TCP-Friendly Traffic Engineering and Provisioning

Matchmaking providers and subscribers in the real world…

Talk Abstract

Cisco.com

- This talk is about the way that Network Providers and Subscribers can look at the big picture.
- Regard real traffic on the Internet, and provide some statistical performance guarantees.
- Understand the way that the traffic sources behave, and carry out appropriate provisioning.
- Develop evolutionary process that is future-proof against new applications.

Talk Outline

- A:The Mix
- B:The Sources
- C:Throughput SLA
- D: Delay SLA
- E:Mice&Elephants
- F:Multipath Routes
- G:Futures: P2P&GRID
- H:What to do about it?

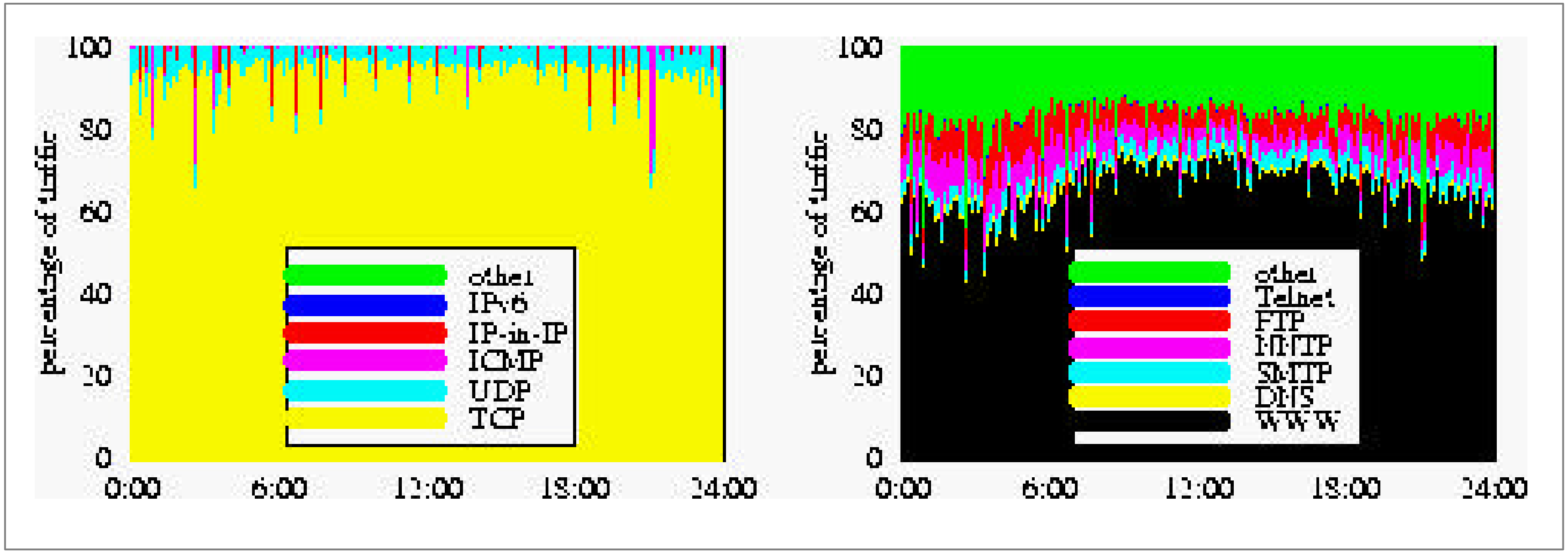
Aims and Objectives

- Subscriber wish:
- Squeeze as much capacity out of a provisioned service as possible for a given price, subject to delay constraints.
- Sites may be underspecified

- Provider wish:
- Squeeze as much income as possible out of a given subscriber set with a given network provisioning, subject to meeting SLAs
- Users may surprise!

A: The Backbone Traffic Mix

Transport Breakout　　TCP Applications



Source: MCI/NSF OC-3MON via http://www.nlanr.net, 1998

TCP Flow Statistics

Cisco.com

- >90% of sessions have ten packets each way or less

  Transactions - small web page – care about latency

- >70% of all TCP traffic results from <10% of the sessions, in high rate bursts

  Large transfers – mirror – care about throughput

TCP Friendly Service Level Agreement

- **IP/Network Level SLA is not Enough**

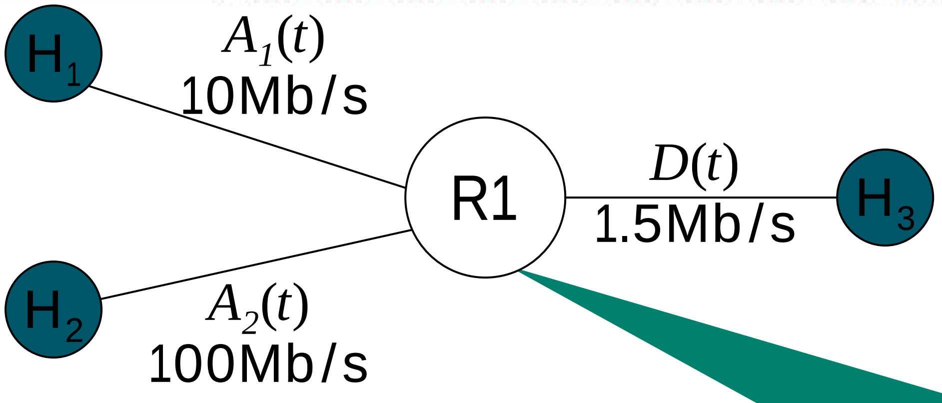
1. **Bulk TCP session/application wants at least a certain bandwidth, or else:**
2. **User Experience: Web Download “completion date”.**

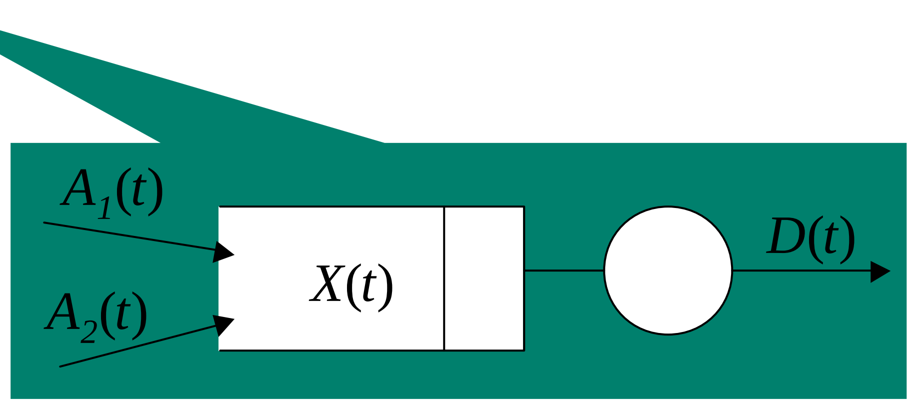
- **Contrast Telephony: call block probability or phone call “commence time”**

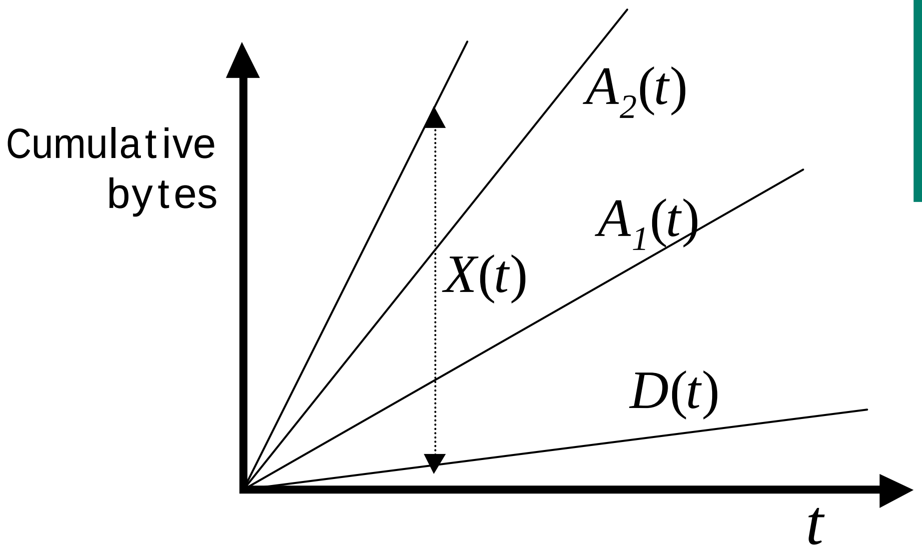
B. The Sources

- Congestion is inevitable
- TCP sources detect congestion and, co-operatively, reduce the rate at which they transmit.
- The rate is controlled using the TCP window size.
- TCP modifies the rate according to “Additive Increase, Multiplicative Decrease (AIMD)”.
- To jump start flows, TCP uses a fast restart mechanism (called “slow start”!).
- TCP achieves high throughput by encouraging high delay.

Congestion


H1
A1(t)
10Mb/s
R1
D(t)
1.5Mb/s
H3
H2
A2(t)
100Mb/s


A1(t)
A2(t)
X(t)
D(t)


Cumulative bytes
A2(t)
A1(t)
X(t)
D(t)
t

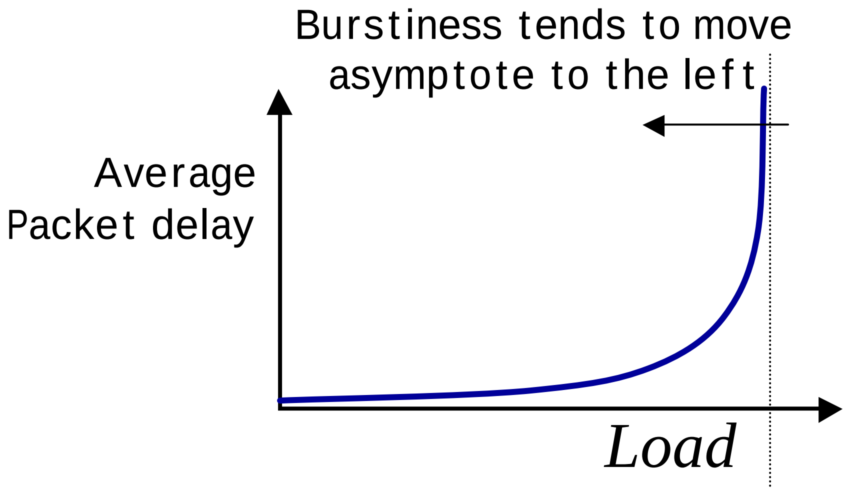
Congestion is unavoidable

Arguably it's good!

