bandwidth estimation: measurement methodologies and applications

kc claffy *

and

Constantinos Dovrolis †

*CAIDA

†Georgia Tech

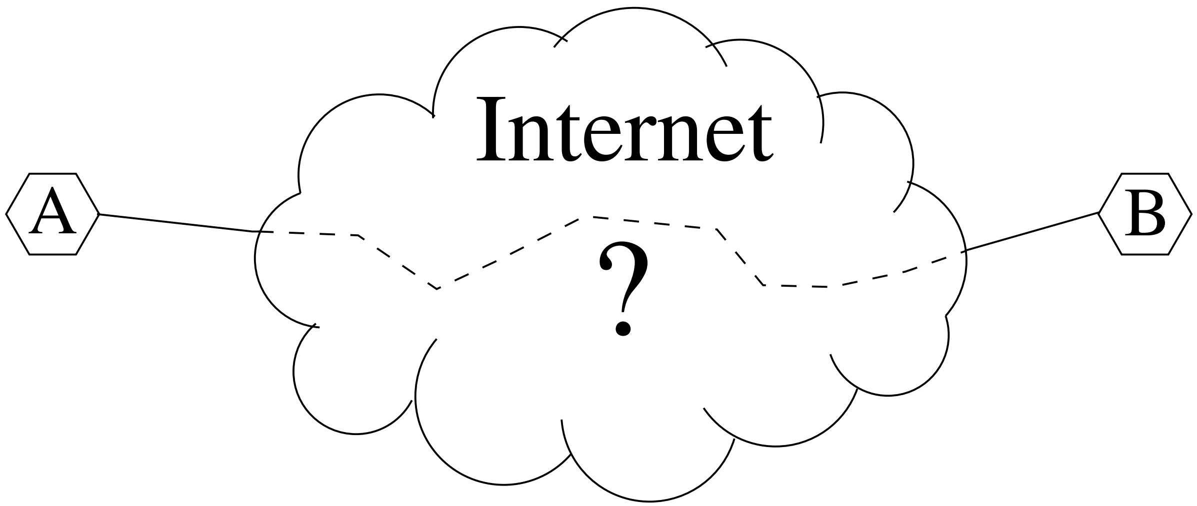
Financials

- GaTech: tool development
  - $100K/year for 3 years
- CAIDA: testing, administrative, workshops
  - $290K/year for 3 years

background

Looking inside a cloud..

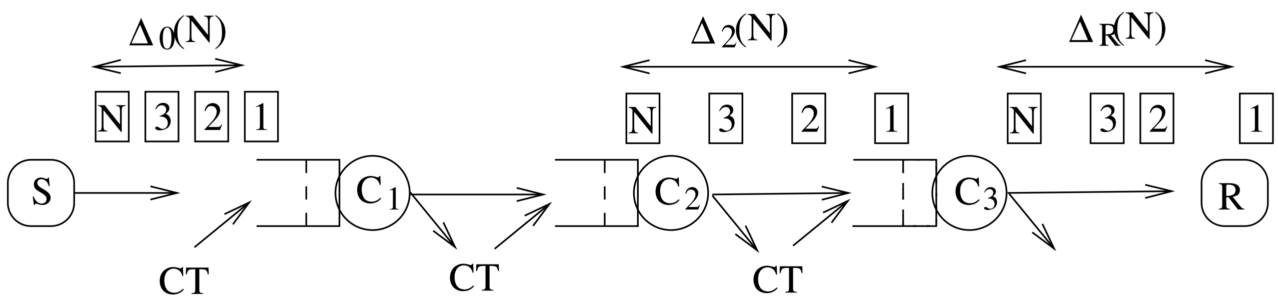
- From the user perspective, the Internet is a big **black** cloud
- Routers do not send explicit feedback to end-systems
- Fundamental to network simplicity and scalability



- End-systems must *infer* network characteristics via end-to-end measurements
- Example: TCP Round-Trip Time (RTT) estimation

