The Feasibility of Supporting Large-Scale Live Streaming Applications with Dynamic Application End-Points

Kay Sripanidkulchai,

Aditya Ganjam, Bruce Maggs, and Hui Zhang

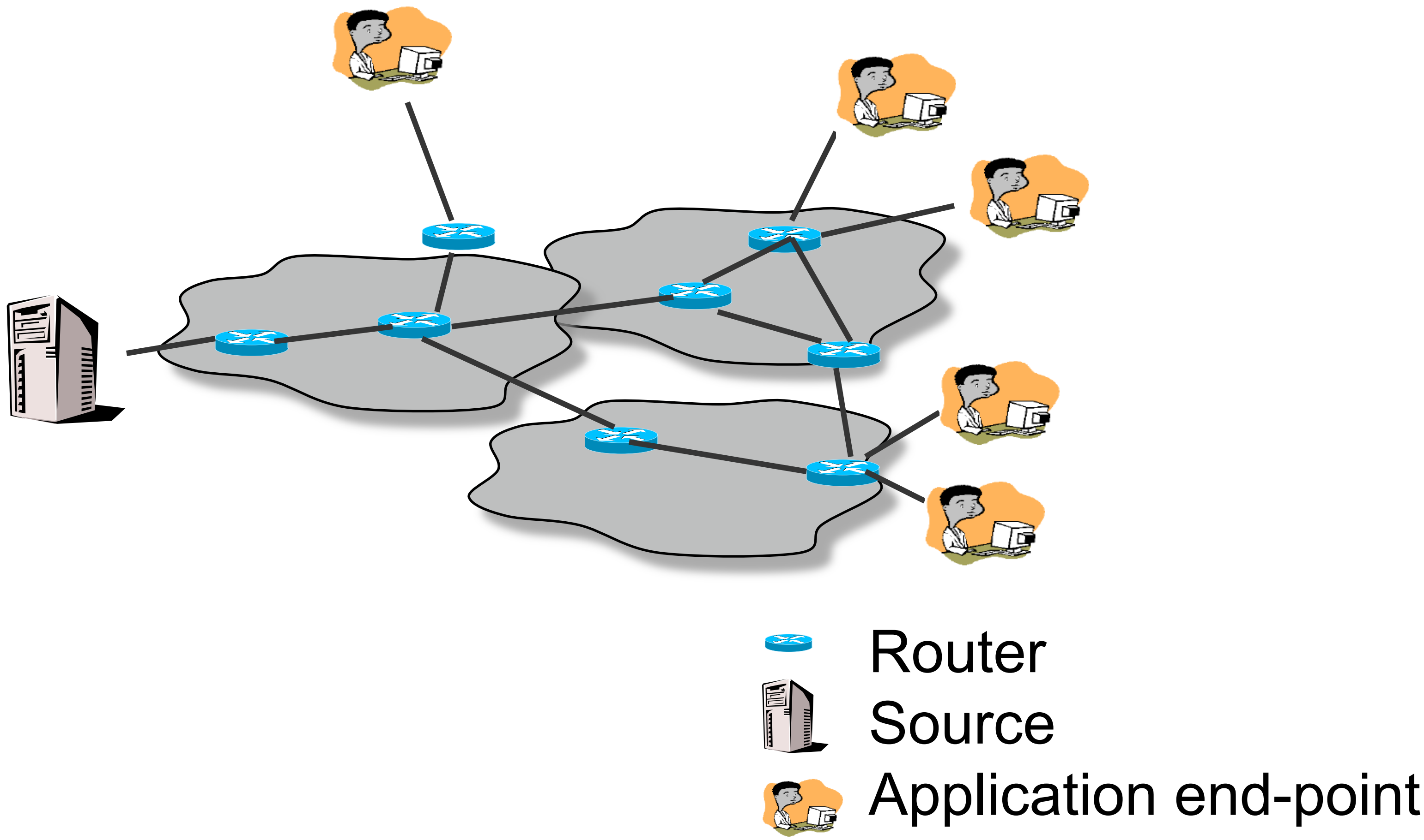
Instructor: Fabian Bustamante

Presented by: Mario Sanchez

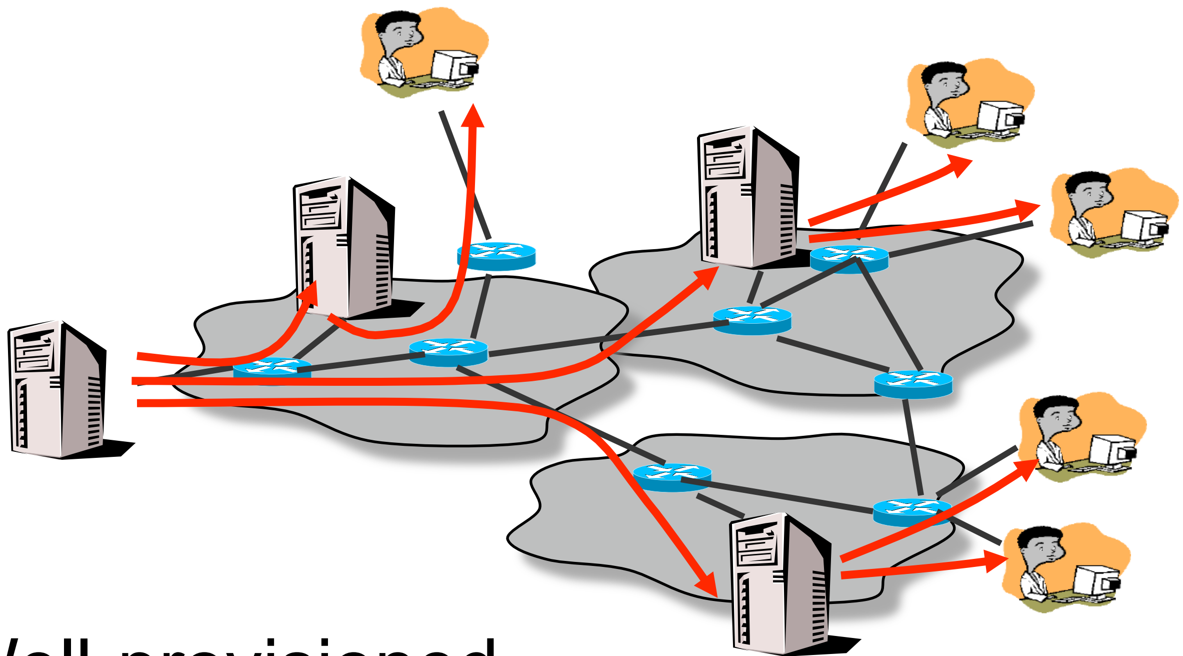
The focus of this paper

- Generate new insight on the feasibility of application end-point architectures for large scale broadcast
- Methodology
  - Analysis and simulation
  - Leverage an extensive set of real-world workloads from Akamai (infrastructure-based architecture)

Overlay multicast architectures



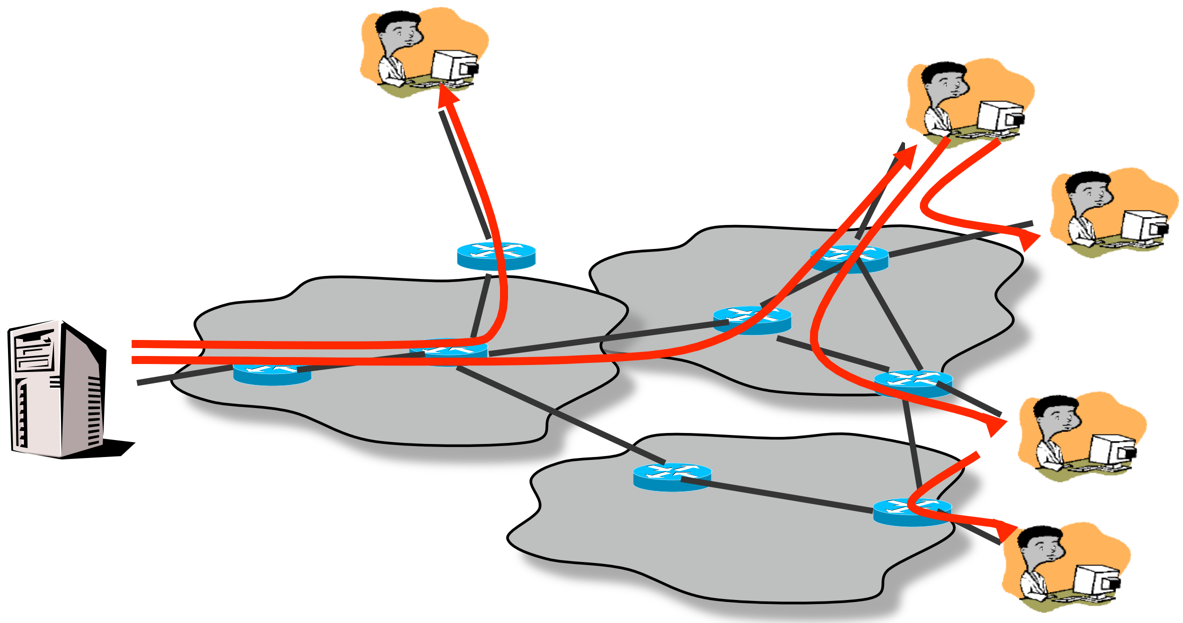
Infrastructure-based architecture [Akamai]



+ Well-provisioned

Router
Source
Application end-point
Infrastructure server

Application end-point architecture [End System Multicast (ESM)]



+ Instantly deployable
+ Enables ubiquitous broadcast

Feasibility of supporting large-scale groups with an application end-point architecture?

- Is the overlay stable enough despite dynamic participation?
- Is there enough upstream bandwidth?
- Are overlay structures efficient?

