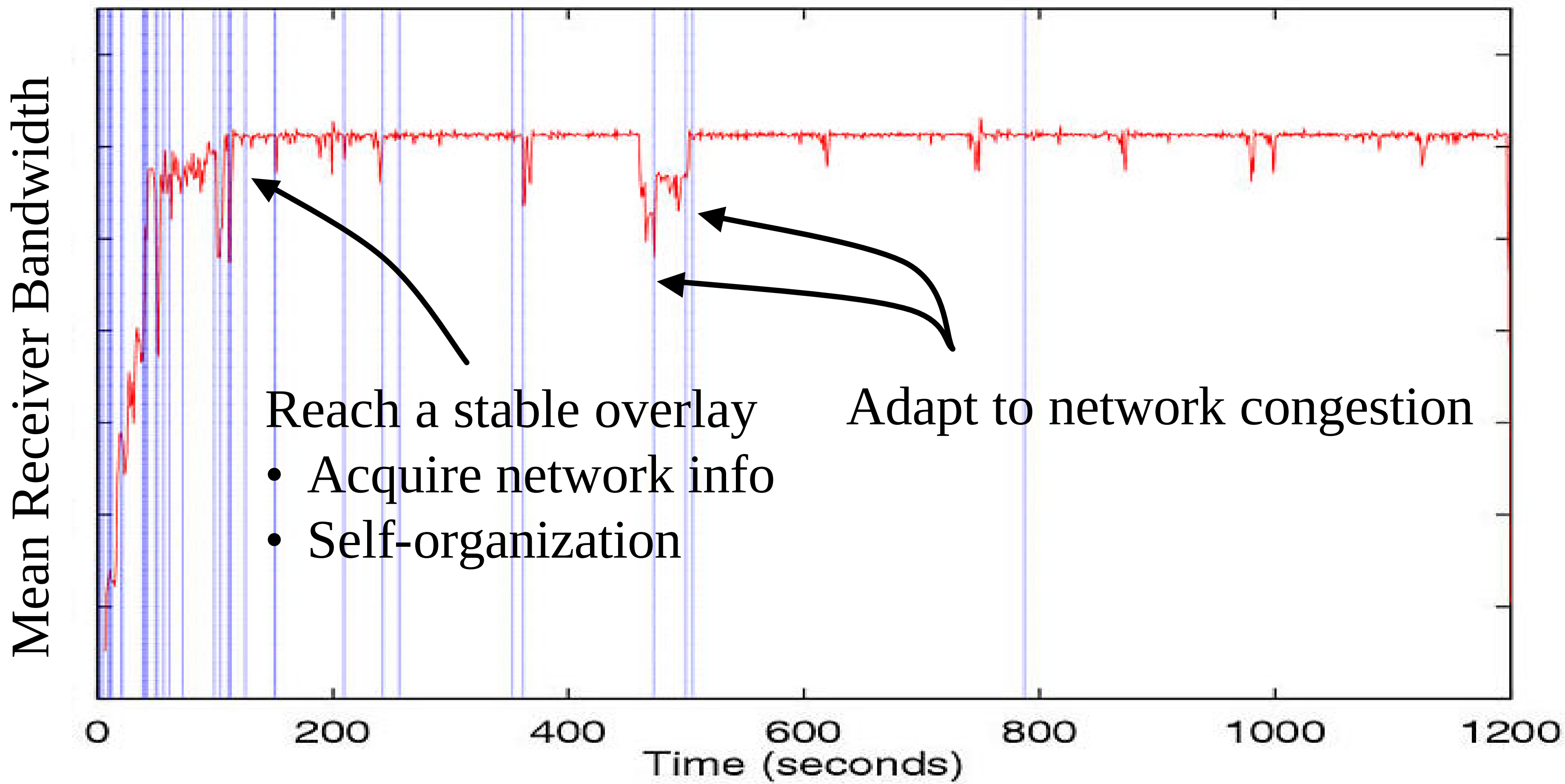
Optimize Overlays for Dual Metrics



- Prioritize bandwidth over latency
- Break tie with shorter latency

Example of Protocol Behavior

- All members join at time 0
- Single sender, CBR traffic



Evaluation Goals

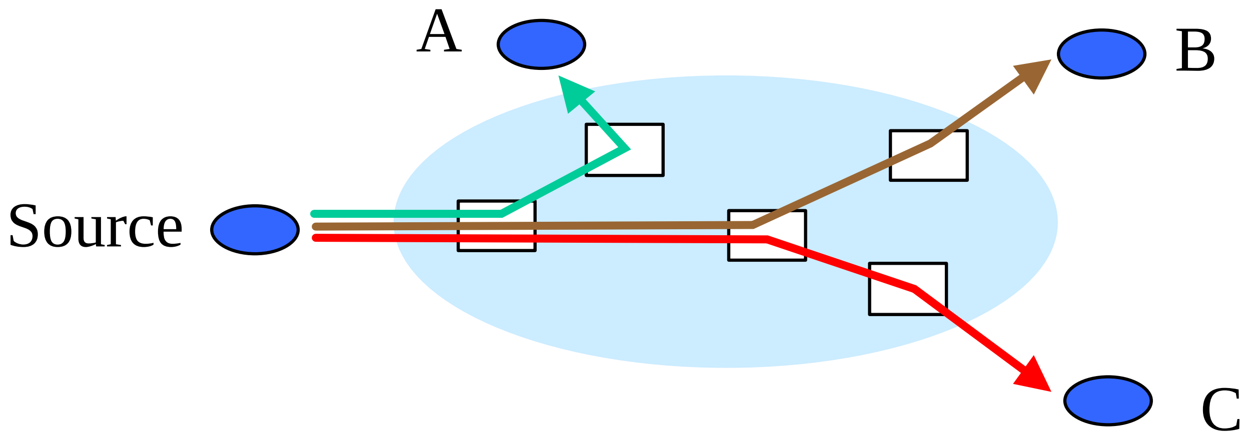
- Can ESM provide application level performance comparable to IP Multicast?
- What network metrics must be considered while constructing overlays?
- What is the network cost and overhead?

Evaluation Overview

- Compare performance of our scheme with
  - Benchmark (IP Multicast)
  - Other overlay schemes that consider fewer network metrics