Path inference

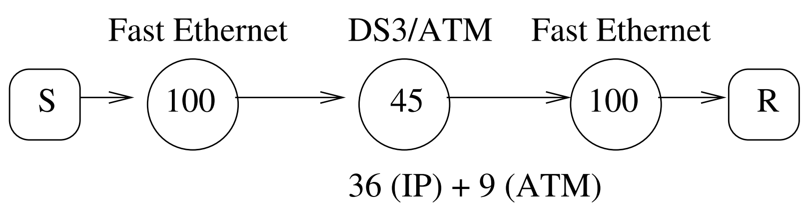
- Network path: sequence of links & routers from sender S to receiver R
- Path characteristics: round-trip time, delay jitter, loss rate, bandwidth, etc
- Inferrence techniques use special probing packets or application packets to measure path characteristics



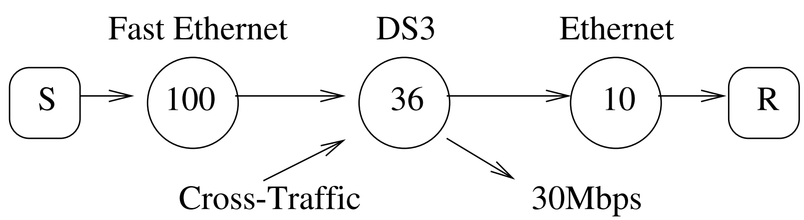
- Objectives: accuracy, non-intrusiveness, timeliness

Bandwidth estimation

- Two throughput-related performance metrics (for fixed paths)
- **Capacity:** maximum possible IP-layer throughput in path



- Independent of cross traffic load
- **Available bandwidth:** idle (non-utilized) part of path's capacity



- Dynamically varying metric

Main project objectives

- Develop original estimation methodologies for bandwidth estimation, based on solid research results
- Write & distribute tools for capacity and available bandwidth estimation
- Meet following objectives:
  1. Accurate
  2. Fast (especially for available bandwidth)
  3. Non-intrusive (do not saturate path)
- Test and evaluate existing and emerging tools under realistic conditions

Applications of bandwidth estimation

- Congestion control and TCP: automatic socket buffer sizing
- Overlay networks: configure overlay routes
- Content distribution networks: select best server
- Streaming applications: adjust encoding rate
- SLA and QoS verification: monitor path load
- End-to-end admission control: check for sufficient bandwidth
- Peer-to-peer networks: construct application-layer topology
- Interdomain traffic engineering: select egress ISP
- And many more..

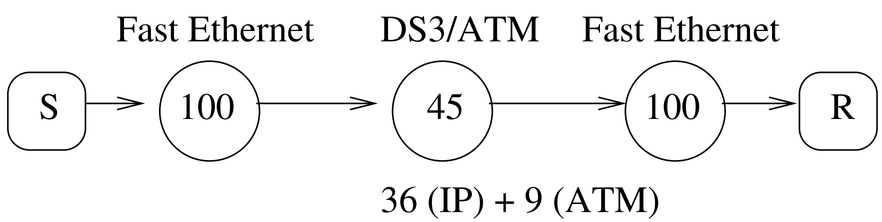
Talk overview

- Capacity estimation and pathrate
- Available bandwidth and pathload
- Per-hop capacity estimation and L2 devices
- Bandwidth estimation tool evaluation
- ANEMOS: Autonomous NEtwork MOnitoring System
- SOBAS: SOcket Buffer Auto-Sizing
- Looking forward: tech transfer to grid community, IETF/IRTF

End-to-end capacity estimation
and
Pathrate
(published at Infocom 2001 and under submission at Transactions in Networking)

Capacity

- **Capacity:** maximum possible end-to-end throughput



- End-to-end capacity C is limited by *narrow link* n:

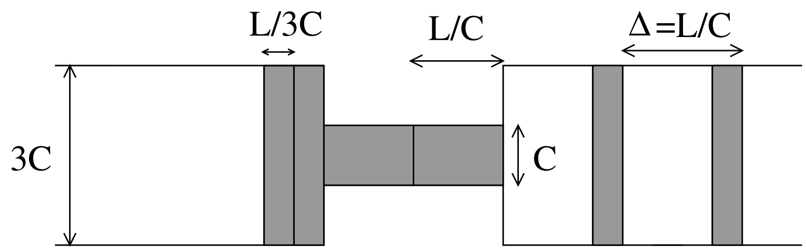
$$C = \min_{i=0 \ldots H} \{C_i\} = C_n$$

- **Pathrate:** measurement tool based on packet pairs/trains

  See www.pathrate.org

Packet pair: basic idea

- Transmission time of L-byte packet at link with capacity C: $\tau = \frac{L}{C}$
- Send two packets 'back-to-back' from source to sink
- Measure *dispersion* (distance) Δ of packet pair at receiver

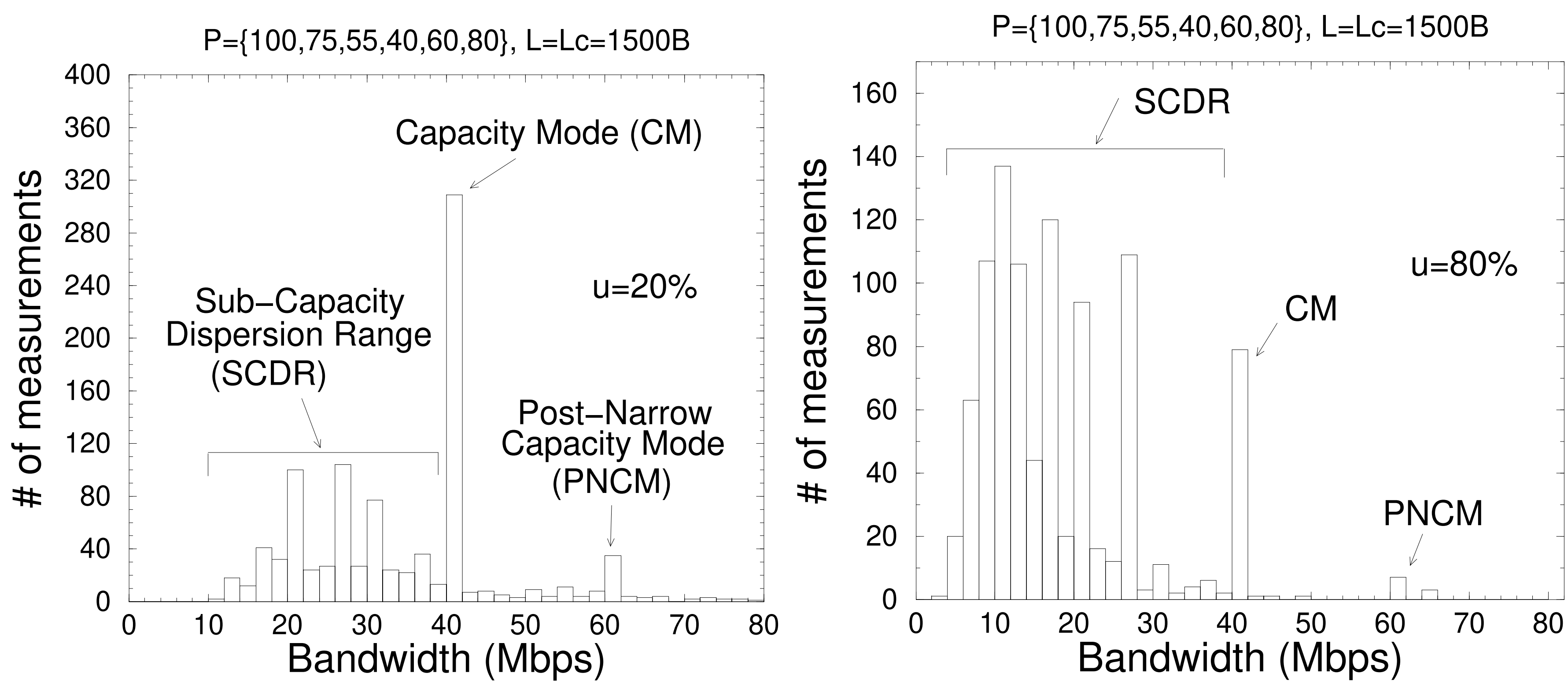


$$\Delta = \max_{i=0\ldots H} \tau_i = \frac{L}{\min_{i=0\ldots H}\{C_i\}} = \frac{L}{C}$$

- Idea: estimate capacity C from L/Δ measurement

Does the packet pair technique really work?