Large-scale groups

- Challenging to address these fundamental feasibility questions
  - Little knowledge of what large-scale live streaming is like

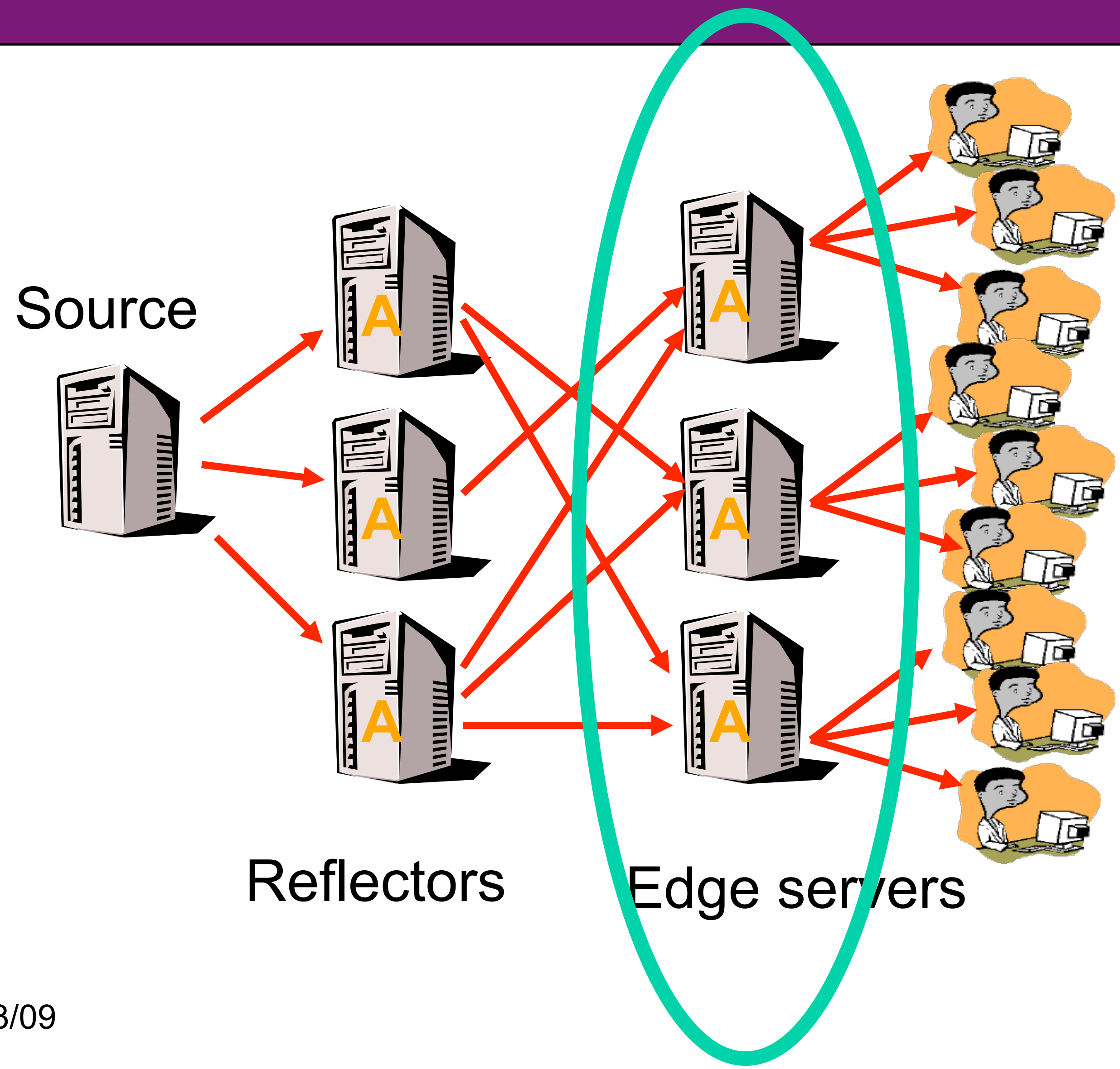
Talk outline

- Akamai live streaming workload
- With an application end-point architecture
  - Is the overlay stable enough despite dynamic participation?
  - Is there enough upstream bandwidth?
  - Are overlay structures efficient?
- Summary

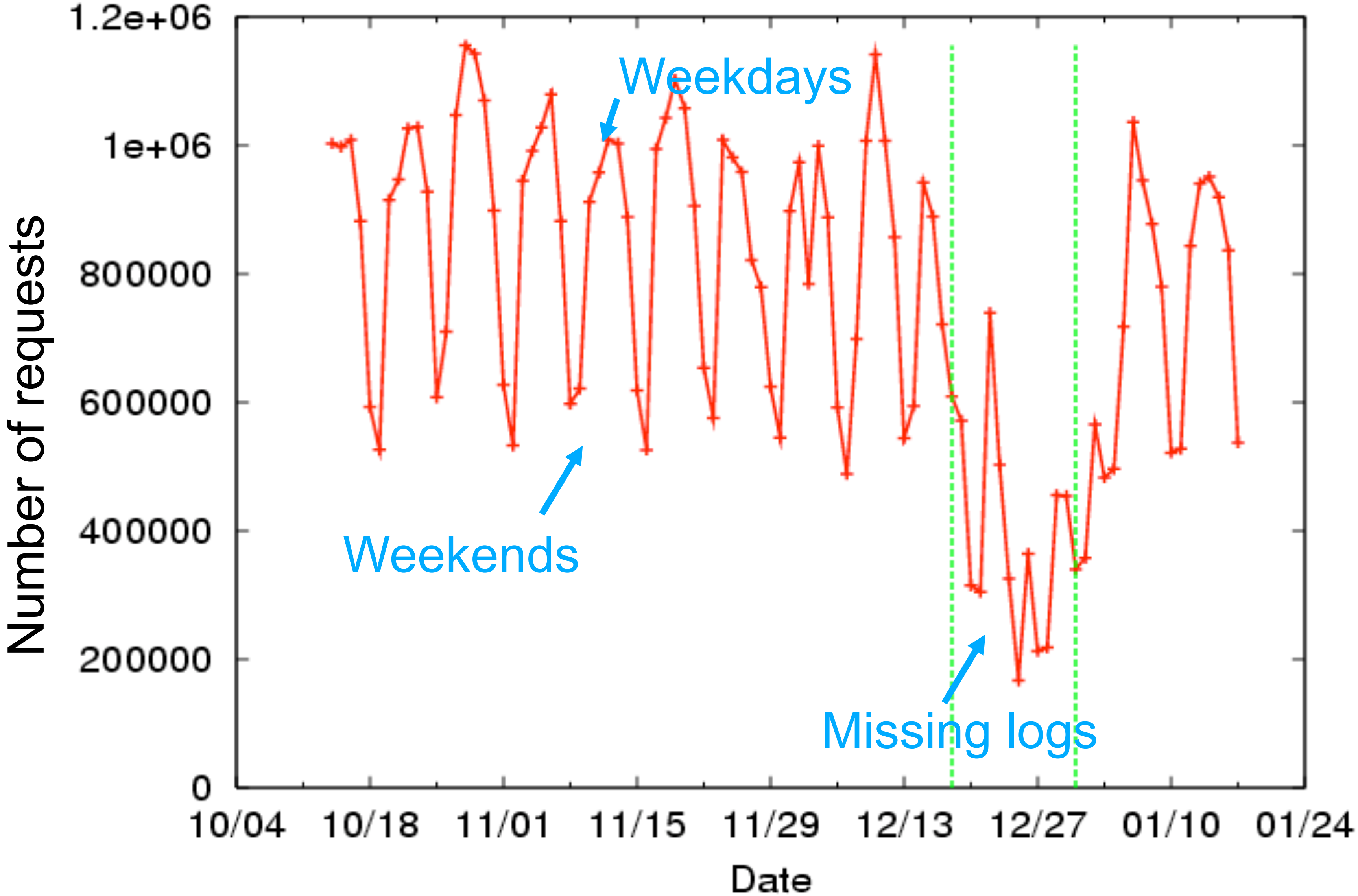
Measurements used in this study

- Akamai live streaming traces
  - Trace format for a request
    [IP, Stream URL, Session start time, Session duration]
- Additional measurements collected
  - Hosts' upstream bandwidth
  - Hosts' GNP coordinates

Akamai live streaming infrastructure



Request volume (daily)



Extensive traces

- ~ 1,000,000 daily requests
- ~ 200,000 daily client IP addresses from over 200 countries
- ~ 1,000 daily streams
- ~ 1,000 edge servers
- ~ Everyday, over a 3-month period
- ~ Quicktime, Real, Windows Media Player

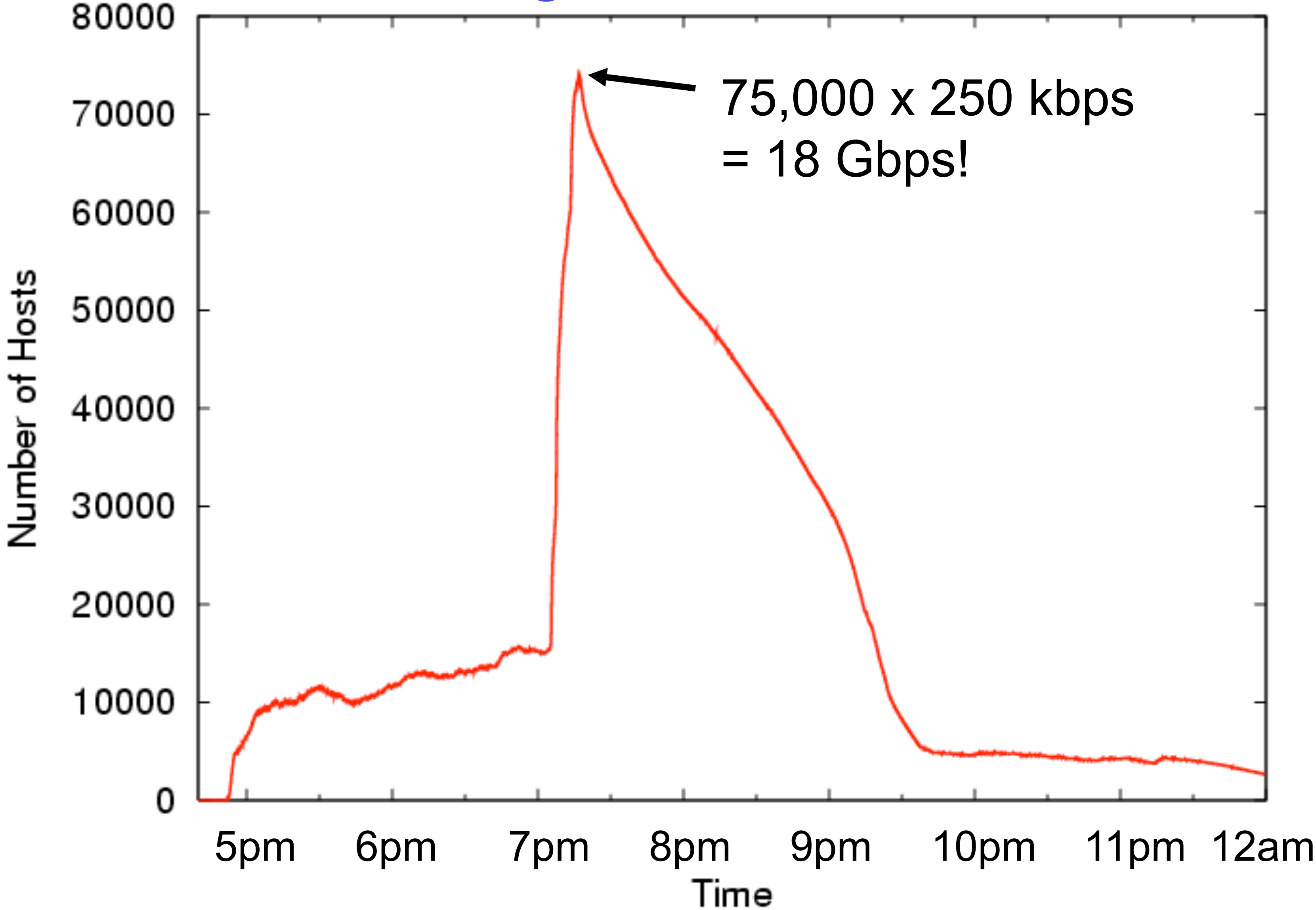
Definitions

- Two categories of streaming (event duration)
  - Non-stop events
  - Short duration events
  - Divided into 24-hour events called STREAMS
- Definitions
  - Large-scale: peak group size of over 1,000 entities
  - Entity: unique host (IP)
  - Incarnation: entity connection to broadcast