- **We use packet switching because it makes efficient use of the links. Therefore, buffers in the routers are frequently occupied.**
- **If buffers are always empty, delay is low, but our usage of the network is low.**
- **If buffers are always occupied, delay is high, but we are using the network more efficiently.**
- **So how much congestion is too much?**

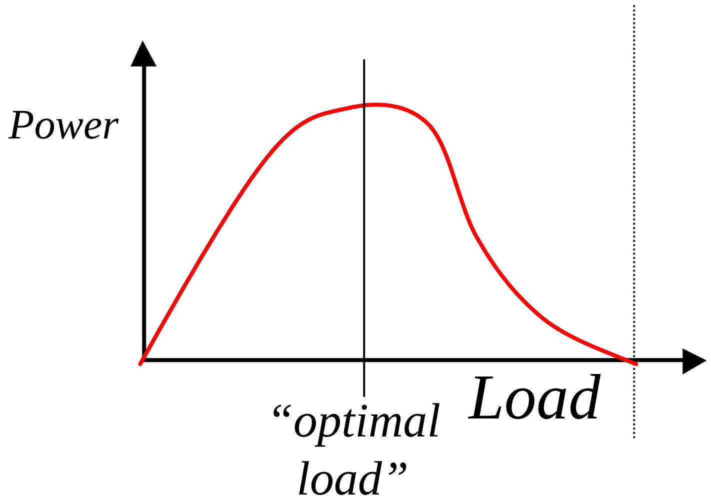
Load, delay and power

Typical behavior of queueing systems with random arrivals:



A simple metric of how well the network is performing:

$$Power \ ? \ \frac{Load}{Delay}$$



Options for Congestion Control

1. Implemented by host versus network
2. Reservation-based, versus feedback-based
3. Window-based versus rate-based.

TCP Congestion Control

- TCP implements host-based, feedback-based, window-based congestion control.
- TCP sources attempts to determine how much capacity is available
- TCP sends packets, then reacts to observable events (loss).

TCP Congestion Control

Cisco.com

- **TCP sources change the sending rate by modifying the window size:**

  Window = min{Advertized window, Congestion Window}

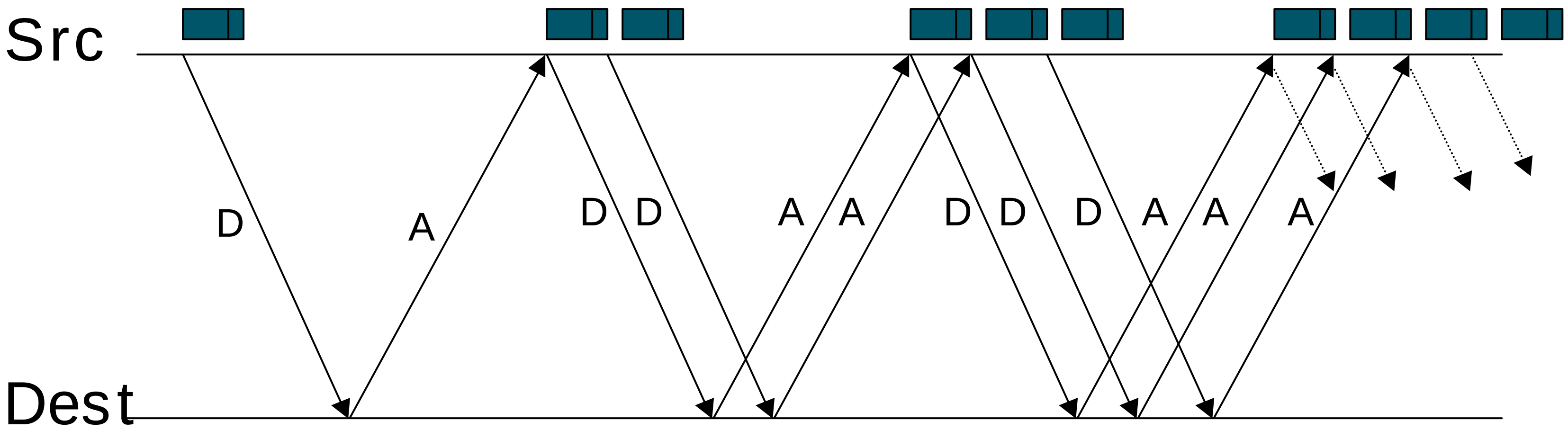
  Receiver Transmitter ("cwnd")

- **In other words, send at the rate of the slowest component: network or receiver.**
- **"cwnd" follows additive increase/multiplicative decrease**

  On receipt of Ack: cwnd += 1/cwnd

  On packet loss (timeout): cwnd *= 0.5

Additive Increase

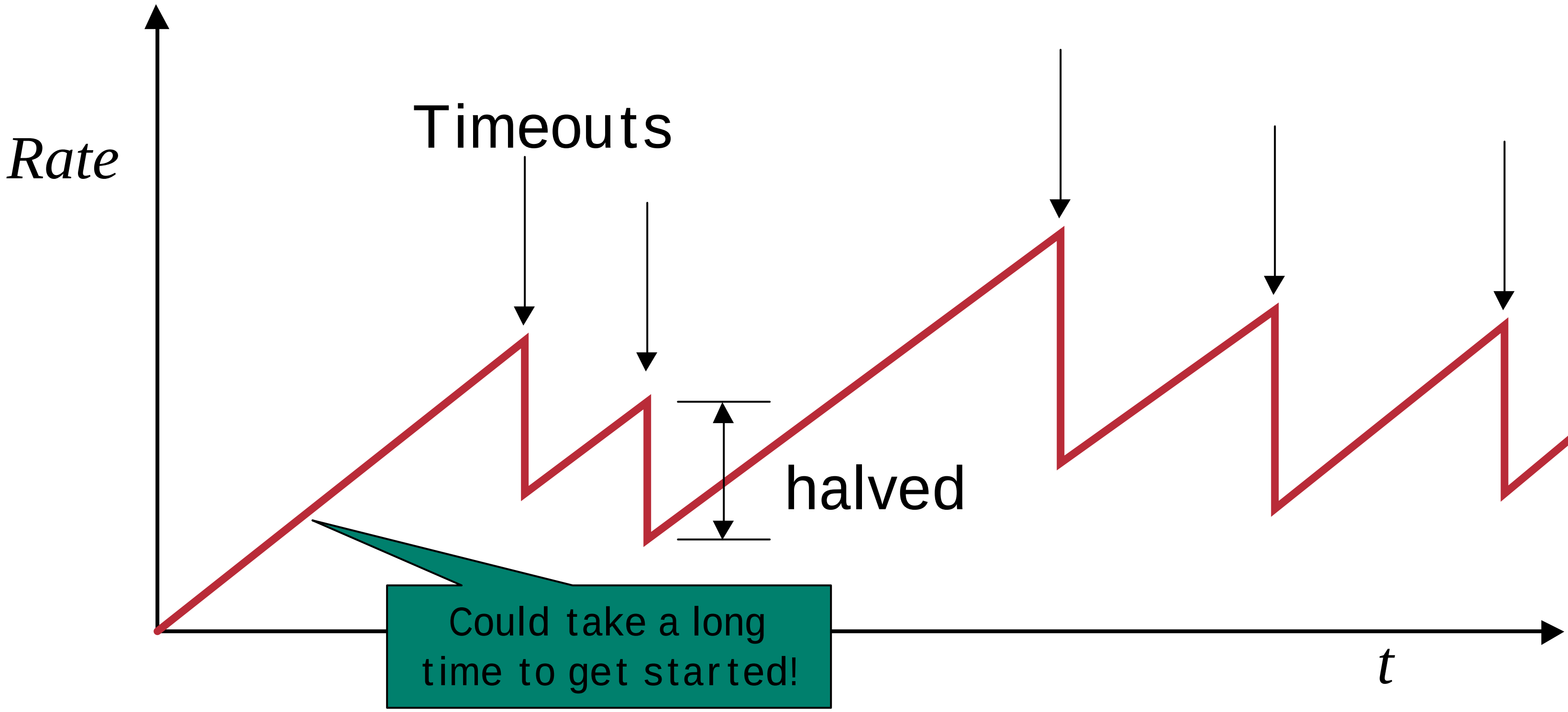


Actually, TCP uses bytes, not segments to count:
When ACK is received:

$$cwnd \mathrel{+}= MSS \left(\frac{MSS}{cwnd} \right)$$

Leads to the TCP “sawtooth”

Cisco.com

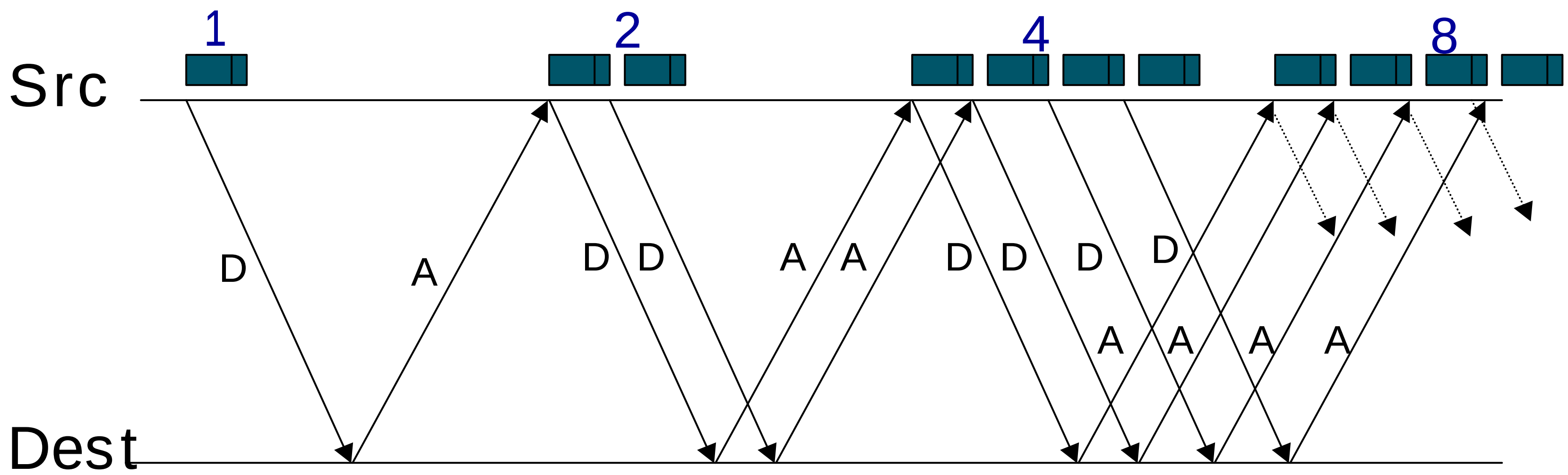


"Slow Start"