- Evaluate schemes in different scenarios
  - Vary host set, source rate
- Performance metrics
  - Application perspective: latency, bandwidth
  - Network perspective: resource usage, overhead

Benchmark Scheme

- IP Multicast not deployed (Mbone is an overlay)
- Sequential Unicast: an approximation
  - Bandwidth and latency of unicast path from source to each receiver
  - Performance similar to IP Multicast with ubiquitous (well spread out) deployment



Overlay Schemes

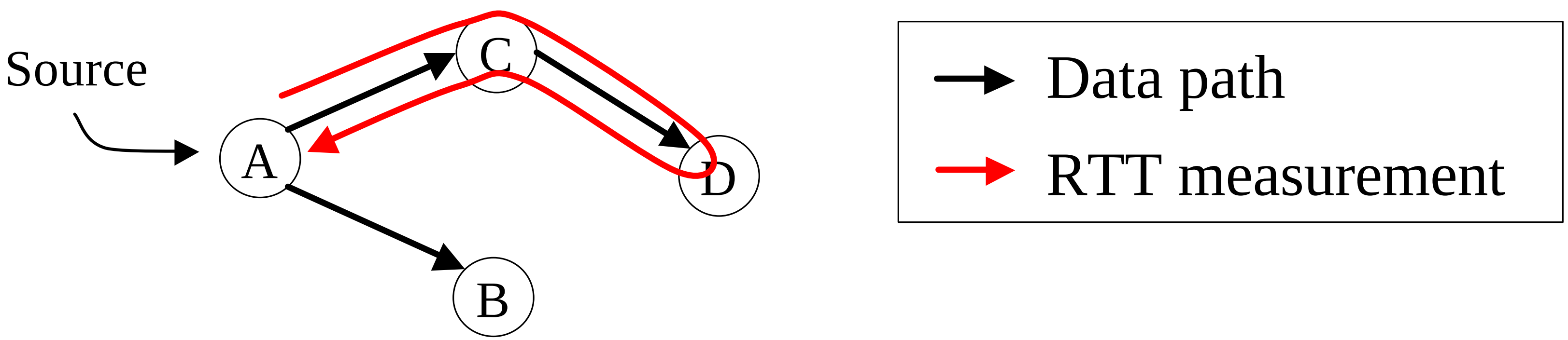
Overlay Scheme	Choice of Metrics	
	Bandwidth	Latency
Bandwidth-Latency	✓	✓
Bandwidth-Only	✓	✗
Latency-Only	✗	✓
Random	✗	✗