Largest Event

- Characteristics of the traces
  - Stream encoding bit-rate < 80kbps = audio.
  - Overall: 71% audio vs. 7% video vs. 22 % unknown
  - 660 large-scale streams: 605 audio, 55 video
- 3 encoding streams: (a) 20 kbps, audio; (b) 100 kbps, audio and video; (c) 250 kbps, audio and video
- 2 hour duration; all three encodings treated as one with 250 kbps requirement

Largest stream

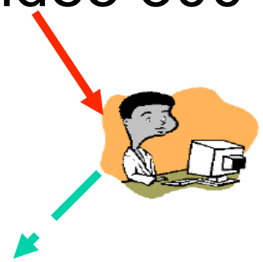


Talk outline

- Akamai live streaming workload
- With an application end-point architecture
  - Is there enough upstream bandwidth?
  - Is the overlay stable enough despite dynamic participation?
  - Are overlay structures efficient?
- Summary

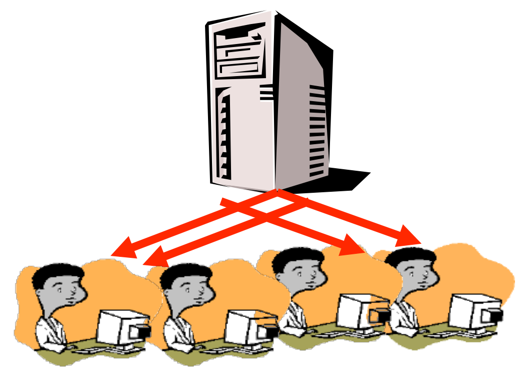
Is there enough upstream bandwidth to support all hosts?

Video 300 kbps



DSL

Upstream bandwidth
only 128 kbps



DSL

DSL

Saturated tree

What if application
end-points are all DSL?

Bandwidth Estimation

- Bandwidth Collection:
  – Direct measurements alone, out of question
- Bandwidth Collection:

  Data Mining

  – 72% of hosts: bandwidth reported by broadbandreports.com

  Active Measurements

  – 7.6%: IP /24 block measurement / packet pair to estimate technology (table bellow)

Bandwidth Estimation

Inference:

– 7.1%: EdgeScape IP to technology

– 2.2%: DNS name to technology

– 1.2%: Manually known domains with not-common-DNS-names to technology

90% of IP addresses with estimates

10% unknown

Access technology	Packet-pair measurement	Outgoing bandwidth estimate
Dial-up modems	0 kbps $\leq$ BW $<$ 100 kbps	30 kbps
DSL, ISDN, Wireless	100 kbps $\leq$ BW $<$ 600 kbps	100 kbps
Cable modems	600 kbps $\leq$ BW $<$ 1 Mbps	250 kbps
Edu, Others	BW $\geq$ 1 Mbps	BW

Outbound BW unit: degree vs. kbps

- Resources: amount of outgoing bandwidth that hosts in the system can contribute.
- Normalized bandwidth value by encoding bit rate:

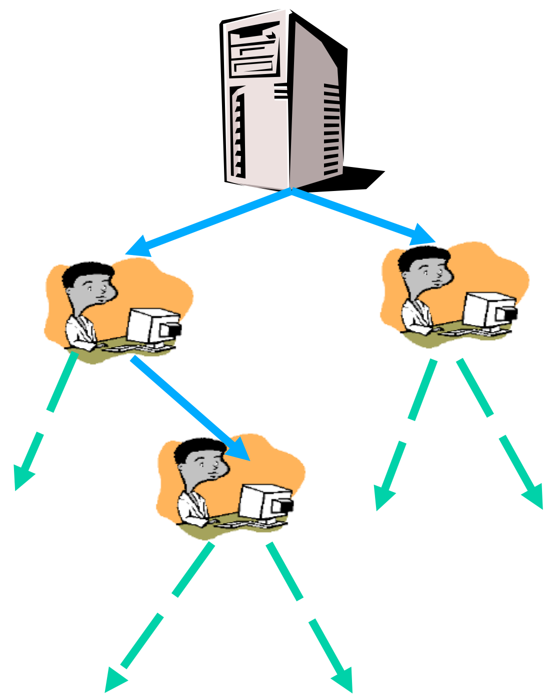
  300 kbps bandwidth, 250 kbps encoding

  = 300/250 = 1 degree
- Largest Event:

Type	Degree-bound	Number of hosts
Free-riders	0	58646 (49.3%)
Contributors	1	22264 (18.7%)
Contributors	2	10033 (8.4%)
Contributors	3-19	6128 (5.2%)
Contributors	20	8115 (6.8%)
Unknown	-	13735 (11.6%)
Total	-	118921 (100%)

Metric: Resource index

- Ratio of the supply to the demand of upstream bandwidth; Resource index == 1 means the system is saturated
- Resource index == 2 means the system can support two times the current members in the system



Resource Index:

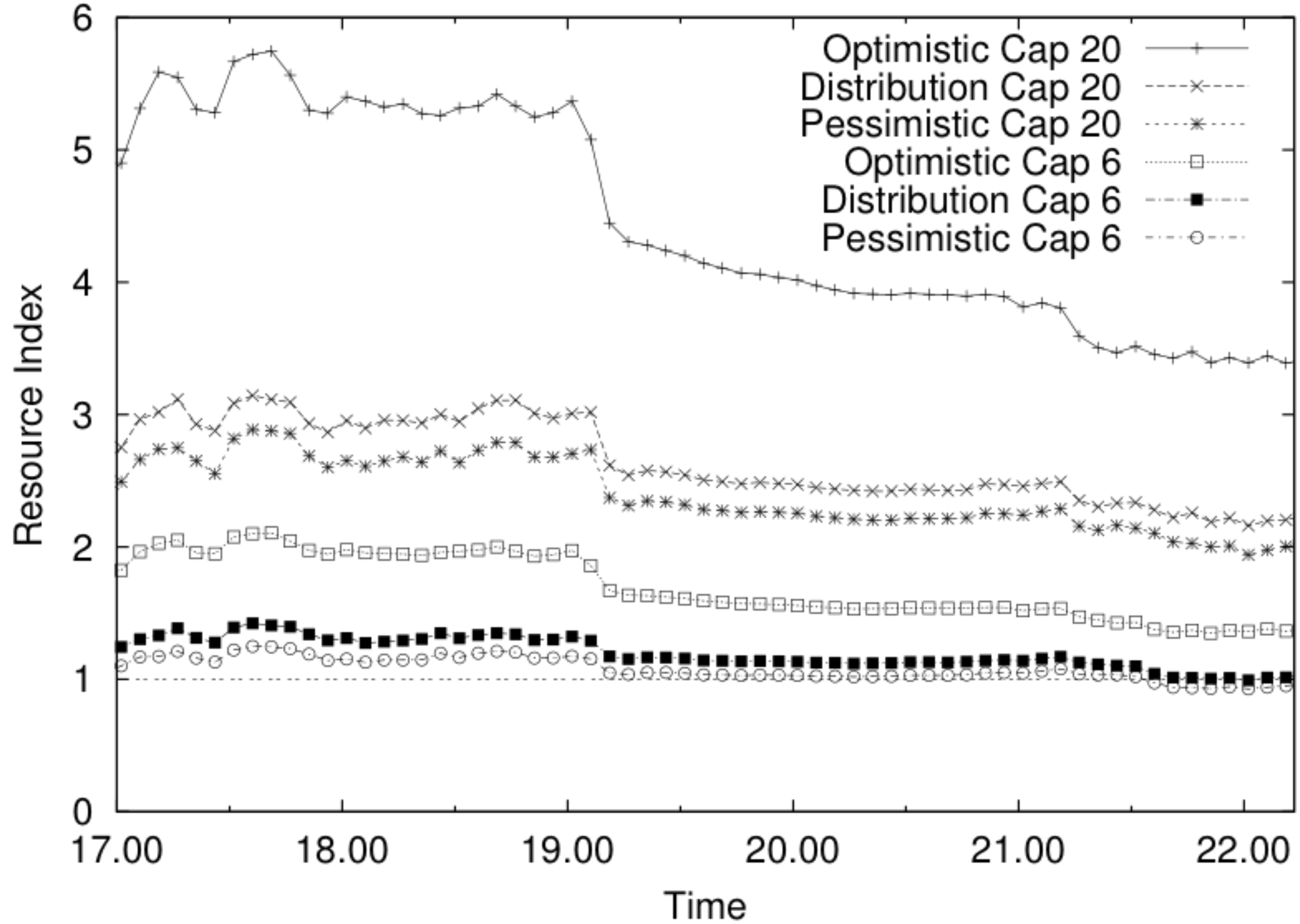
(3+5)/3 = 2.7

Metric: Resource index

- 10% unknown:
  - Optimistic,
  - Pessimistic (free-rider),
  - Distribution

- Degree is dependent on the encoding bit rate, so is the Resource Index

Single-Tree Protocol

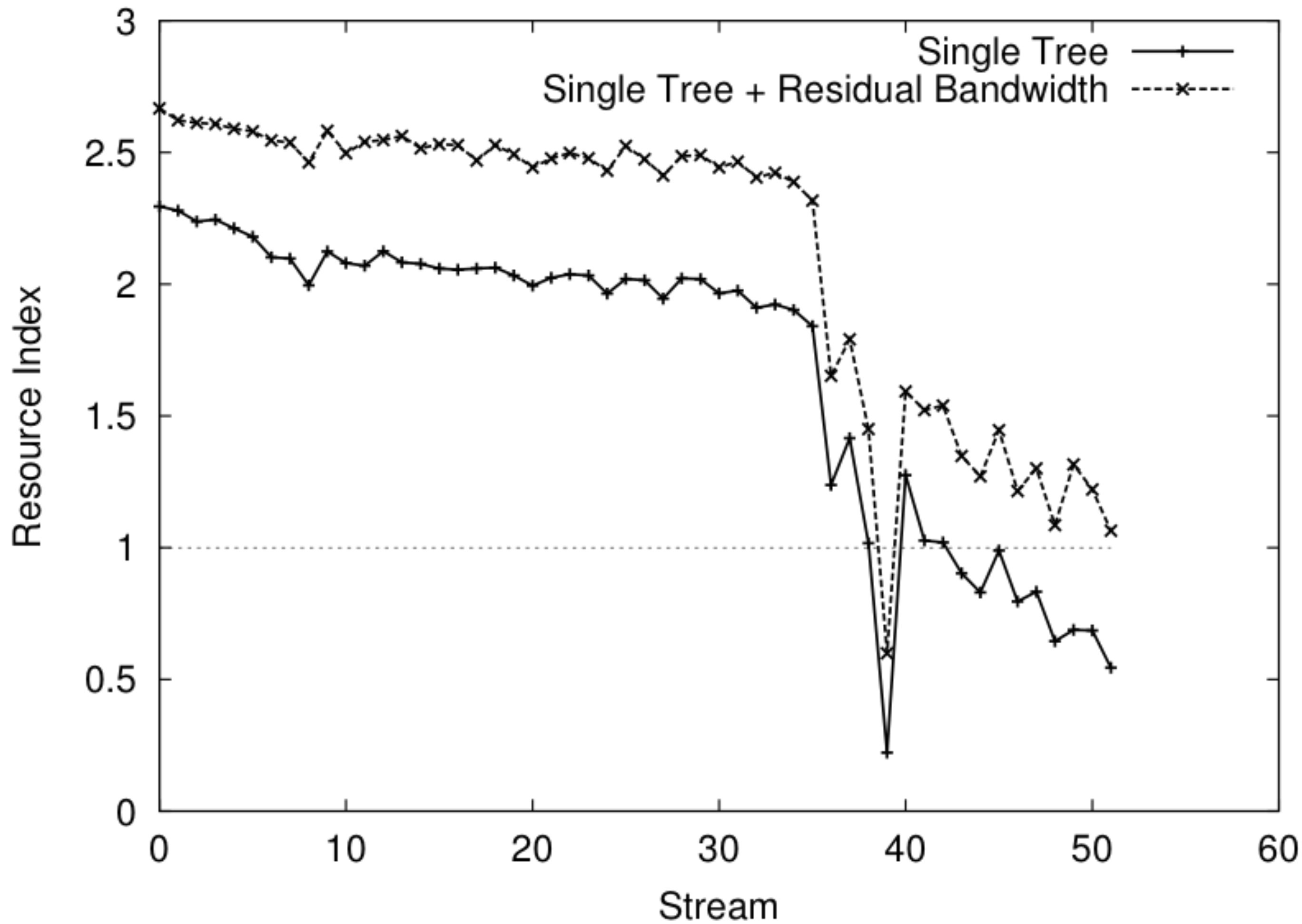


81 streams

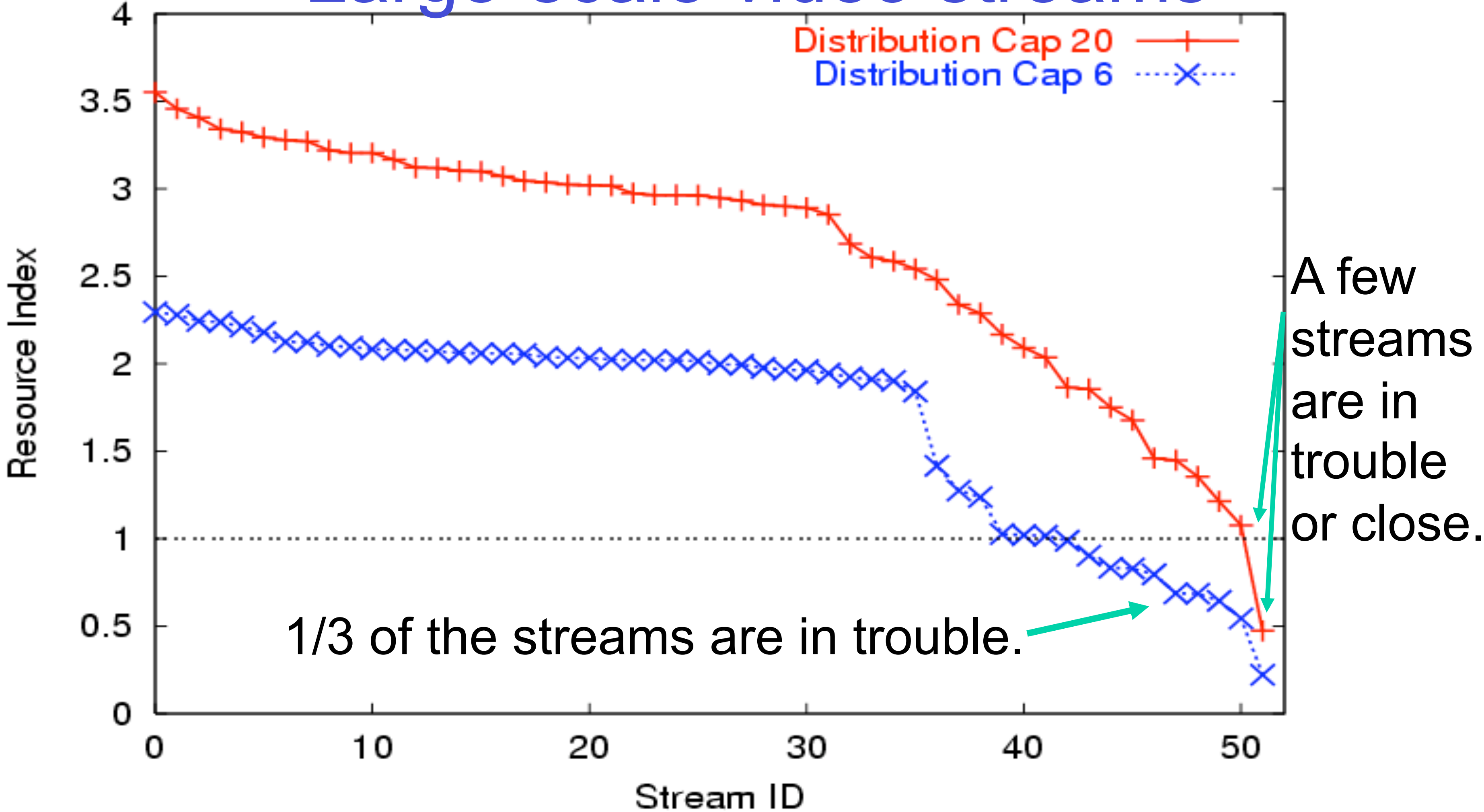
Resource index Multiple Trees

- Multiple Description Coding: Video stream is encoded into k independent sub-streams and distributed across k independent trees.
- Fractional supply: 250 kbps encoding split into 50 kbps sub-streams = 300/250 = 1.2 degree
- MDC:
  - Increases amount of resources,
  - Increases the feasibility of overlay multicast

Multiple-Trees Protocol



Large-scale video streams



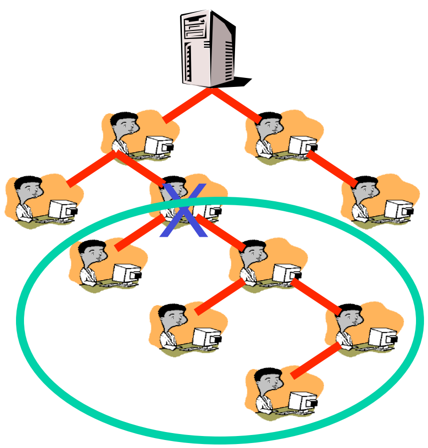
Most streams have sufficient upstream bandwidth.

Talk outline

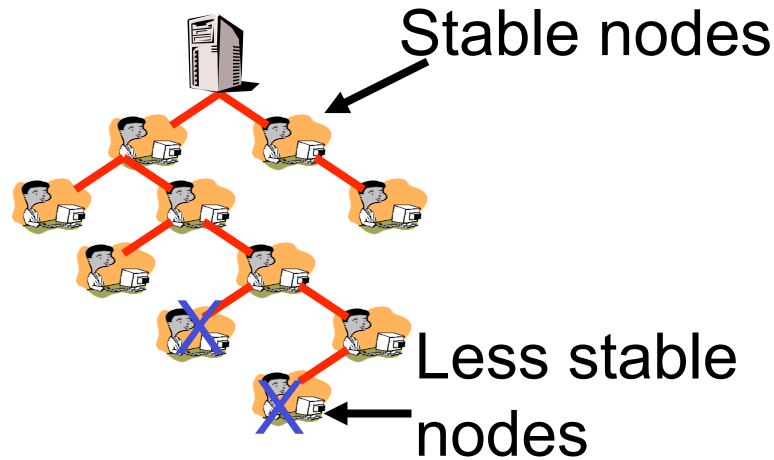
- Akamai live streaming workload
- With an application end-point architecture
  - Is there enough upstream bandwidth?
  - Is the overlay stable enough despite dynamic participation?
  - Are overlay structures efficient?
- Summary

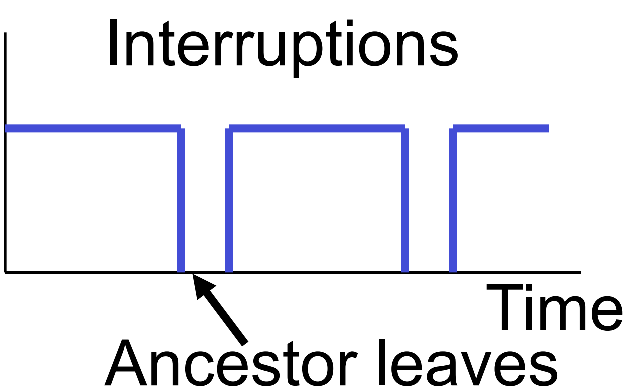
When is a tree stable?