Cisco.com

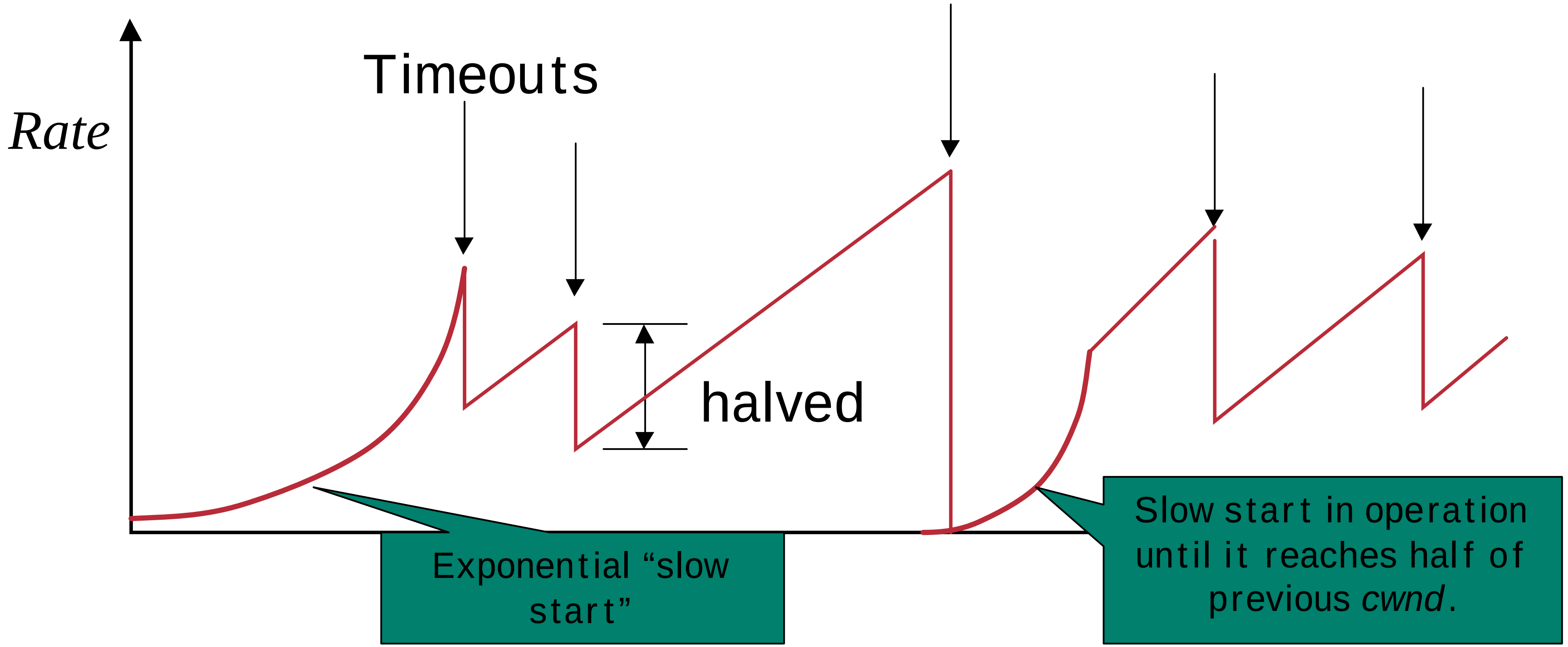
Designed to cold-start connection quickly at startup or if a connection has been halted (e.g. window dropped to zero, or window full, but ACK is lost).

How it works: increase cwnd by 1 for each ACK received.



Slow Start

Cisco.com



Why is it called slow-start? Because TCP originally had no congestion control mechanism. The source would just start by sending a whole window's worth of data.

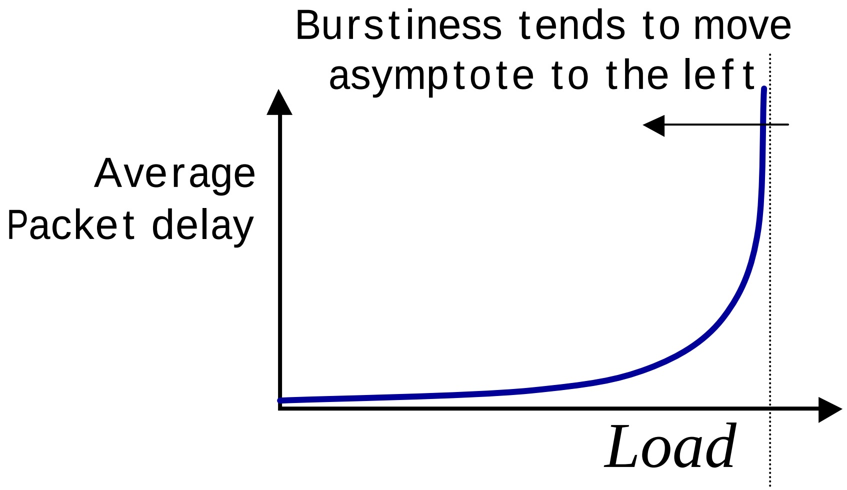
Fast Retransmit & Fast Recovery

- TCP source can take advantage of an additional hint: if a duplicate ACK arrives out of sequence, there was probably some data lost, even if it hasn't yet timed out.
- Upon 3 duplicate ACKs, TCP retransmits.
- Does not enter slow-start: there are probably ACKs in the pipe that will continue correct AIMD operation.

Where does TCP operate?

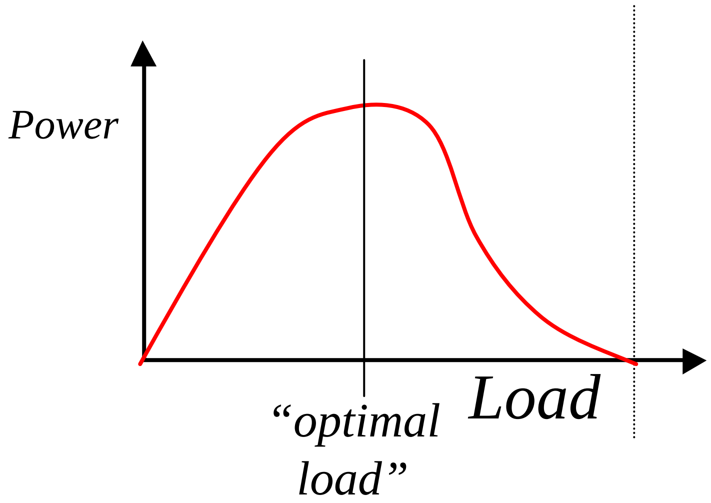
Cisco.com

Typical behavior of queueing systems with random arrivals:



A simple metric of how well the network is performing:

$$Power\ ?\ \frac{Load}{Delay}$$



Therefore—TCP QoS Definition:

- **Normally at most one drop per round trip**
- **Mean variation in latency bounded by predictable network**

C. Throughput SLA

Cisco.com

- **Mix is dominated by TCP**
- **Mice & Elephants split 50/50**
- **Throughput of Elephants determines loss for Mice (&Elephants) < 1/RTT**
- **Loss determines E2E delay for Mice**

TCP Throughput Equations

Cisco.com

- **For Long Lived Transfer:**

  TCP Send Rate = k/[rtt * sqrt(packet loss)]

- **For Short Downloads:**

  TCP Send Rate = nk/[rtt * 2]

- **K is packet size, n number of packets in short download, rtt is round trip time…**

Given *Network* SLA

- **Above Transport Service, we still need an SLA**
- **Lets say the IP level specifies**
  - **Throughput**
  - **Availability**
- **But the IP level also has a packet loss probability – can work out what that is?**

Loss Concealment Cost, not an option!

- We have 2 transport techniques for loss concealment
- whether random noise induced, or much more commonly, congestive packet loss
- note that congestive loss doesn't mean congestion (c.f. fast retransmit)
- only persistent loss does….occasional loss is a rate feedback mechanism

1. Retransmission (TCP, PGM)
2. Forward erasure/error correction (e.g. in PGM)

Costs of Loss Concealment

Cisco.com

- **Retransmitted packet still has independent loss probability – hence expected mean delay for packets is (assuming binomial back-off for subsequent retransmits!):**

  E(mean delivered packet delay) =

  Sum over i to infinity of rtt*2^i * (1-p)^i

  Luckily for us, this converges for small p!

  Can compute delay variance similarly….

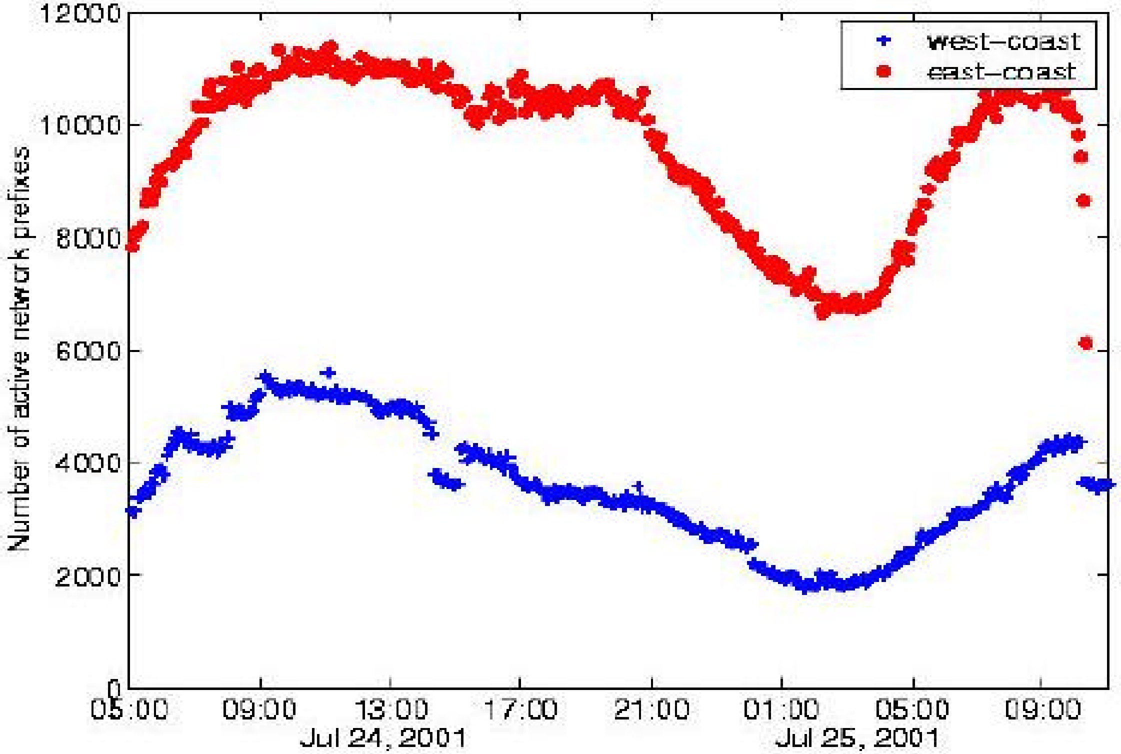
- **FEC has no delay penalty: adds fixed delay at source + takes a percentage network overhead**
- **E(txput) = 1 + p + epsilon**