Experiment Methodology

- Compare different schemes on the Internet
  - Ideally: run different schemes concurrently
  - Interleave experiments of schemes
  - Repeat same experiments at different time of day
  - Average results over 10 experiments
- For each experiment
  - All members join at the same time
  - Single source CBR traffic with **TFRC** adaptation
  - Each experiment lasts for 20 minutes

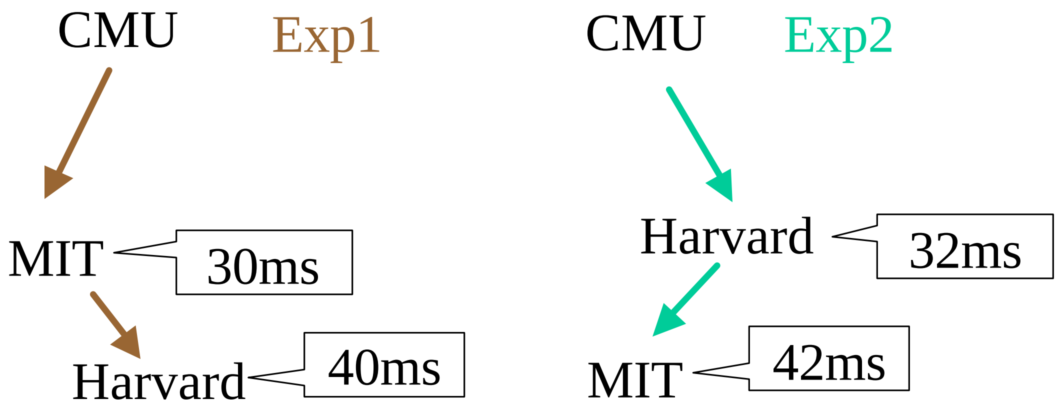
Application Level Metrics

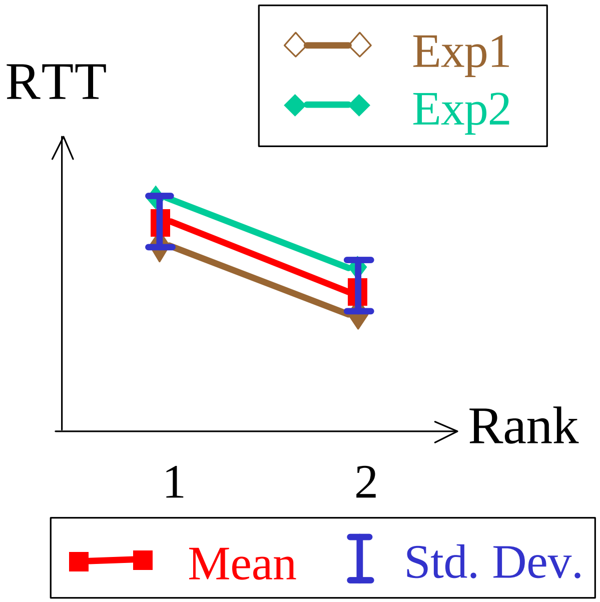
- Bandwidth (throughput) observed by each receiver
- RTT between source and each receiver along overlay