Not stable



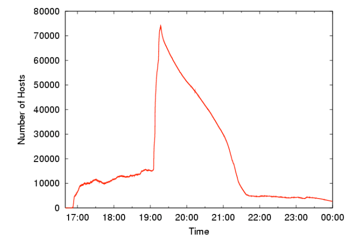
More stable

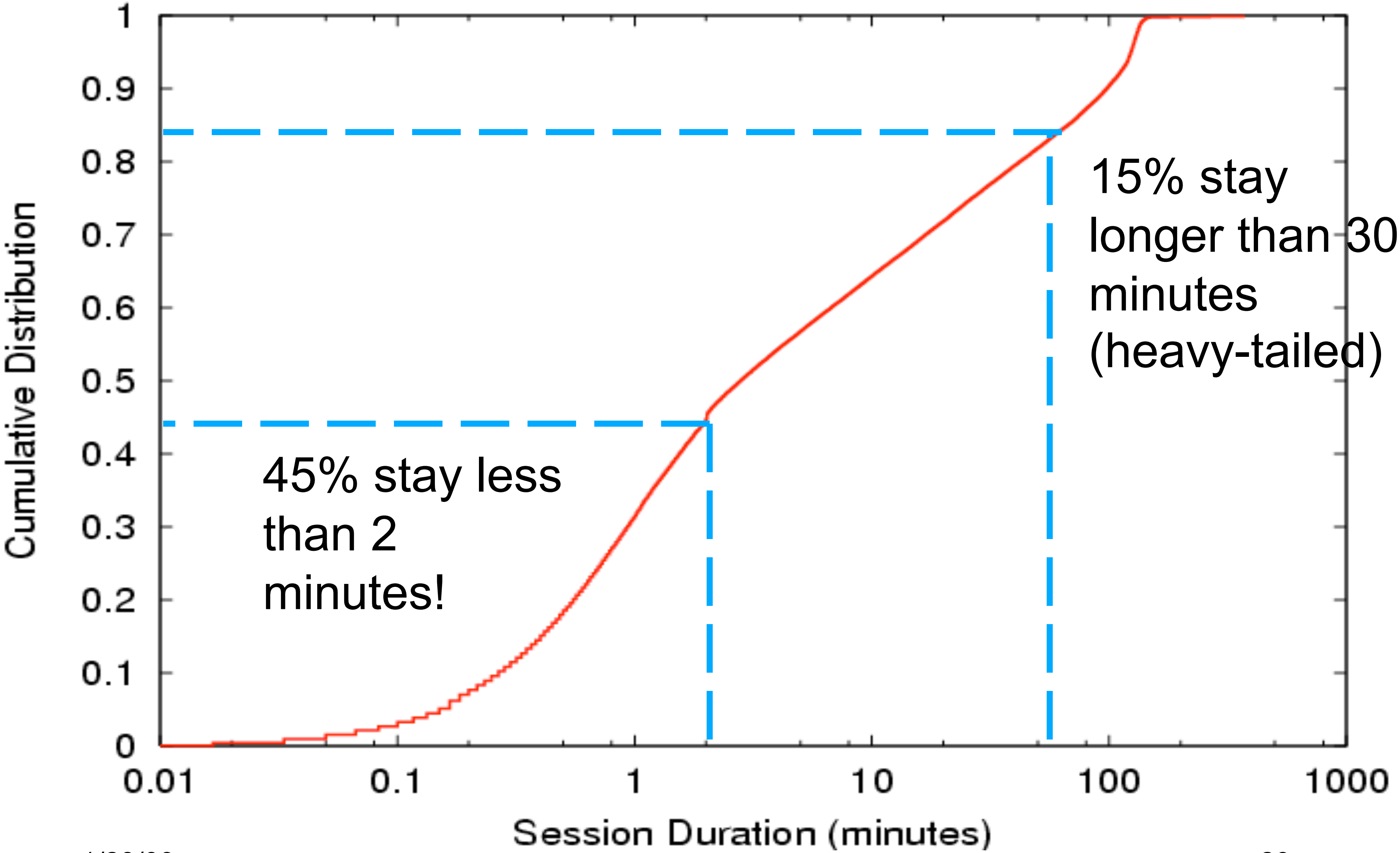




- Departing hosts have no descendants
- Stable nodes at the top of the tree

Extreme group dynamics

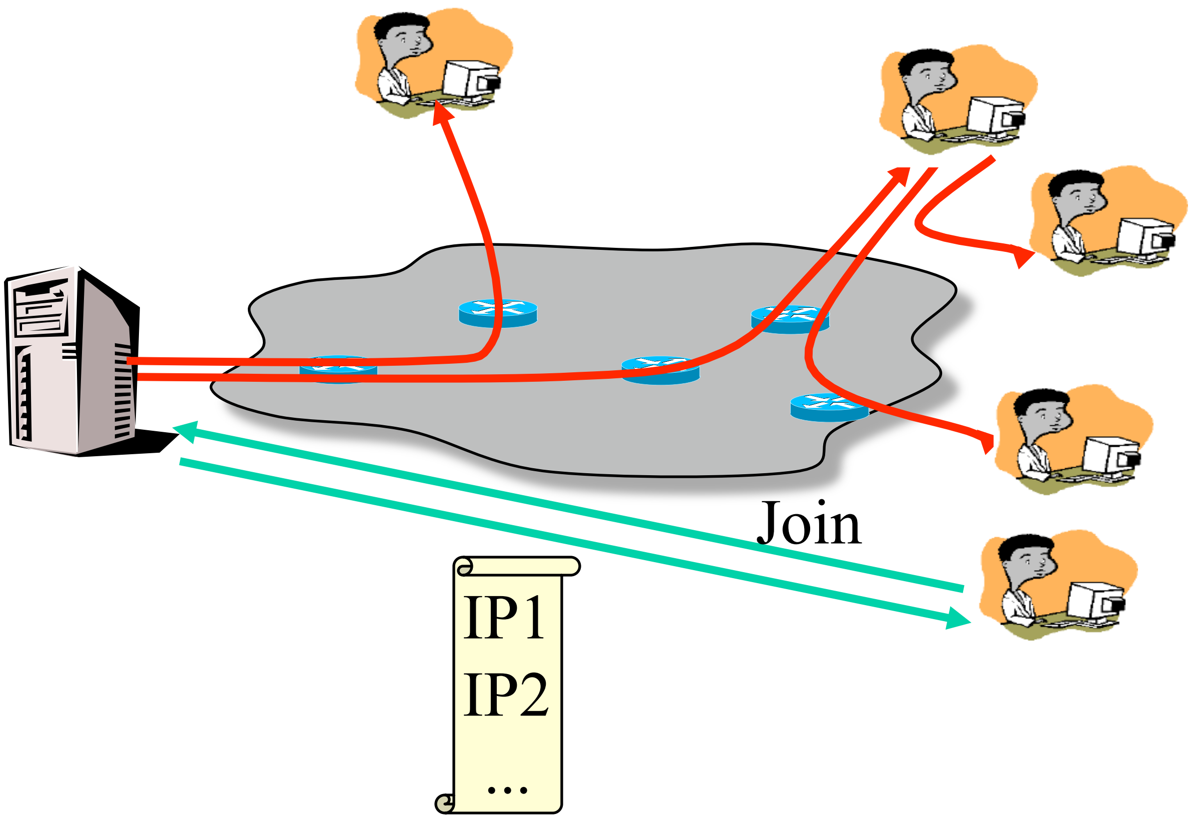




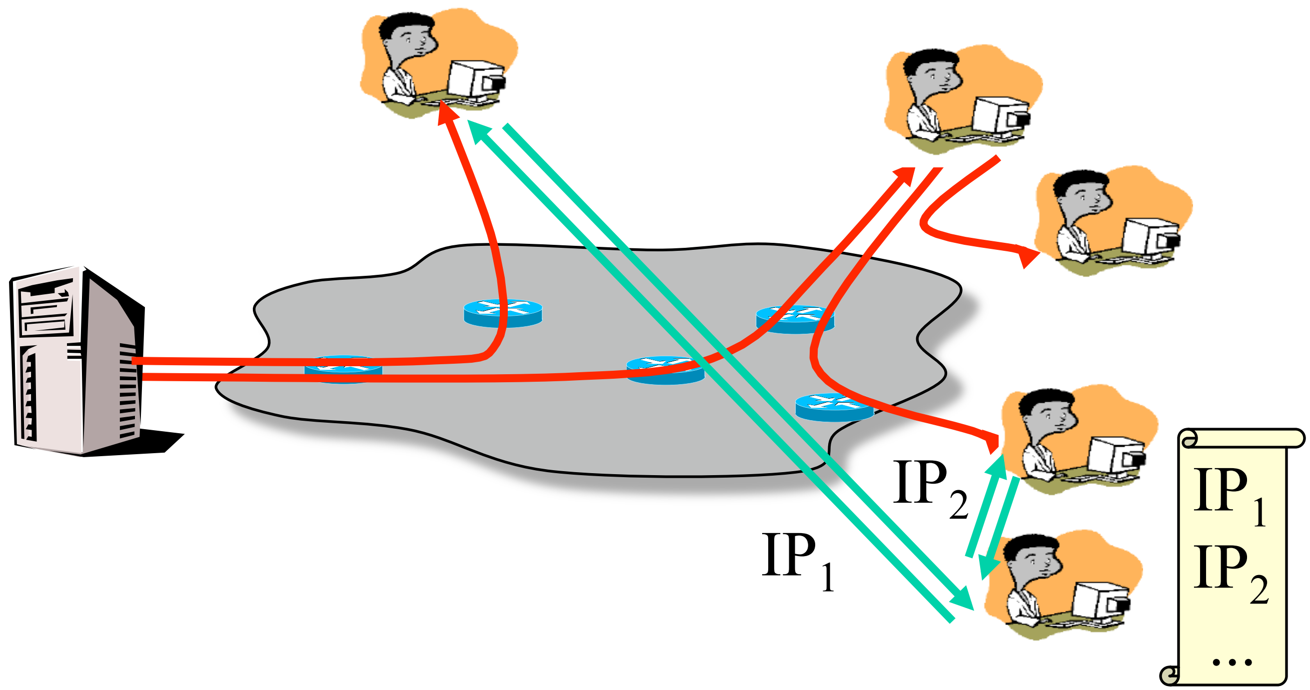
Stability evaluation: simulation

- Hosts construct an overlay amongst themselves using a single-tree protocol
  - Goal: construct a stable tree
    - Parent selection is key
- Group dynamics from Akamai traces (join/ leave)
- Honor upstream bandwidth constraints
  - Assign degree based on bandwidth estimation