E:Lets look at some Mice& Elephants data

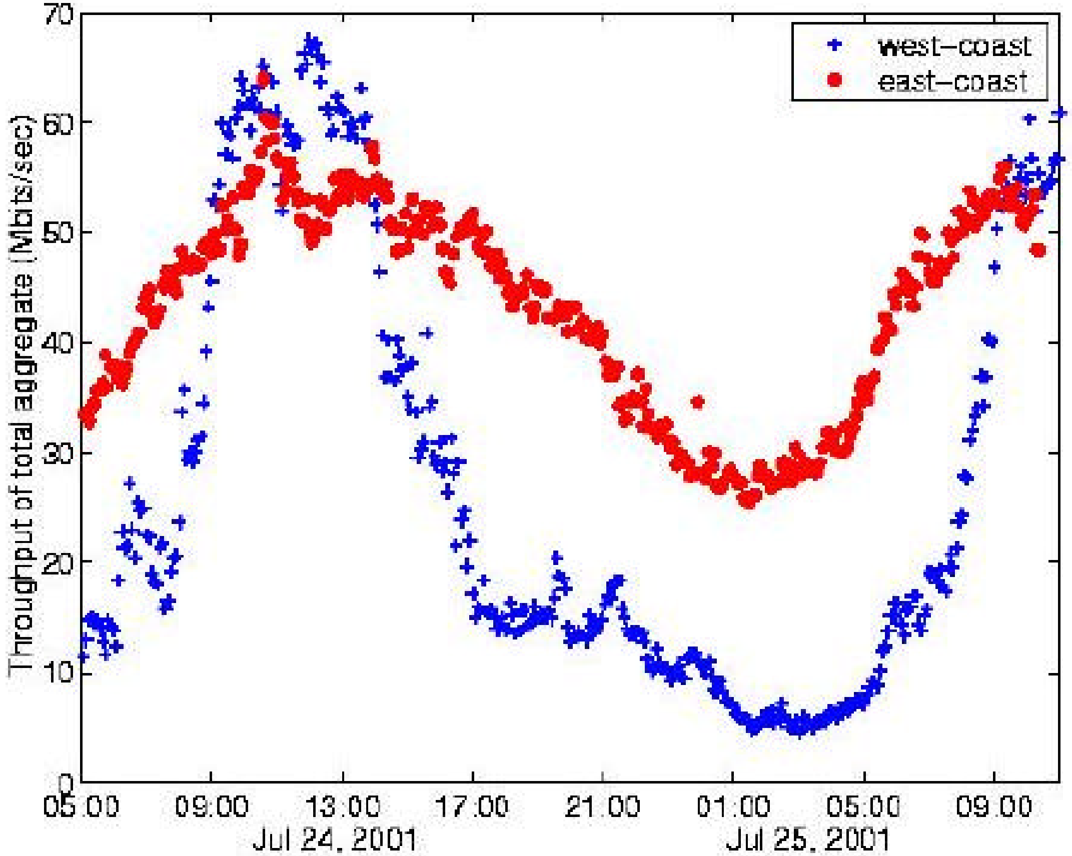
- Papagiannaki&Diot looked at the Sprint core inter-pop traffic
- Represents about 10 man years effort
- Their Goal: provisioning/prediction/protection

Number of active prefixes through the day

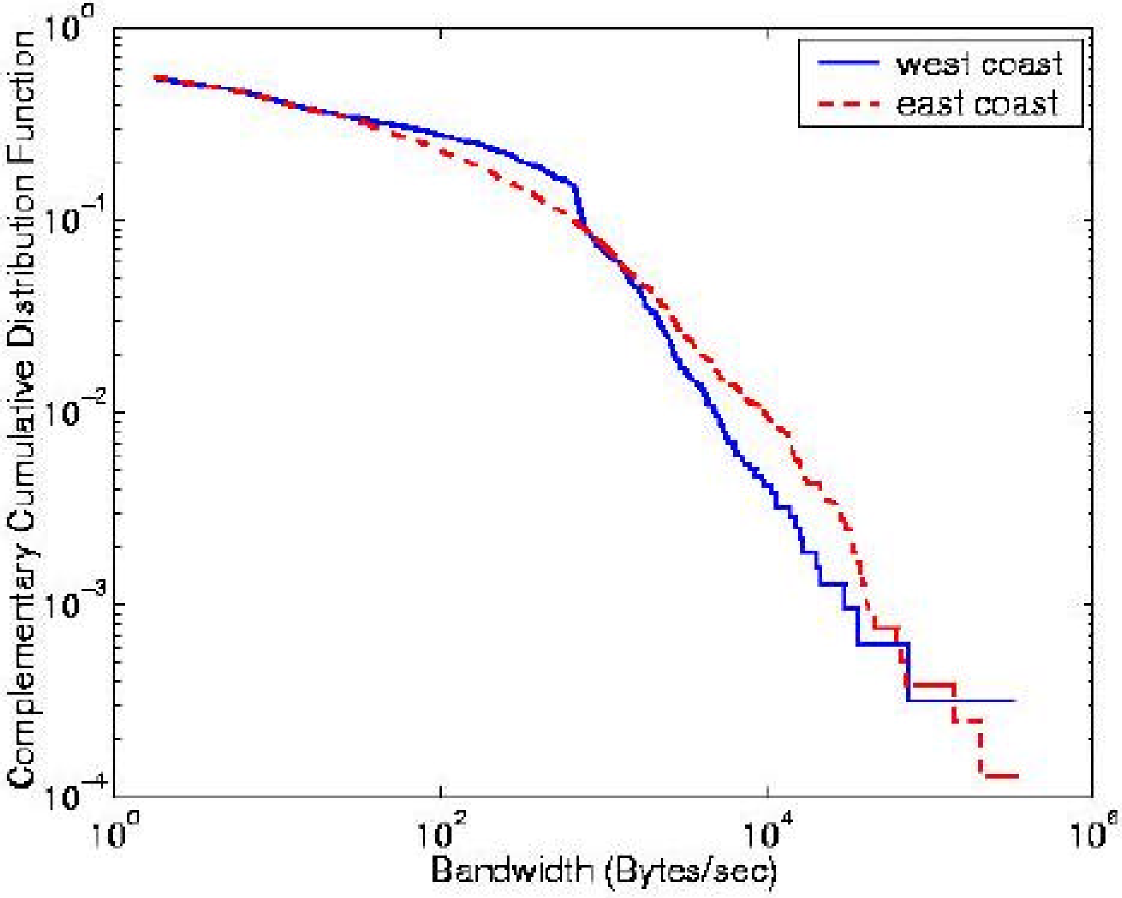
Cisco.com



Aggregate Throughput throughout the day



Who makes up how much of the aggregate

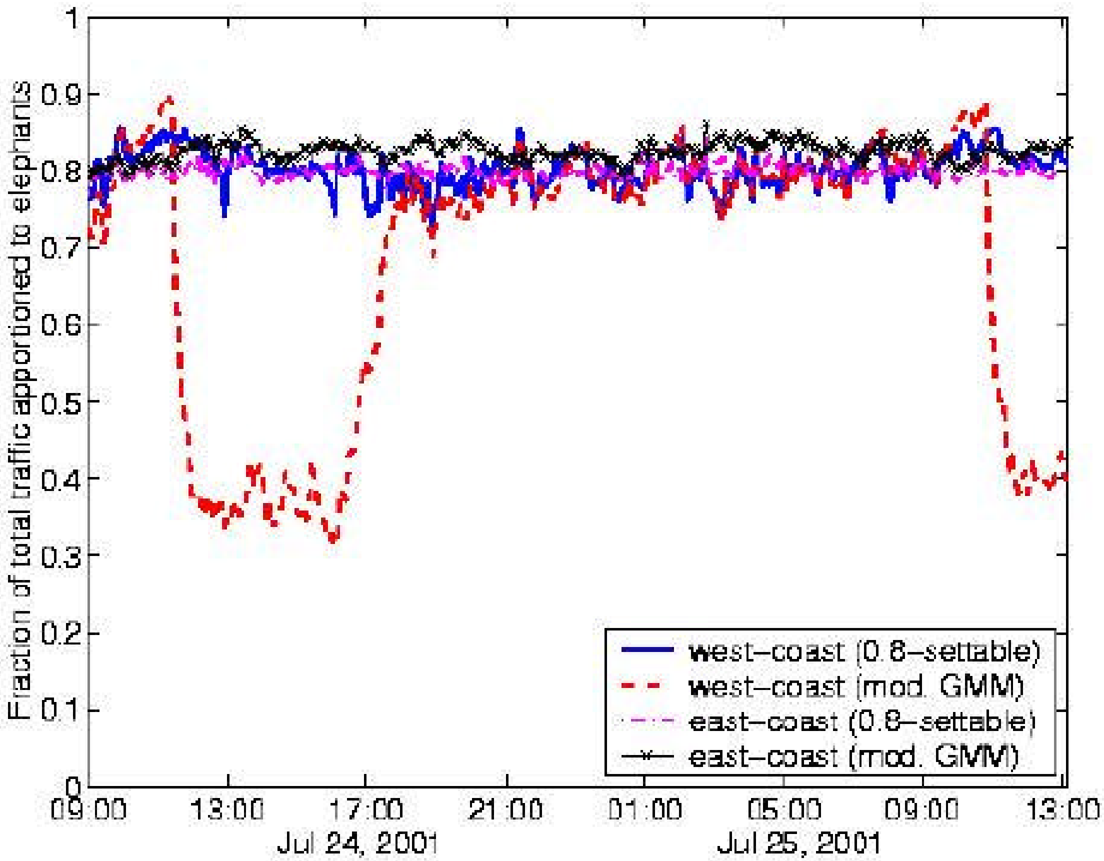


Try to find mice/elephants

Cisco.com

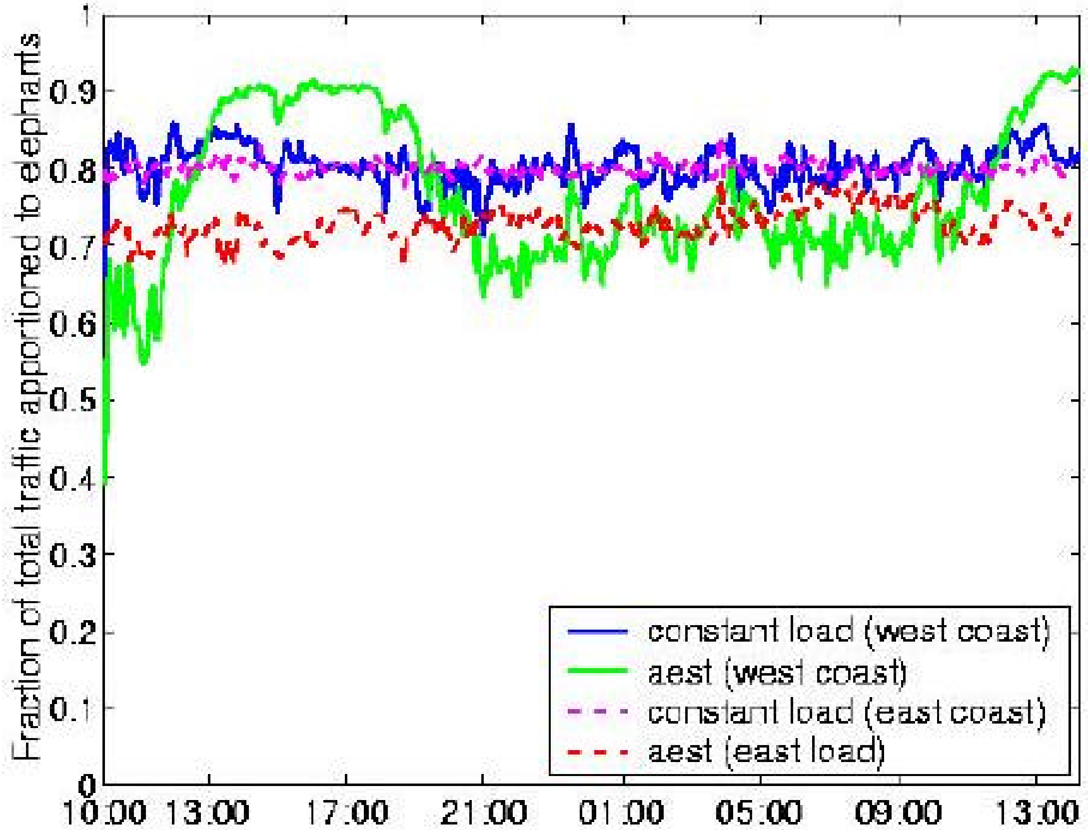
- First try simple threshold (byte/packet rate from/to prefix pair about a certain amount)
- Next try Markov Model
- Its actually quite tricky, but in the end can get a reasonable match
- Why? Allows automatic placement of mice on low delay and elephants on high capacity paths (or MPLS FEC or DiffServe Queues)