These measurements include queueing and processing delays at end systems

Performance of Overlay Scheme





Different runs of the same scheme may produce different but "similar quality" trees

"Quality" of overlay tree produced by a scheme

- Sort ("rank") receivers based on performance
- Take mean and std. dev. on performance of same rank across multiple experiments
- Std. dev. shows variability of tree quality

Factors Affecting Performance

- Heterogeneity of host set
  - *Primary Set*: 13 university hosts in U.S. and Canada
  - *Extended Set*: 20 hosts, which includes hosts in Europe, Asia, and behind ADSL
- Source rate
  - Fewer Internet paths can sustain higher source rate
  - More intelligence required in overlay constructions

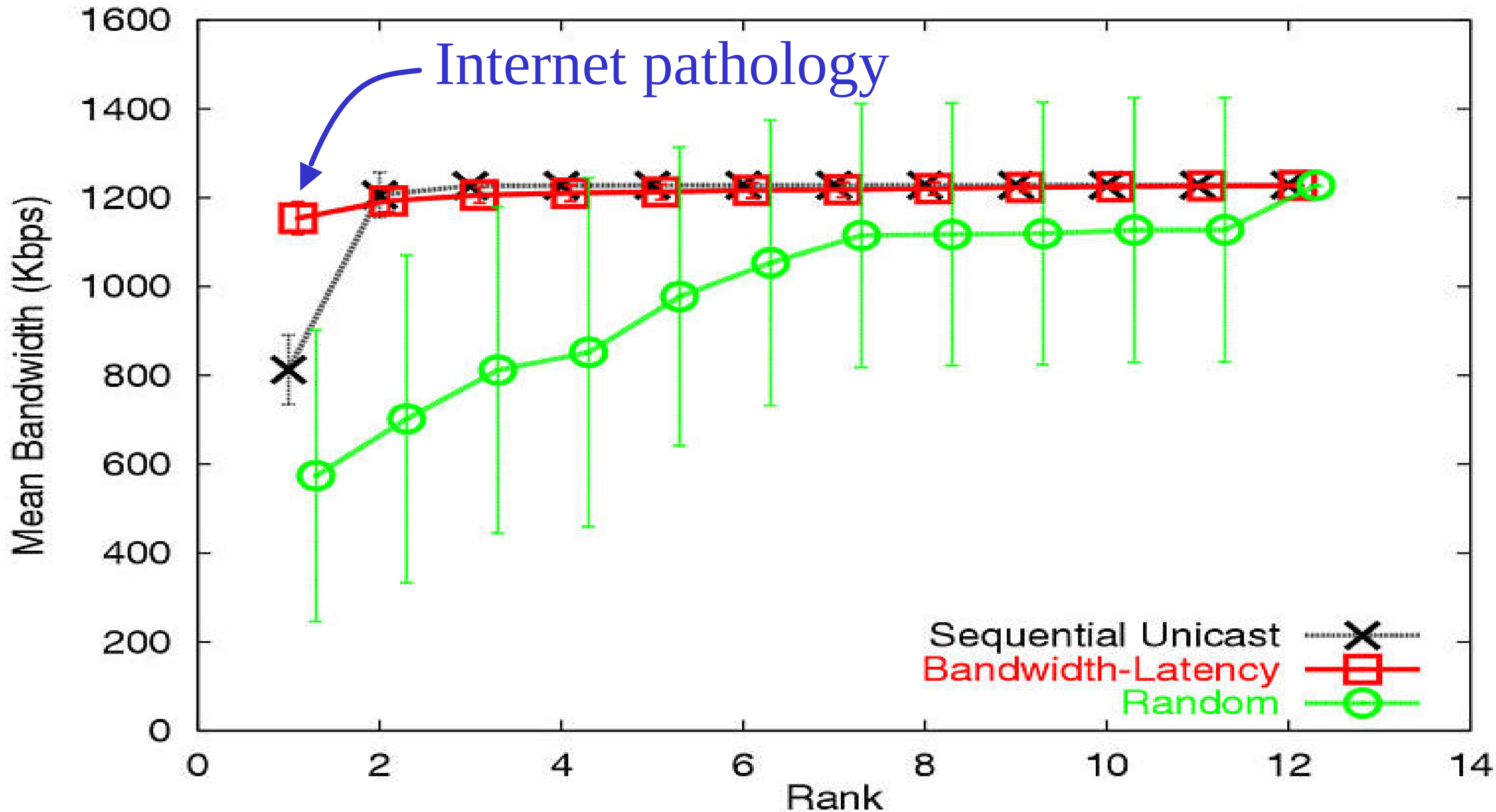
Three Scenarios Considered

Primary Set 1.2 Mbps	Primary Set 2.4 Mbps	Extended Set 2.4 Mbps

(lower) ← "stress" to overlay schemes → (higher)

- Does ESM work in different scenarios?
- How do different schemes perform under various scenarios?