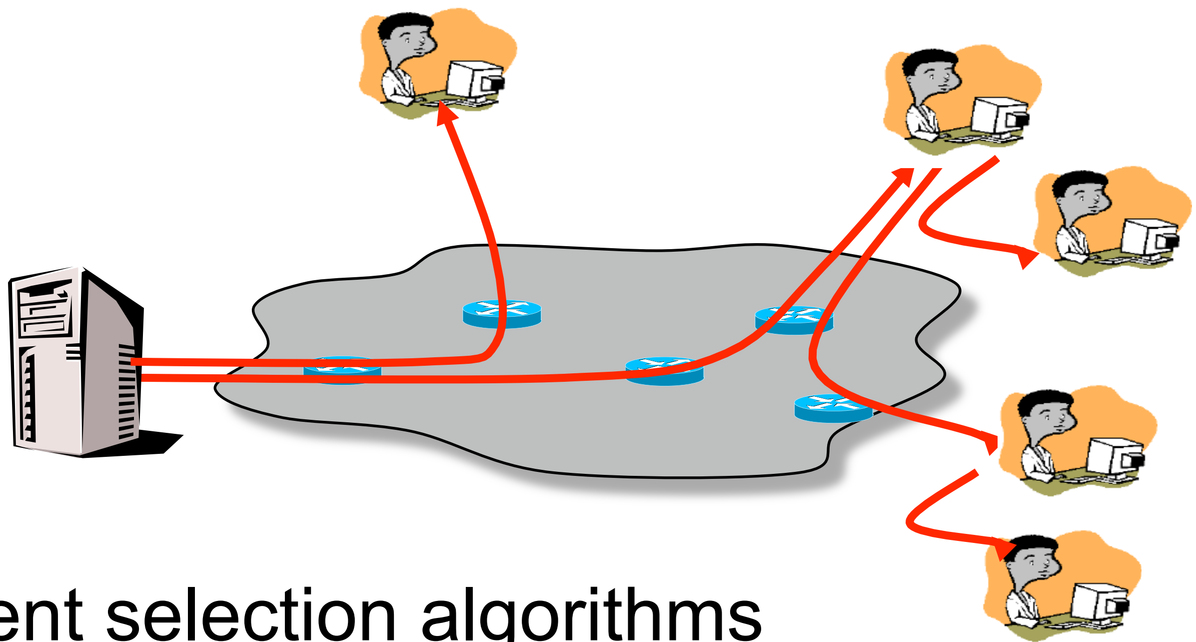
Overlay Protocol Simulation: Join



Probe and select parent



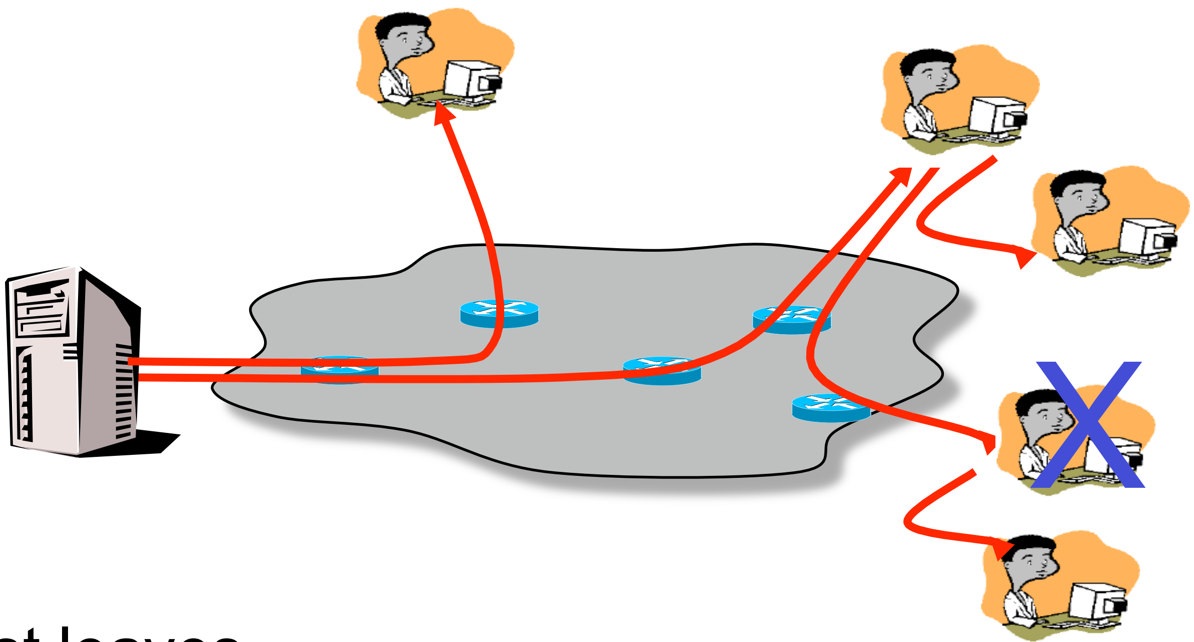
Probe and select parent



Parent selection algorithms

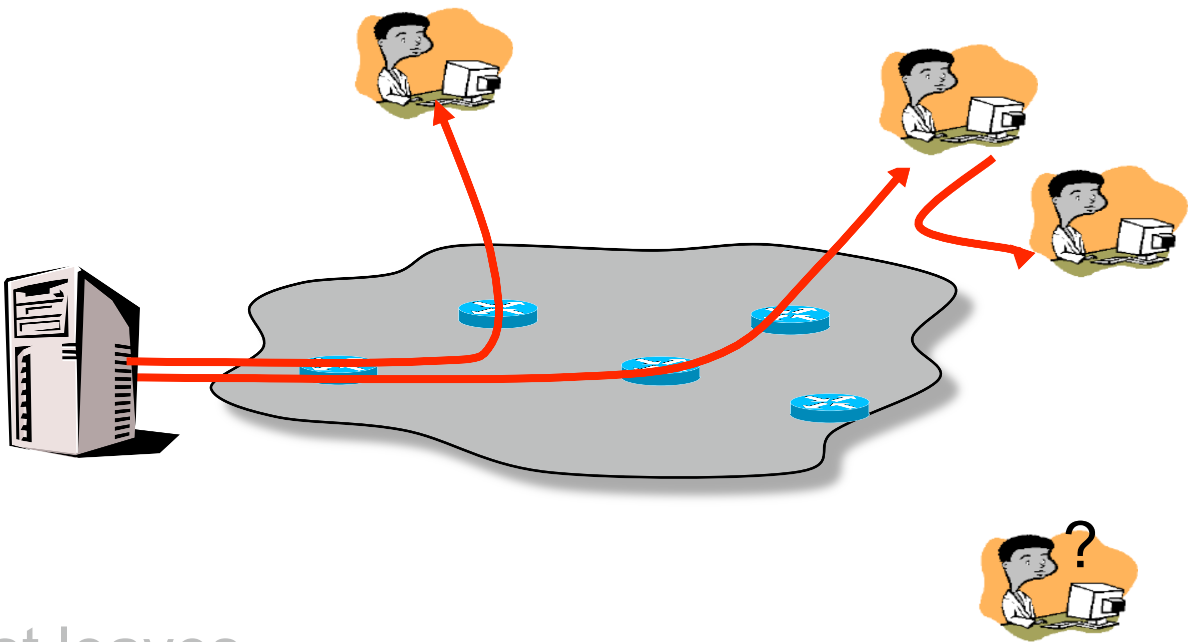
- Oracle: pick a parent who will leave after me
- Random
- Minimum depth (select one out of 100 random)
- Longest-first (select one out of 100 random)

Parent leave



Host leaves

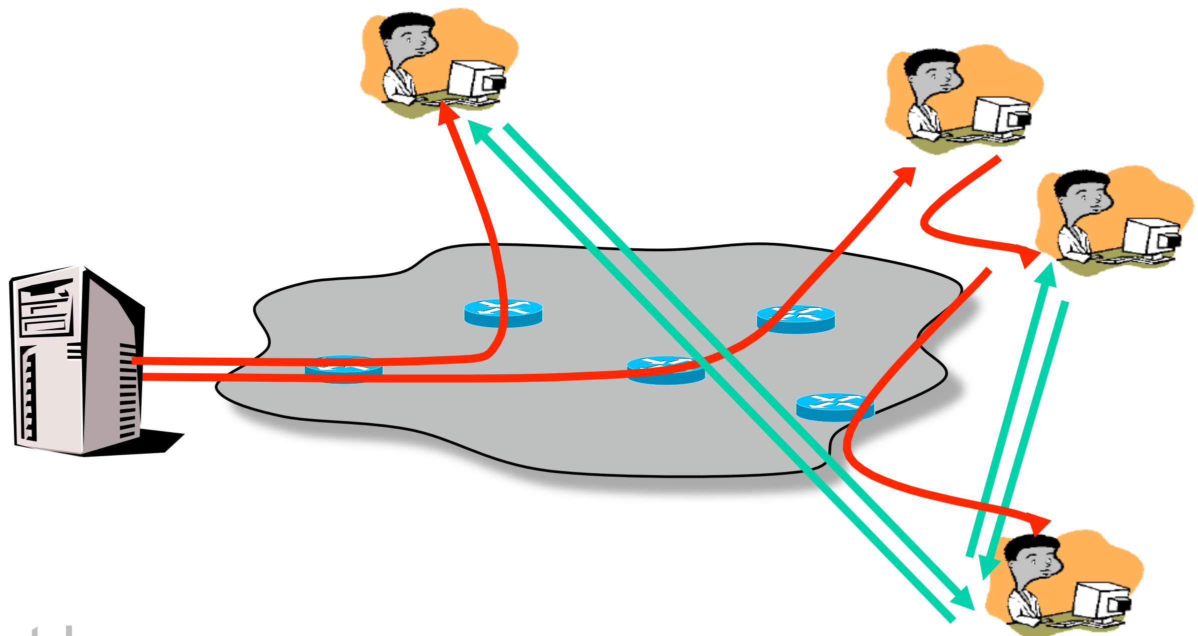
Parent leave



Host leaves

All descendants are disconnected

Find new parent



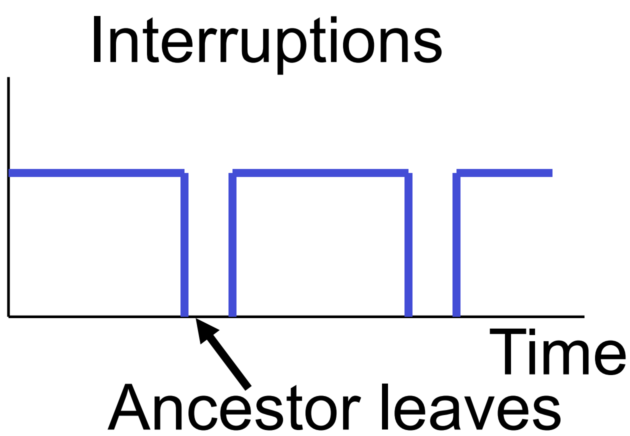
Host leaves

All descendants are disconnected

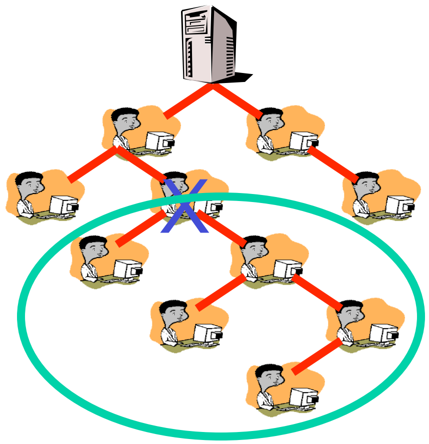
All descendants probe to find new parents