Hunting ELephants



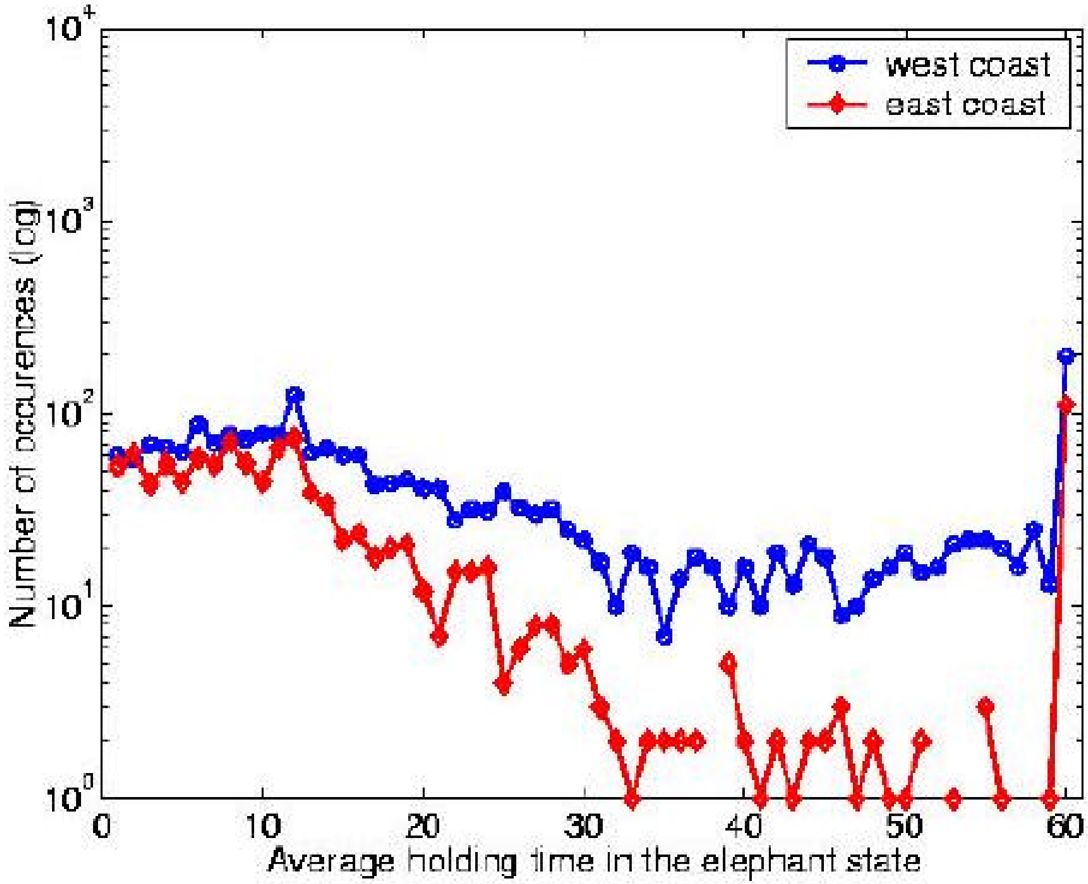
How heavy are the elephants?

Cisco.com



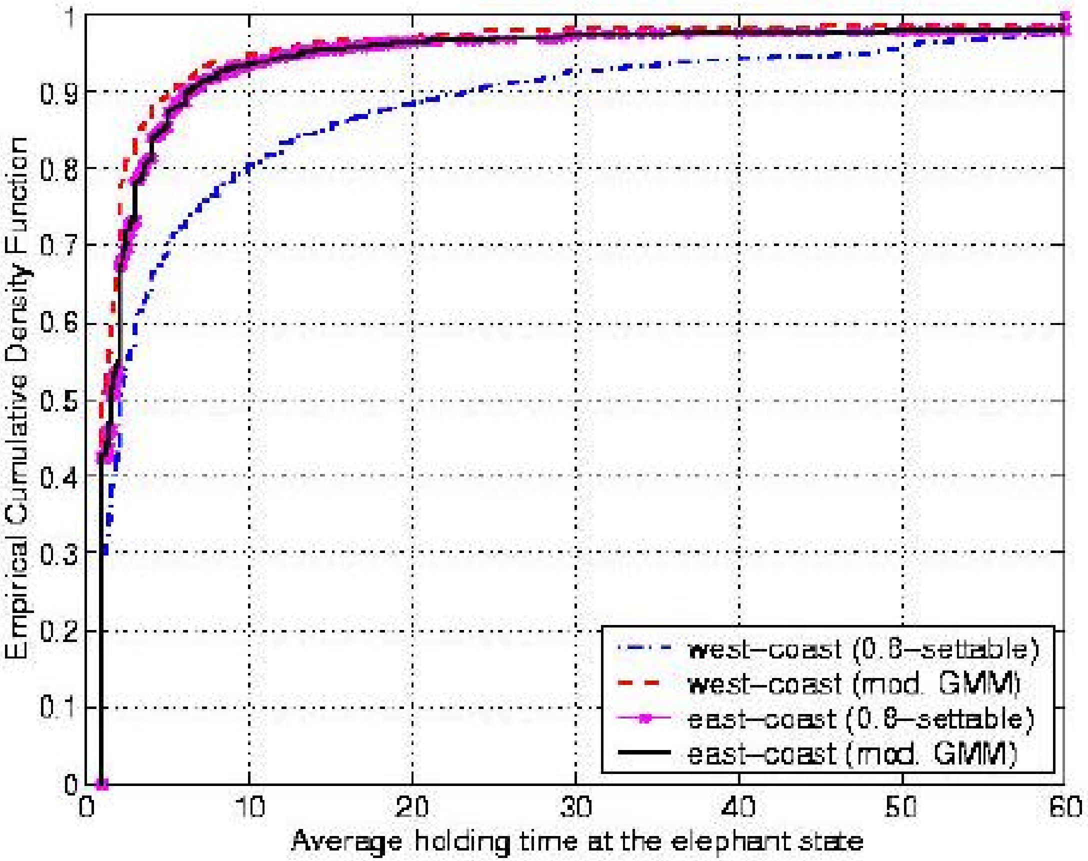
How long is an elephant?

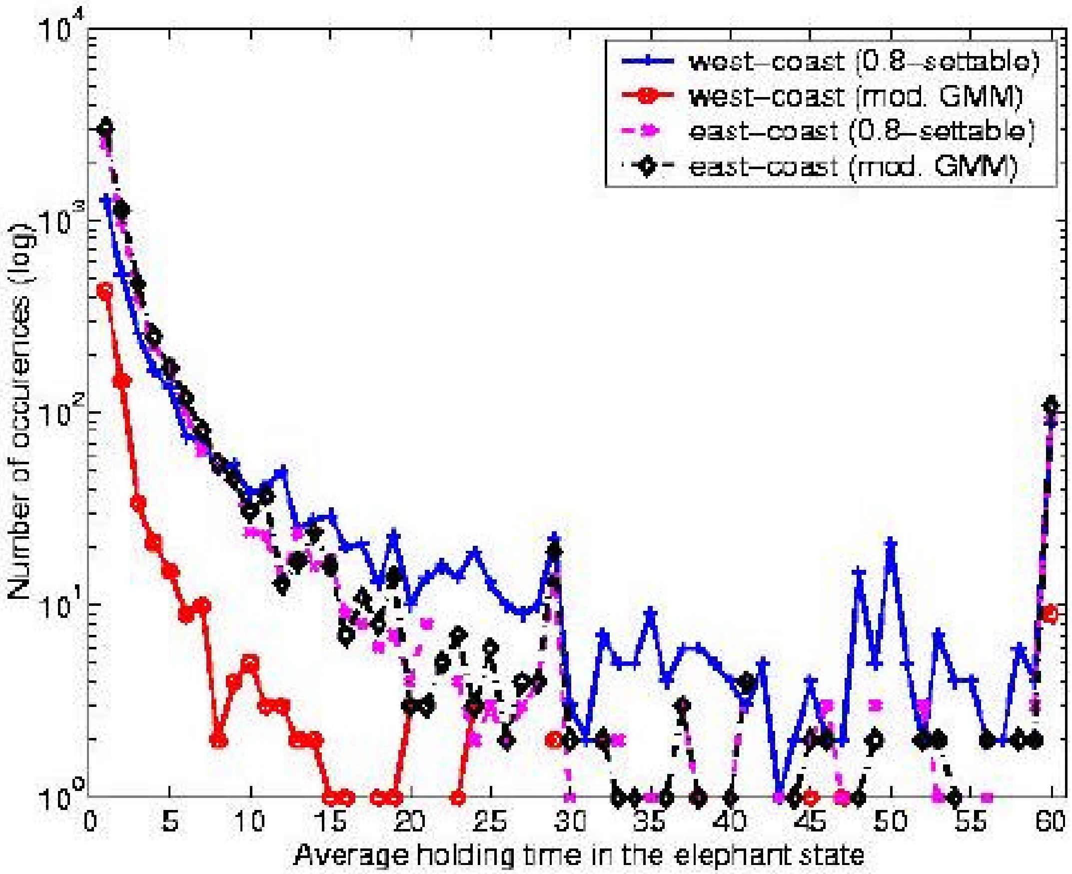
Cisco.com



How many elephants did you say were that long?