BW, Primary Set, 1.2 Mbps



Naïve scheme performs poorly even in a less “stressful” scenario

RTT results show similar trend

Scenarios Considered

Primary Set
1.2 Mbps

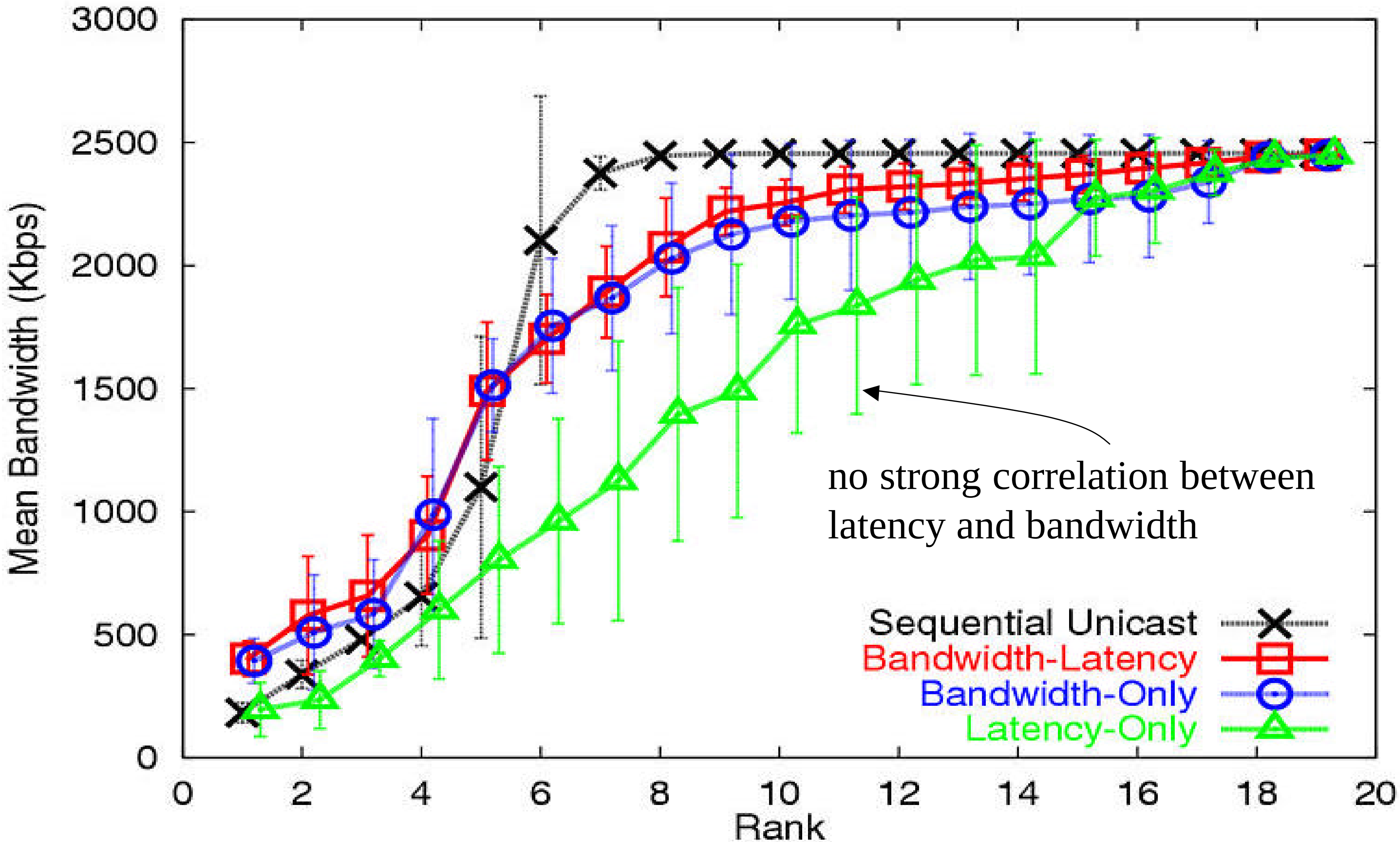
Primary Set
2.4 Mbps

Extended Set
2.4 Mbps