Stability metrics

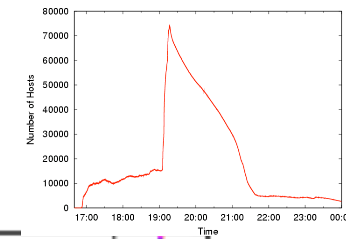
- Mean interval between ancestor change

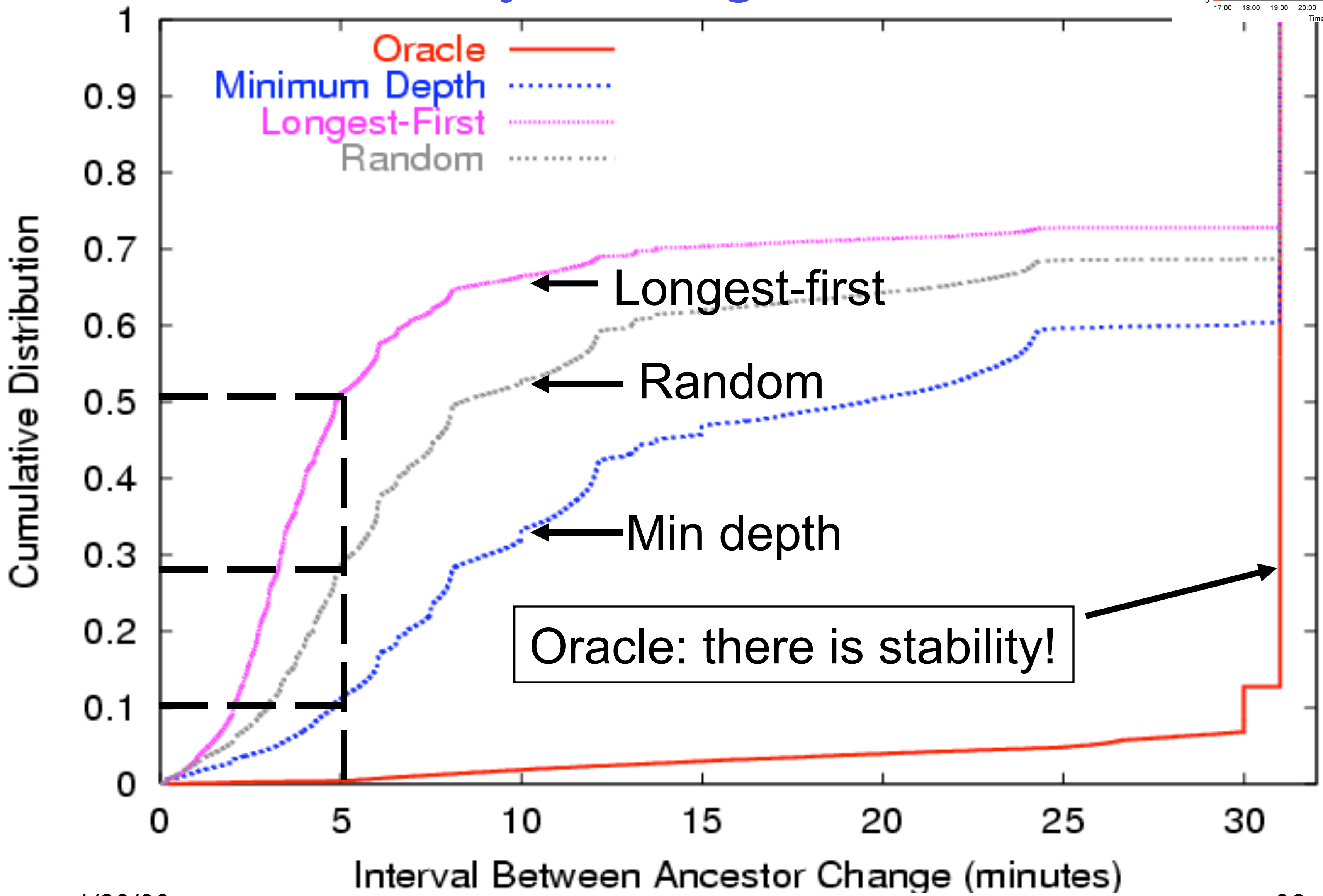


- Number of descendants of a departing host

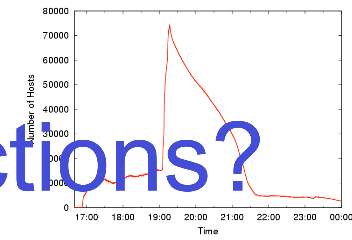


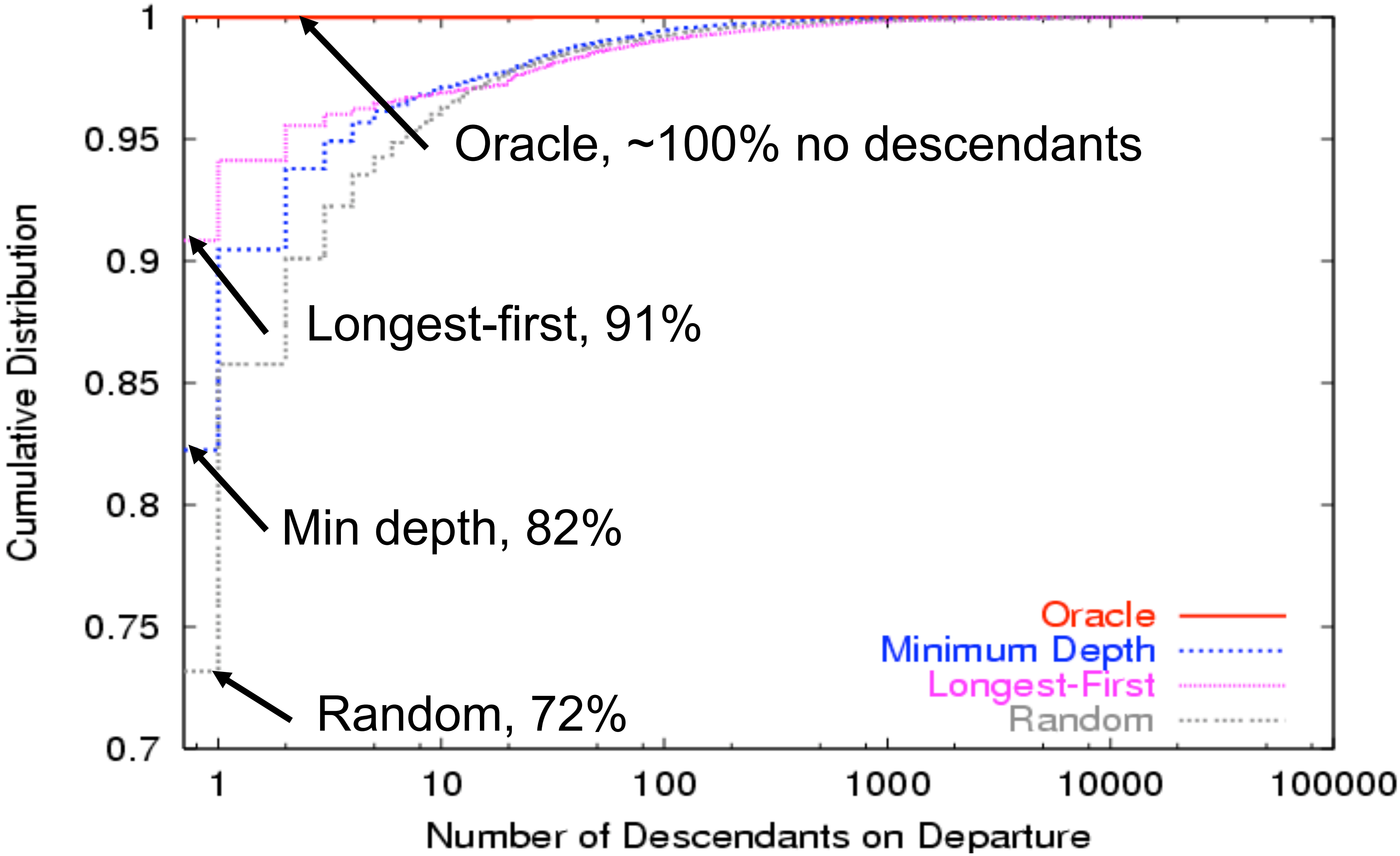
Stability of largest stream



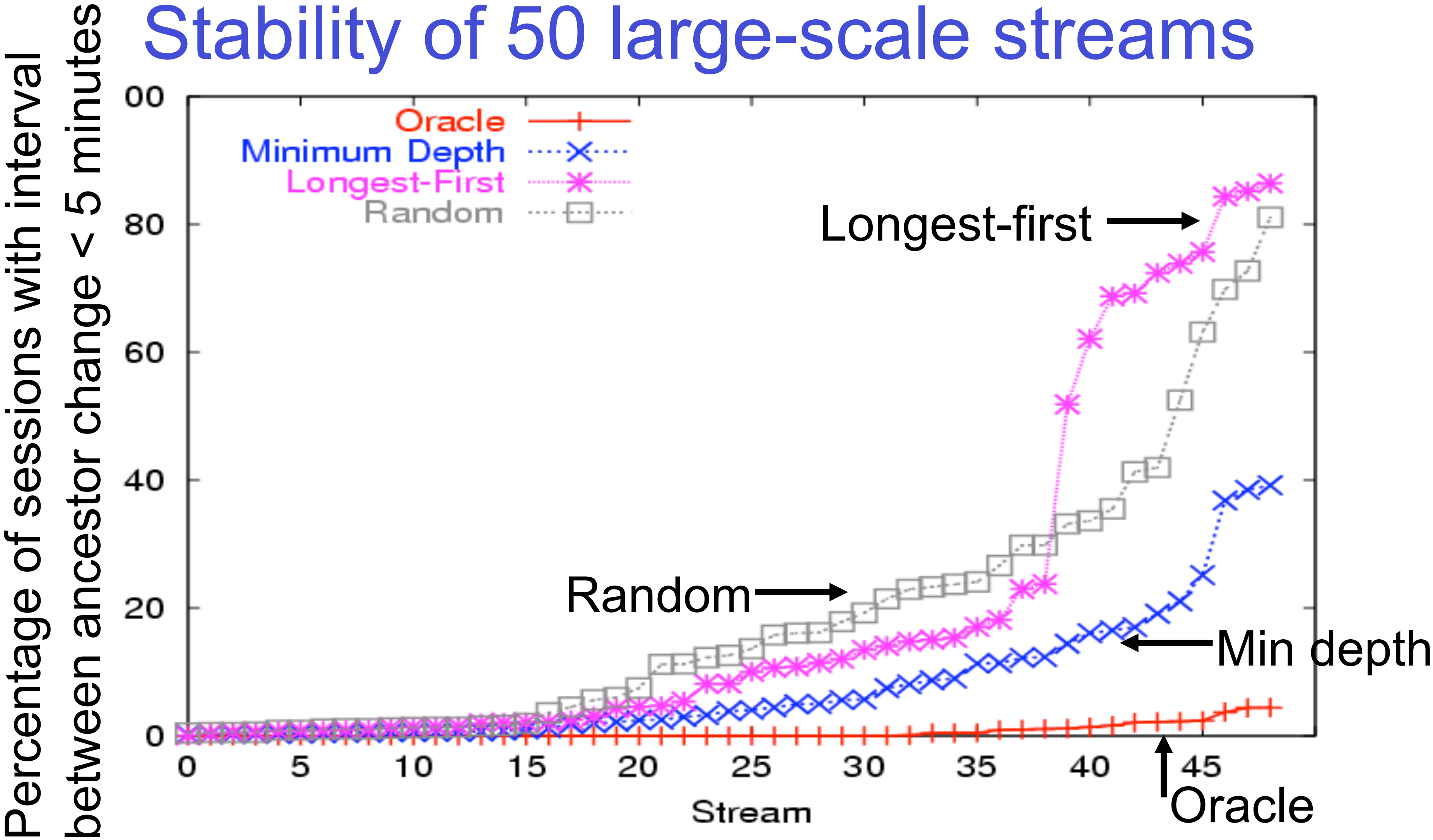


Is longest-first giving poor predictions?





Stability of 50 large-scale streams

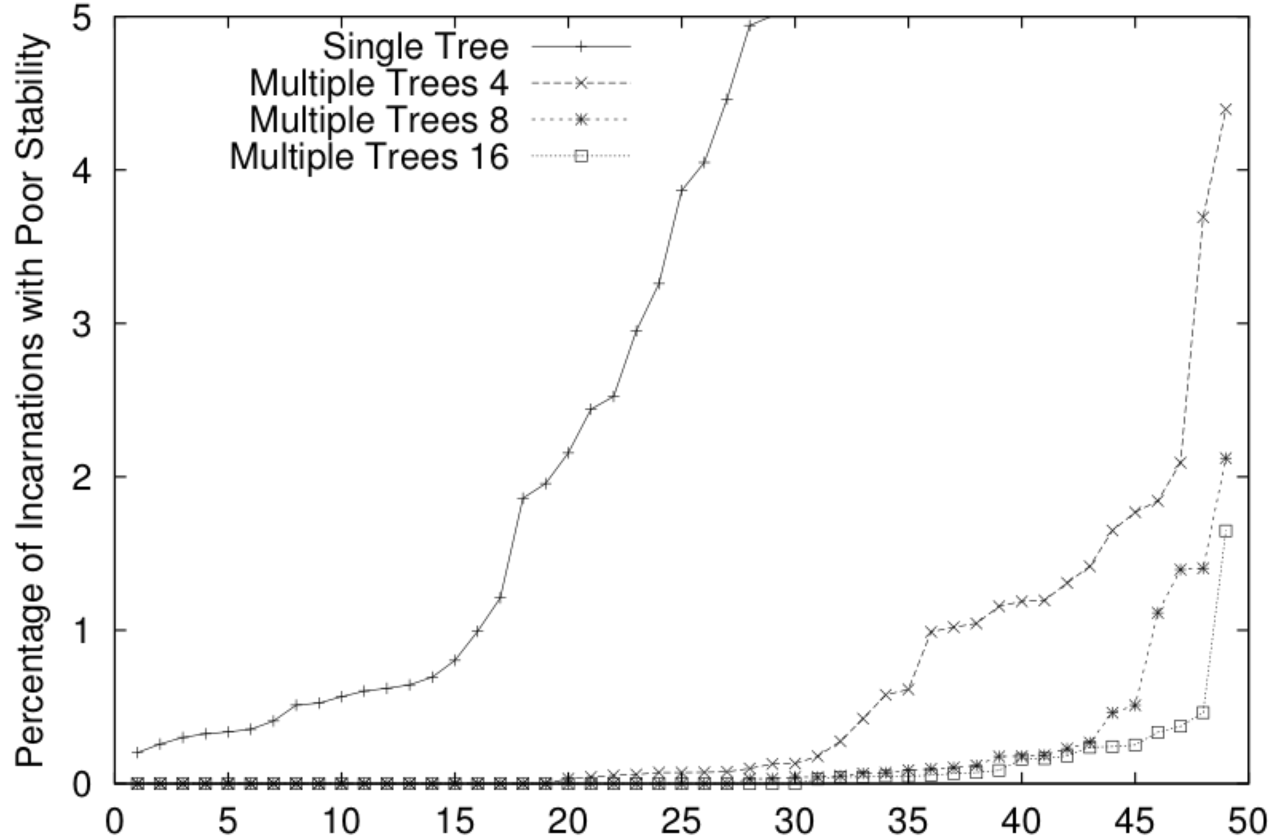


There is stability! Of the practical algorithms, min depth performs the best.

There is inherent stability

- Given future knowledge, stable trees can be constructed
- In many scenarios, practical algorithms can construct stable trees
  - Minimum depth is robust
  - Predicting stability (longest-first) is not always robust; when wrong, the penalty is severe
- Mechanisms to cope with interrupts are useful
  - Multiple trees

Stability Multiple Trees



Poor stability = being disconnected from at least 25% of the trees

There is inherent stability

- Multiple trees can increase the perceived quality of the streams but improved performance comes at a cost of more frequent disconnects, more protocol overhead and more complex protocol.

Talk outline

- Akamai live streaming workload
- With an application end-point architecture
  - Is the overlay stable enough despite dynamic participation?
  - Is there enough upstream bandwidth?
  - Are overlay structures efficient?
- Summary

Efficient Overlay

- Efficient overlay: one in which the overlay structure closely reflects the underlying IP network.
- The Challenge: to enable hosts to discover other nearby hosts that may be used as parents.