Cisco.com




west–coast (0.8–settable)
west–coast (mod. GMM)
east–coast (0.8–settable)
east–coast (mod. GMM)
Number of occurences (log)
Average holding time in the elephant state
10^0
10^1
10^2
10^3
10^4
0
5
10
15
20
25
30
35
40
45
50
55
60

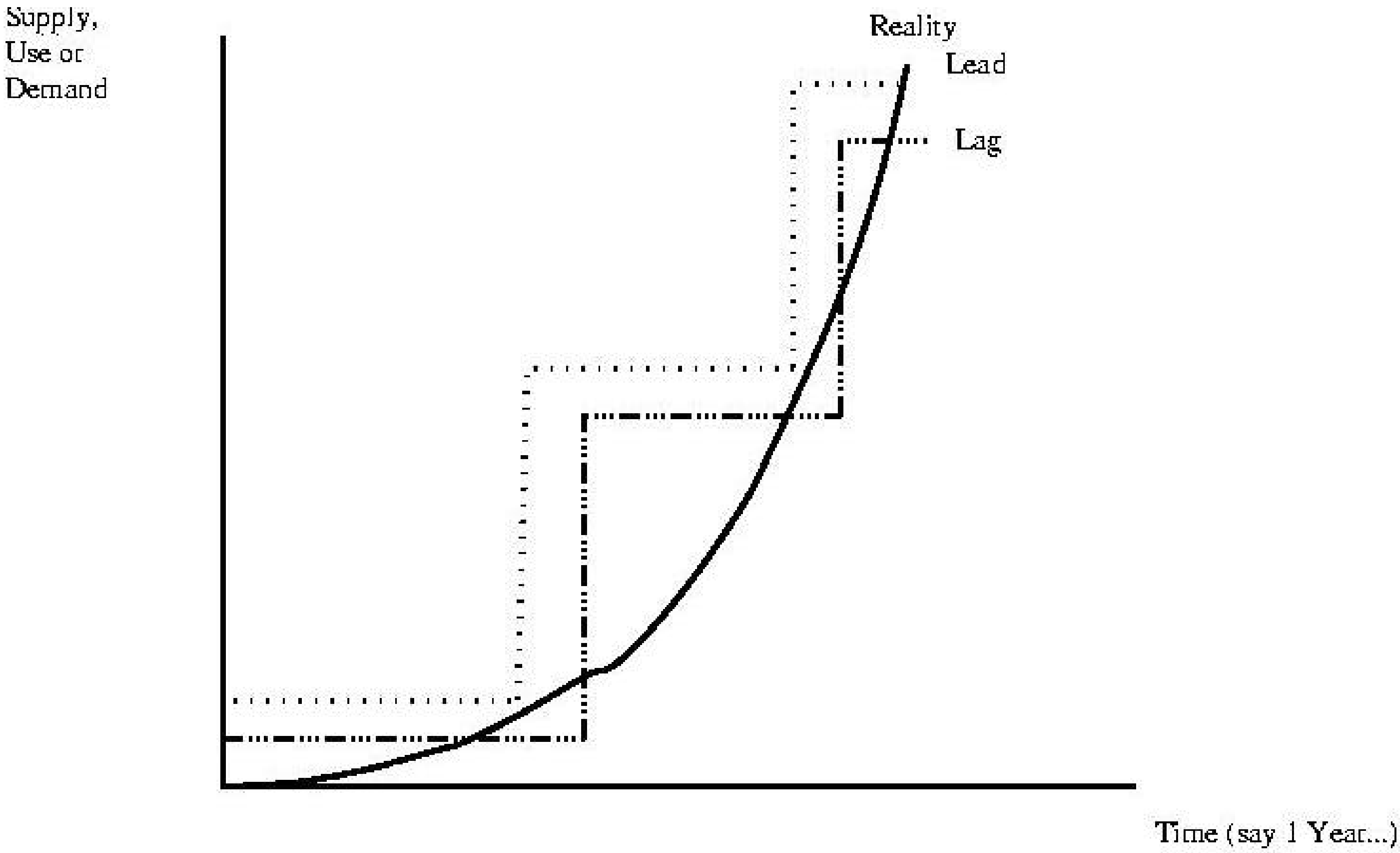
So what might we do with this?

- **We want separation of traffic types so that we can provide statistical *protection***
- ***E.g. latency/jitter for interaction***
- ***Minimum throughput for bulk transfer***
- ***E.g. 50% elephant, 1 second RTT, can compute expected mean latency for mice.***

Provisioning for Mice+Elephants

- Given required elephant capacity E and # Elephants e, bottleneck capacity =eE
- Mean loss=1/(rtt*e)^2
- Given required mice capacity is 1/rtt and # Mice is m, to get target delay, need additional capacity of m/rtt + epsilon
- Epsilon is a lot if you want low delay, since it is how you keep p low, so latency is low…can do equations…

Supply and Demand, Steps & Curves



F: Multi-path Routing: What if we seggregate Mice and Elephants?

Cisco.com

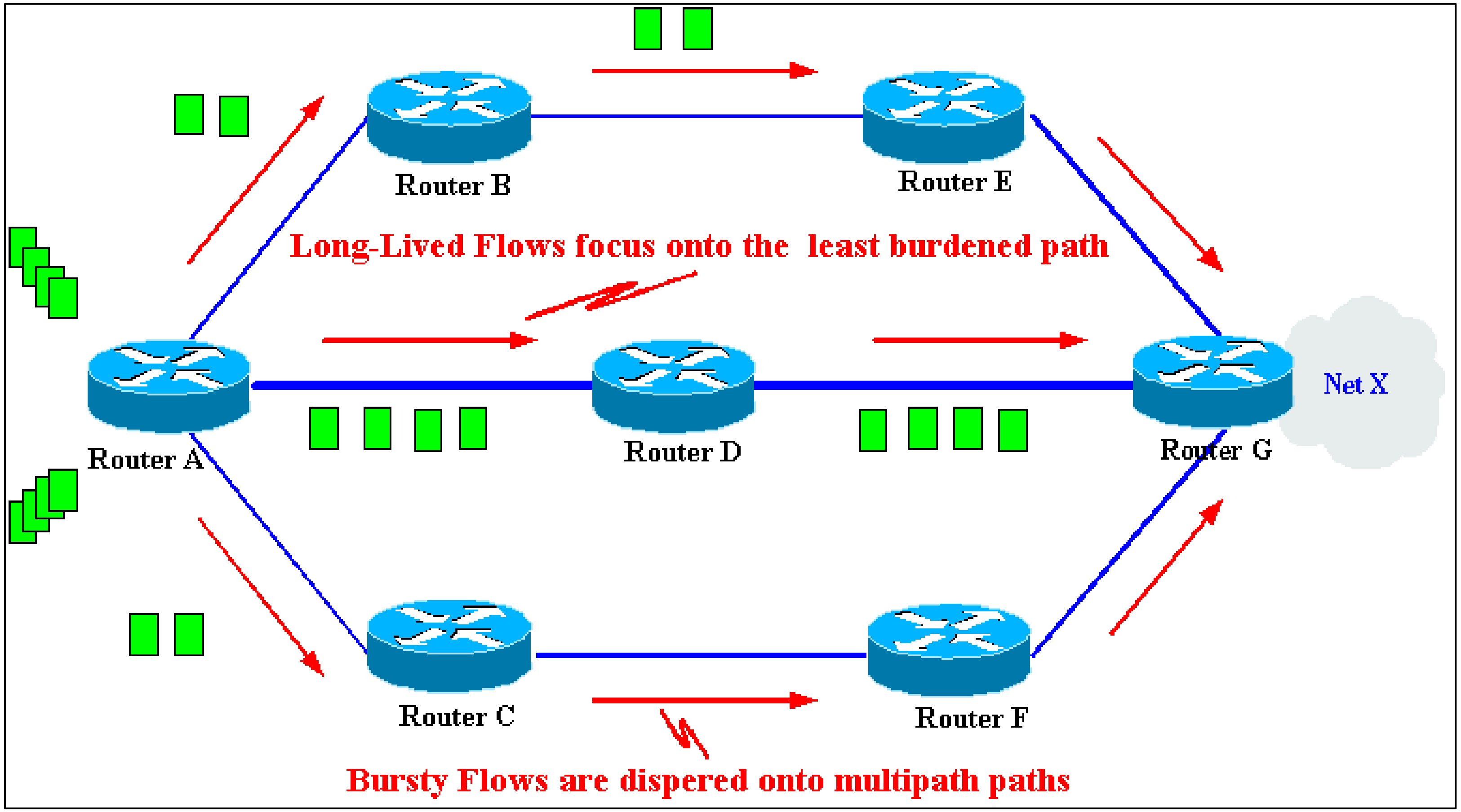
- We could have clever queues:

  e.g. premium service IP, with differentiated services EF for interactive

- But if there are enough of them, why not route them separately:

  Proactive Multi-path

Could Differentially Route Mice and Elephants



Basic ideas

- **Key decisions**
  - Spreading traffic flows according to path quality
  - Migrating long-lived flows only if alternate path can take it
- **Advantages**
  - More resources to absorb bursts
  - Flow distribution is optimized
  - Fewer link state update messages

Key Components of Proactive Multipath

Cisco.com

- Compute and establish K paths
- Path evaluation and traffic dispersion
- Detecting long-lived flows and rerouting

Compute and Establish K paths

- Link Metrics

  Bandwidth related metric

- Algorithm

  Extended Dijkstra or labling algorithm

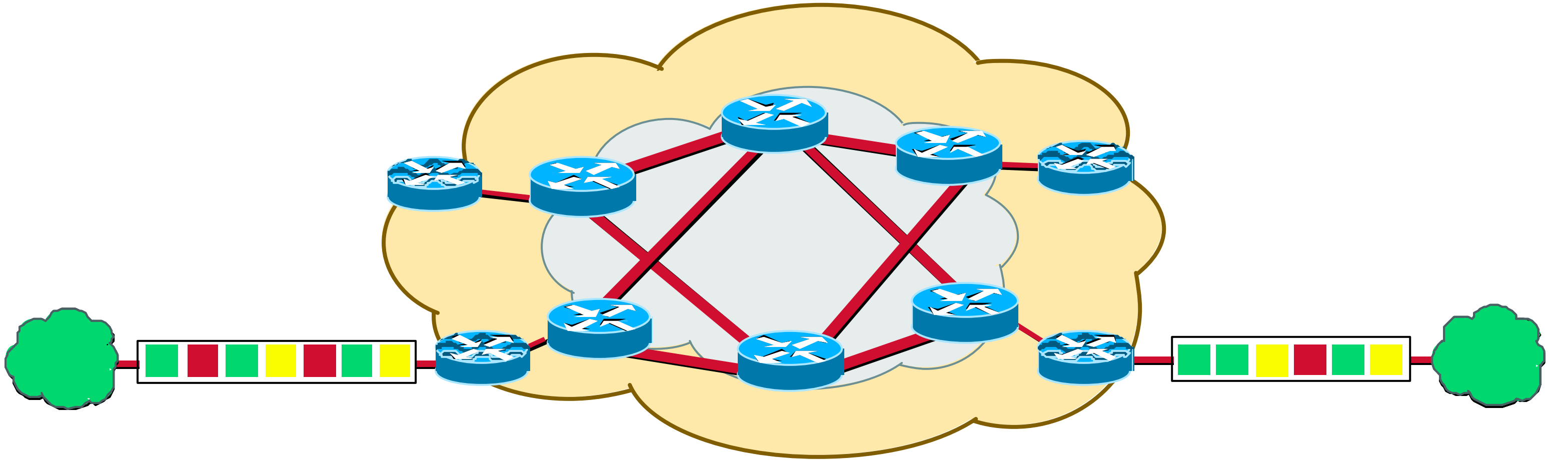
- Path Establishment

  CR-LDP/RSVP-TE in MPLS networks

  Signaling as that of MPOA in ATM networks

- Periodic Re-computation

Path evaluation and traffic dispersion



Evaluation Policy	Quality Evaluation	Flow Assignment Policy
"random"	Each path is considered the same	Random
"wrr-bw"	Bottleneck_bandwidth	W.R.R.
"wrr-bp"	Bottleneck_bandwidth/path_cost	W.R.R.
"wrr-bh"	Bottleneck_bandwidth/hop_count	W.R.R.