(lower) ← "stress" to overlay schemes → (higher)

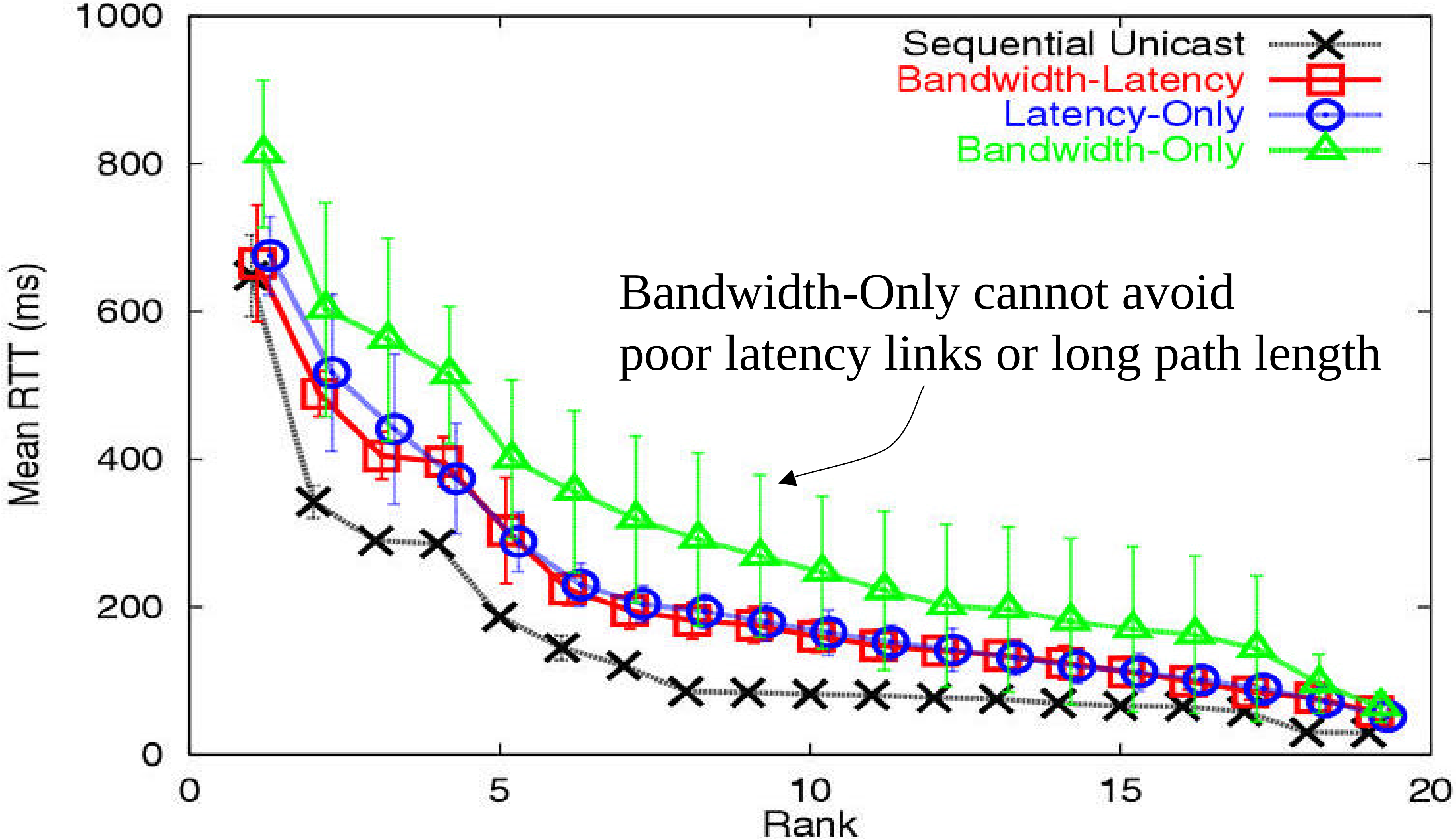
- Does an overlay approach continue to work under a more "stressful" scenario?
- Is it sufficient to consider just a single metric?
  - *Bandwidth-Only, Latency-Only*