- Large number of hosts: prohibitive to know everyone else.
- Solution: partition end-points into clusters.

Cluster Membership

- Membership server: One member of each cluster is designated as the cluster head.
- Hosts in the same cluster maintain knowledge about one another.

Cluster Membership

- Handling host join: obtain list of member servers from rendezvous point
- Creating Membership servers: rendezvous point create servers on-demand as needed
- Recovering from membership server dynamics: before leaving a membership server looks to promote host inside cluster
- State maintenance: servers exchange state with the rendezvous point; among themselves and random set of hosts inside cluster.

Cluster policies

- Naïve clustering, 3 policies: Random, delay-based clustering, geographic clustering.
- Two critical requirements:
  - Cluster size (redirection, new cluster creation)
  - Resources within cluster (redirect free-riders)

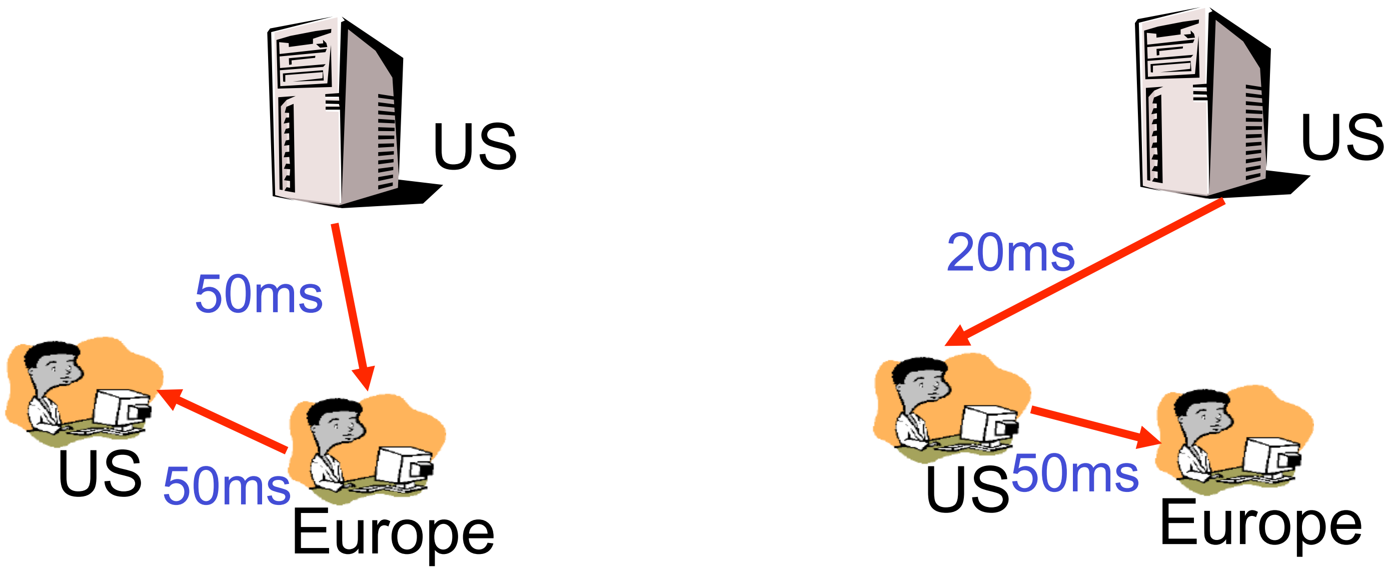
Cluster Quality

- Proximity Data, GNP: network delay, geographic distance

Relative Delay Penalty (RDP)

- How well does the overlay structure match the underlying network topology?

$$RDP = \frac{\text{Overlay distance}}{\text{Direct unicast distance}}$$



Results are more promising than previous studies using synthetic workloads and topologies.

Summary

- Indications of the feasibility of application end-point architectures
  - The overlay can be stable despite dynamic participation
  - There often is enough upstream bandwidth
  - Overlay structures can be efficient
- These findings can be generalized to other protocols

Thank you!