Detecting long-lived flows and rerouting

Cisco.com

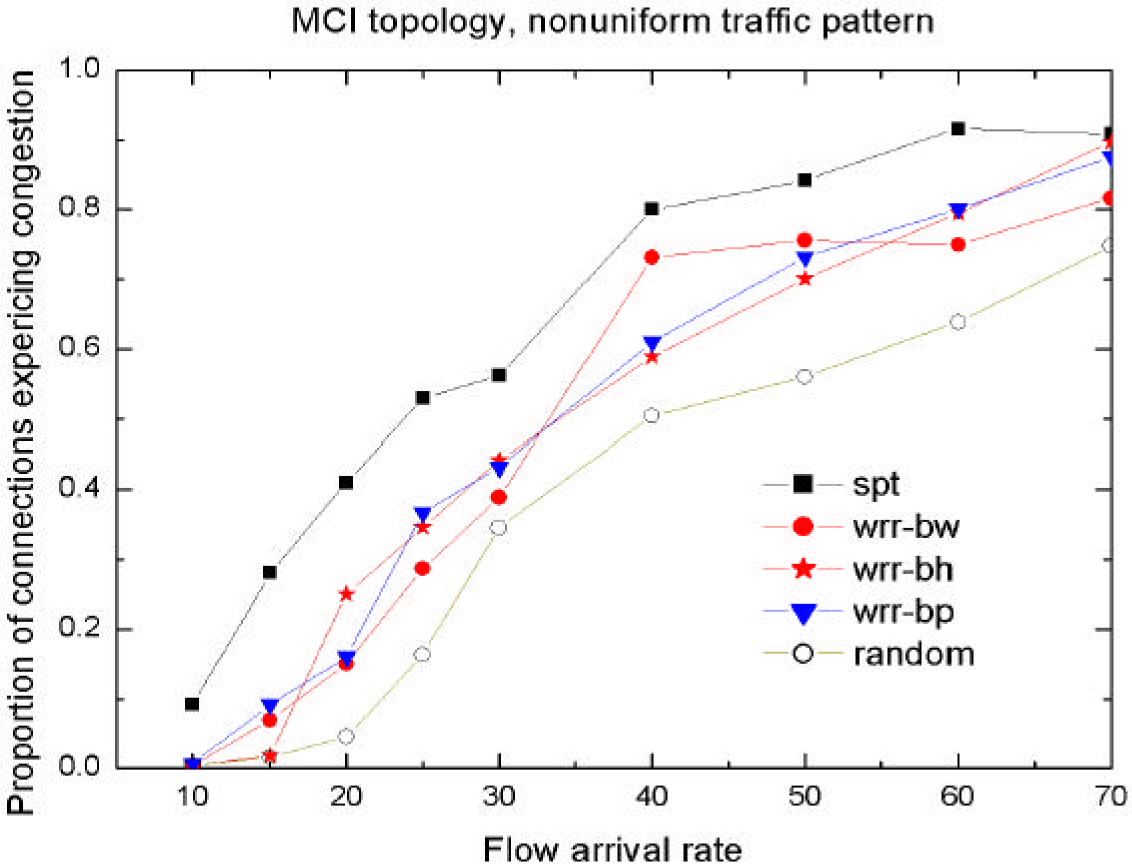
- Flow is identified by information in packet header
- Edge router monitor flow transmission
- Two or more feature flow identification is preferred

Evaluation by Simulation

Ingress	Egress	Hop Distance	Arrival rate
1	12	4	10.6
0	14	4	10.6
2	17	4	10.6
9	11	4	10.6
6	17	3	10.6
4	13	3	10.6
10	8	2	10.6

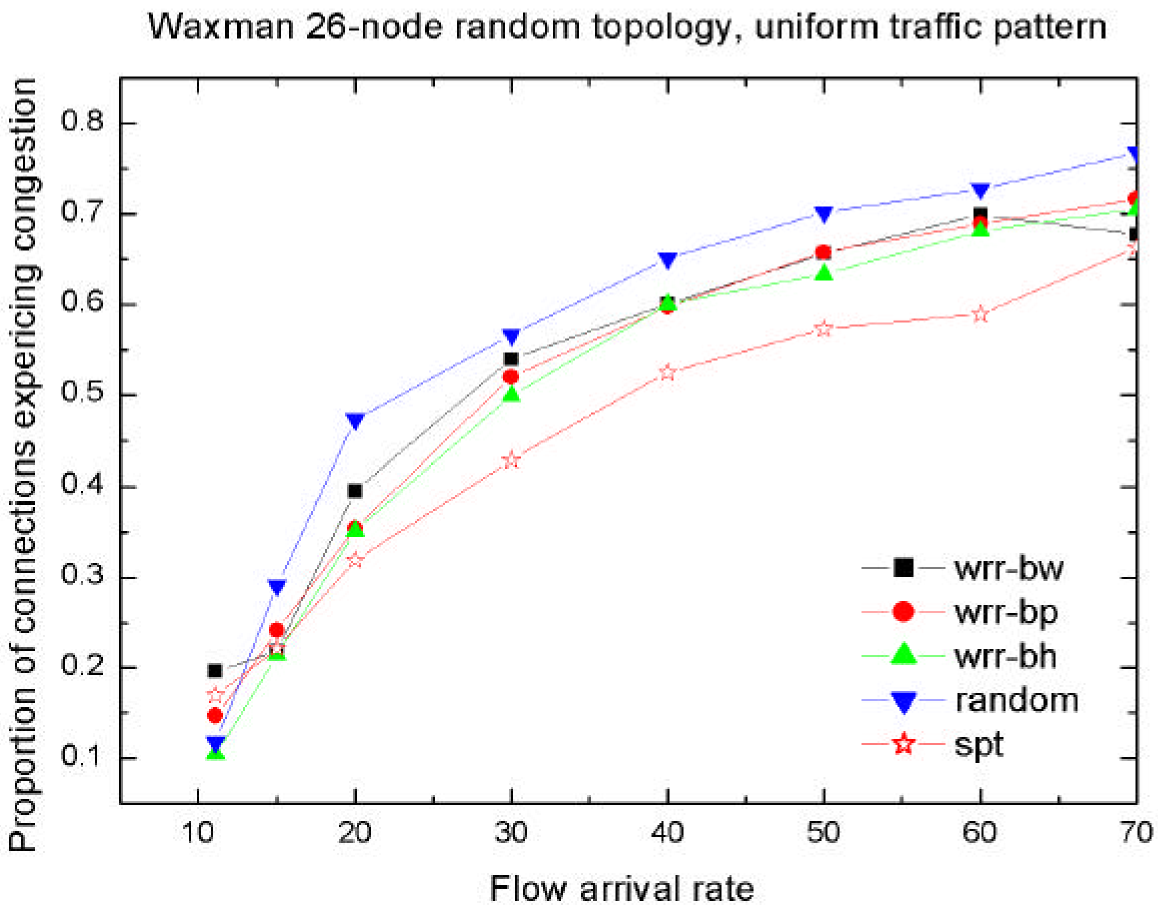
Network Throughput (non-uniformed traffic pattern)

Cisco.com



Network Throughput (uniformed traffic pattern)