BW, Extended Set, 2.4 Mbps



Optimizing only for latency has poor bandwidth performance

RTT, Extended Set, 2.4Mbps



Optimizing only for bandwidth has poor latency performance

Summary so far…

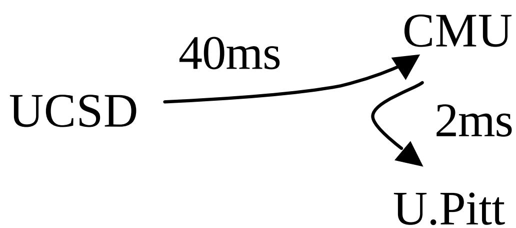
- For best application performance: adapt dynamically to **both latency and bandwidth** metrics
- *Bandwidth-Latency* performs comparably to IP Multicast (*Sequential-Unicast)*

- What is the network cost and overhead?

Resource Usage (RU)

Captures consumption of network resource of overlay tree

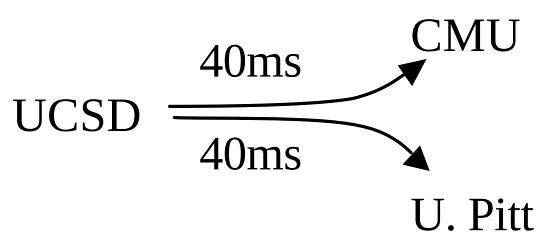
- Overlay link RU = propagation delay
- Tree RU = sum of link RU



Efficient (RU = 42ms)

Scenario: Primary Set, 1.2 Mbps
(normalized to IP Multicast RU)

IP Multicast	1.0
Bandwidth-Latency	1.49
Random	2.24
Naïve Unicast	2.62



Inefficient (RU = 80ms)

Protocol Overhead

$$\text{Protocol overhead} = \frac{\text{total non-data traffic (in bytes)}}{\text{total data traffic (in bytes)}}$$

- Results: Primary Set, 1.2 Mbps
  - Average overhead = 10.8%
  - 92.2% of overhead is due to bandwidth probe
- Current scheme employs active probing for available bandwidth
  - Simple heuristics to eliminate unnecessary probes
  - Focus of our current research

Contribution

- First detailed Internet evaluation to show the feasibility of End System Multicast architecture
  - Study in context of a/v conferencing
  - Performance comparable to IP Multicast
- Impact of metrics on overlay performance
  - For best performance: use both latency and bandwidth
- More info: http://www.cs.cmu.edu/~narada