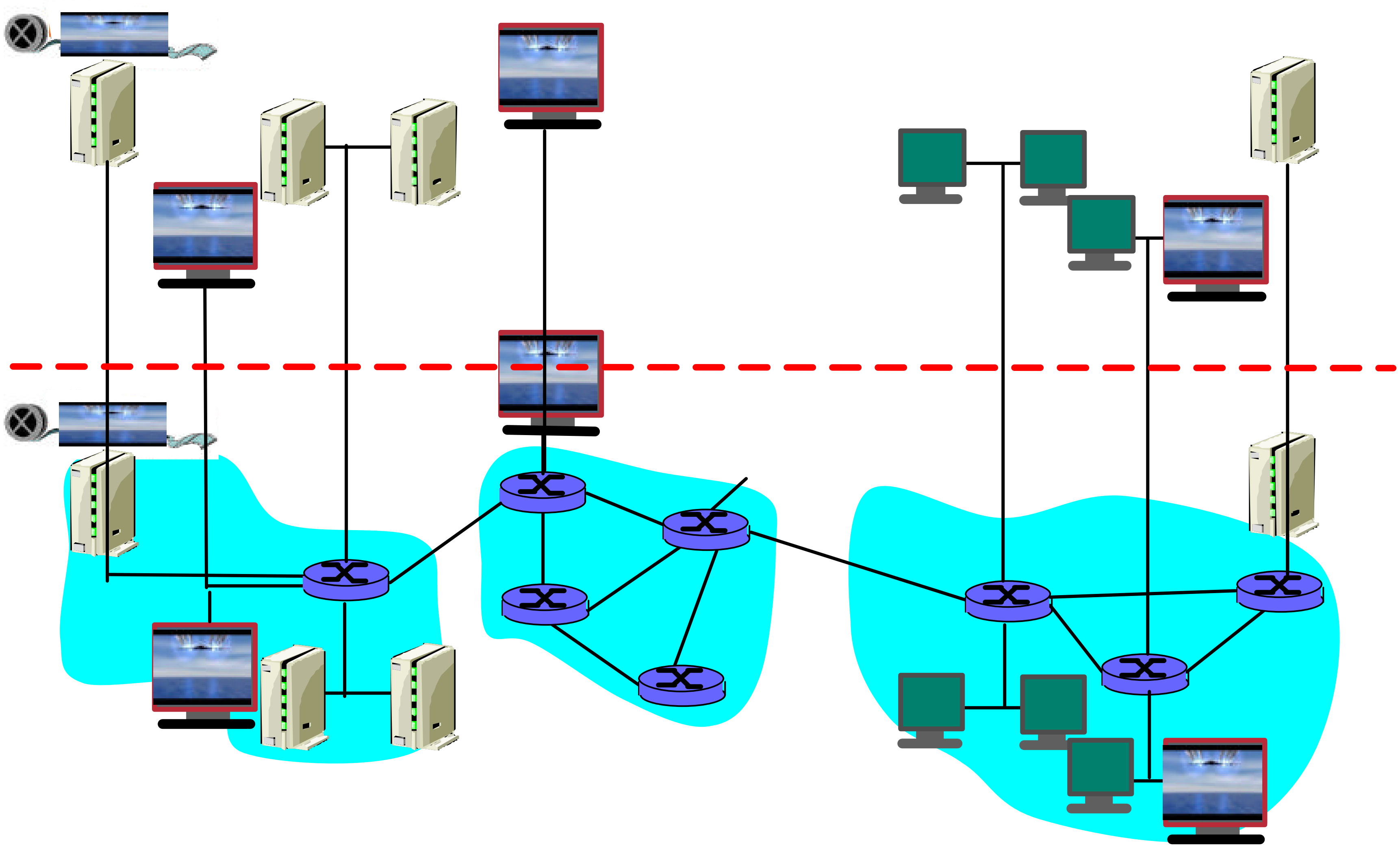
Cisco.com



G: The Future: p2p, grid, unknown…

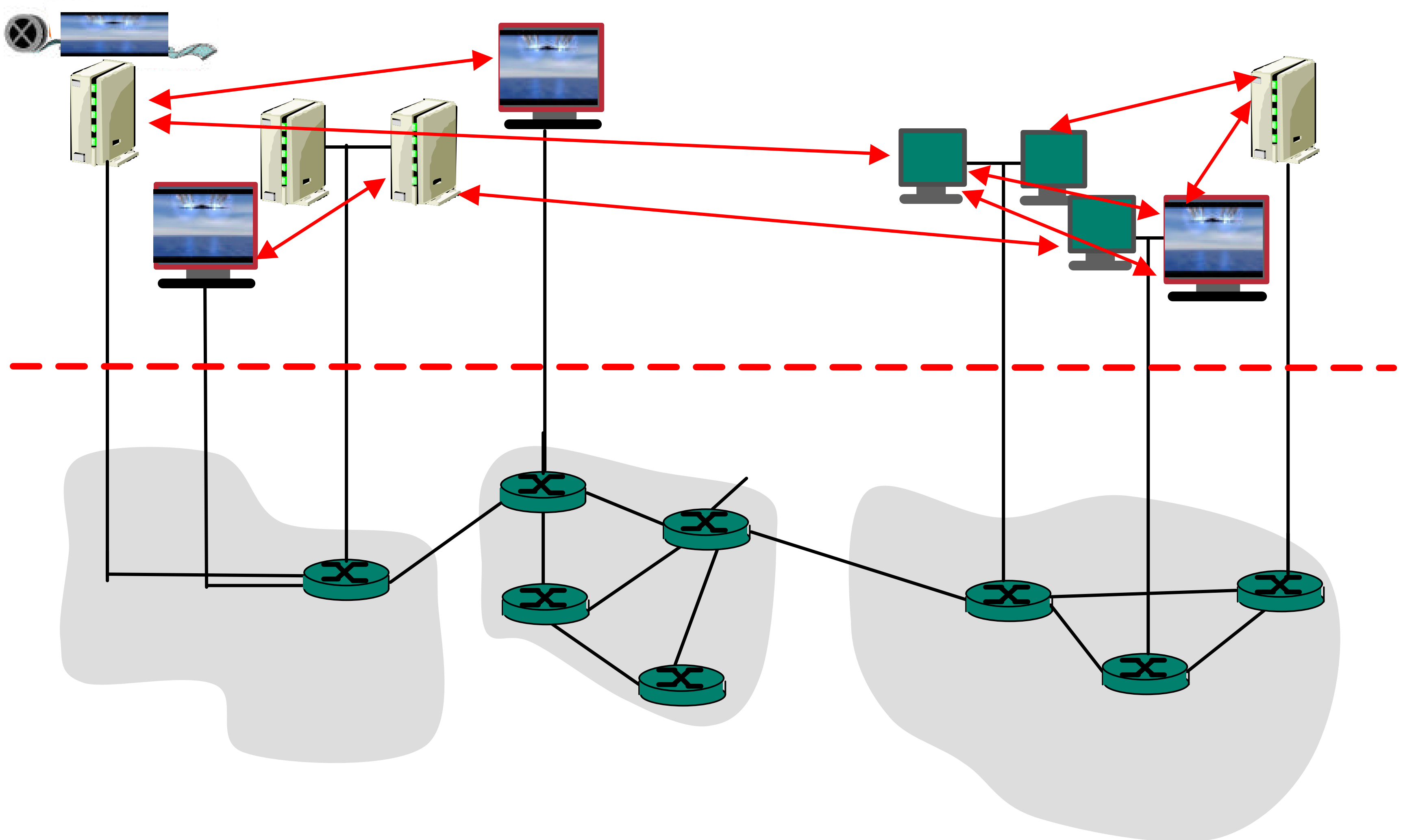
- Traffic could change in space and time
- It already is….
- P2p and Grid Computing Exhibit New patterns…

Peer-peer networking



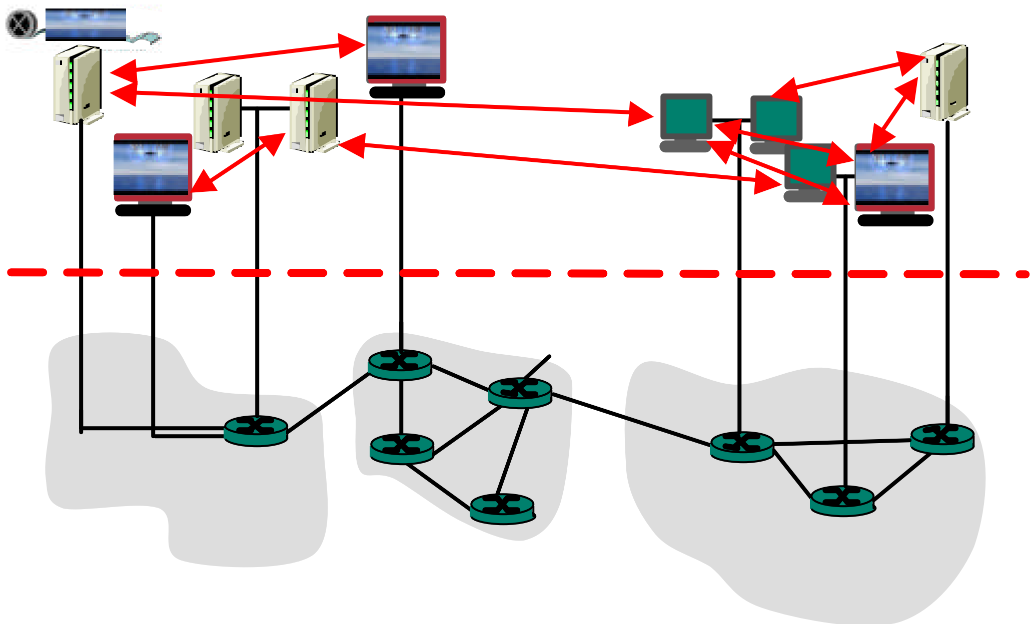
Peer-peer networking

Focus at the application level



Peer-peer networking

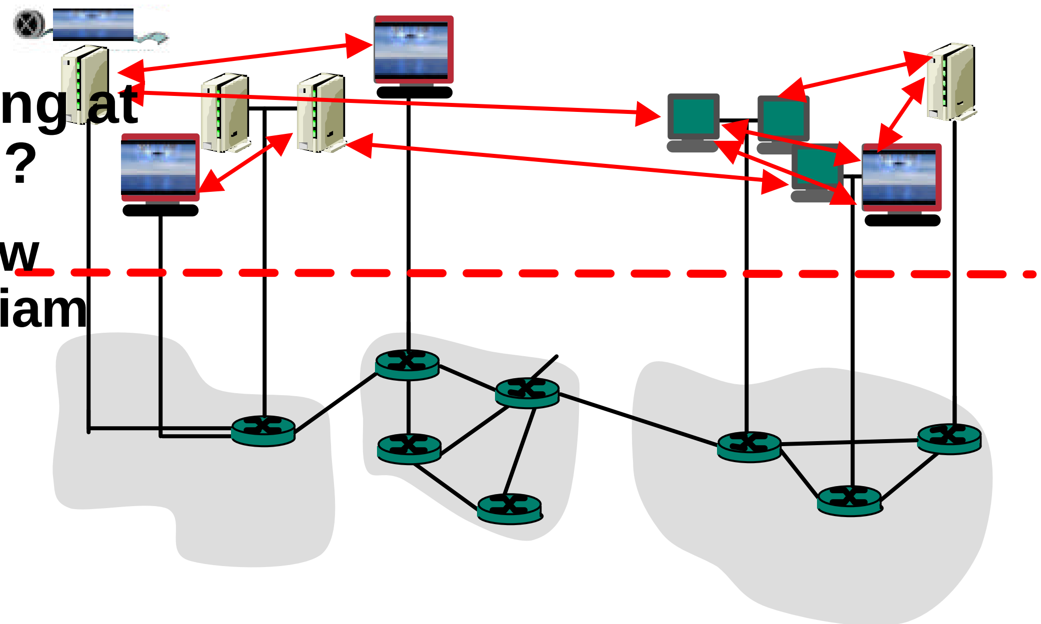
- Peer-peer applications
- Napster, Gnutella, Freenet: file sharing
- ad hoc networks
- multicast overlays (e.g., video distribution)



Peer-peer networking

- Q: What are the new technical challenges?
- Q: What new services/applications enabled?
- Q: Is it just "networking at the application-level"?

  "There is nothing new under the Sun" (William Shakespeare)



Napster

:

5/99: Shawn Fanning (freshman, Northeasten U.) founds Napster Online music service

12/99: first lawsuit

3/00: 25% UWisc traffic Napster

2000: est. 60M users

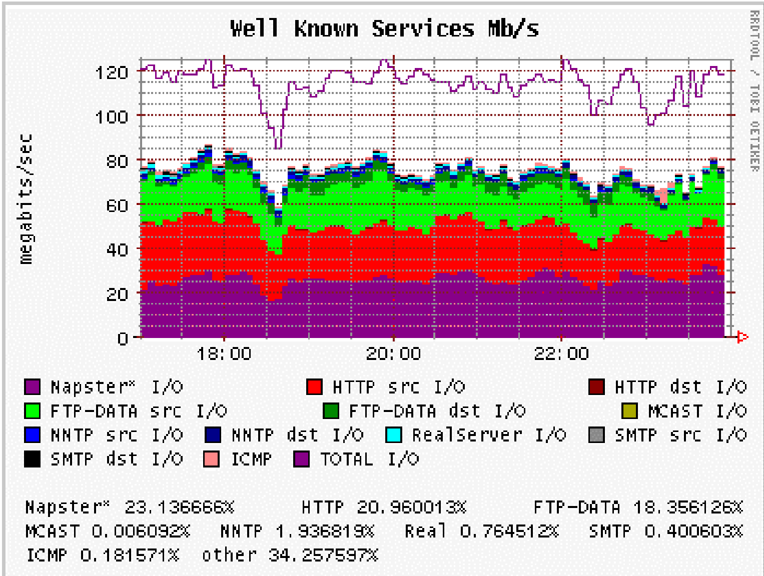
2/01: US Circuit Court of

Appeals: Napster knew users

violating copyright laws

7/01: # simultaneous online users:

Napster 160K, Gnutella: 40K, Morpheus: 300K



H. What to do to stay futureproof?

Cisco.com

- Agile measurement program
- Modeling Program
- Provisioning Plan

Aims and Objectives Revisited

- Subscriber wish:
- Squeeze as much capacity out of a provisioned service as possible for a given price, subject to delay constraints.
- Sites may be underspecified

- Provider wish:
- Squeeze as much income as possible out of a given subscriber set with a given network provisioning, subject to meeting SLAs
- Users may surprise!

User Experience!

- Throughput for Bulk Tasks

  And

- Delay for interactive Tasks
- Alternatives harder to deploy (e.g. smallest TCP first scheduling or priority queueing – all non end2end, and therefore need AAA).

Ack for material

Cisco.com

- Fred Baker (outline)
- Nick McKeown (tcp)
- Dina Papagiannaki (mice&elephants)
- Jing Shen (mpr)
- Jim Kurose (p2p)