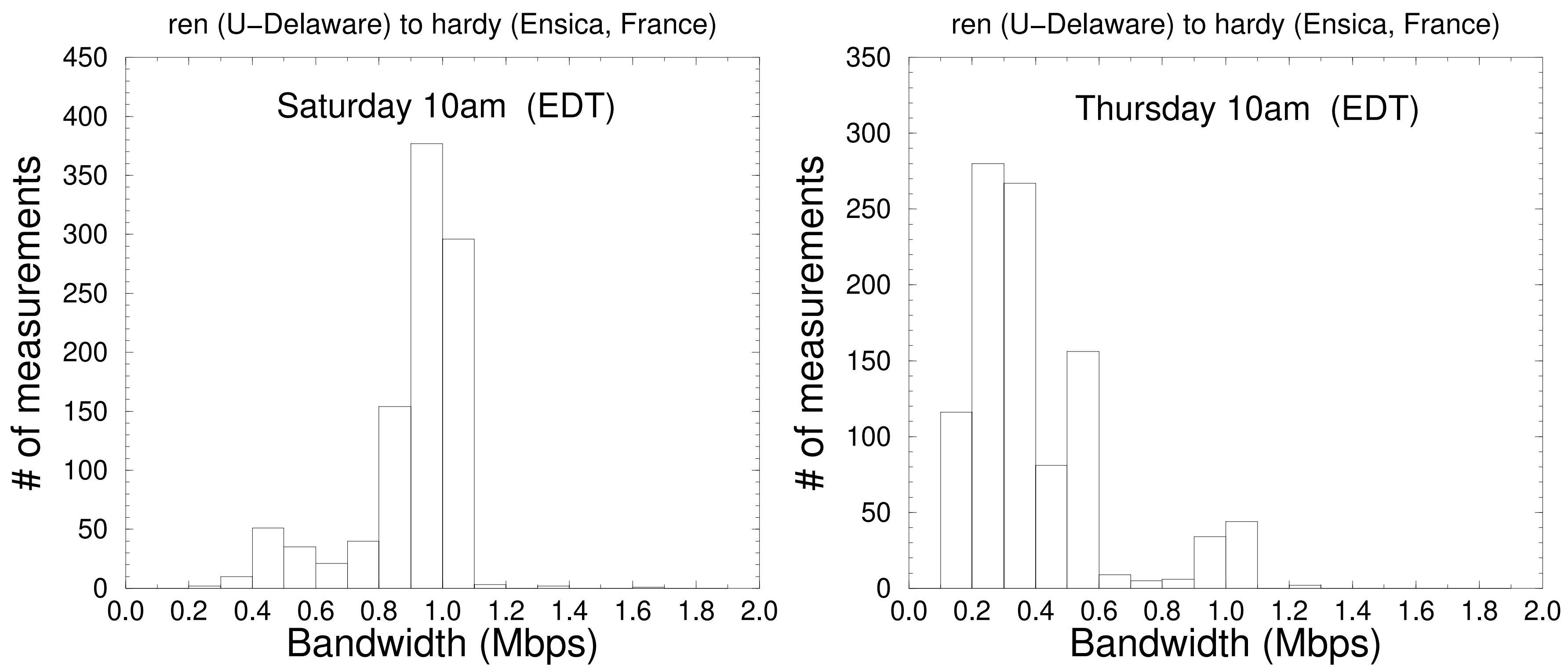
- Plot histogram of capacity measurements from 1000 packet pair experiments



- Cross traffic creates local modes below and above capacity

Loaded paths are harder to measure

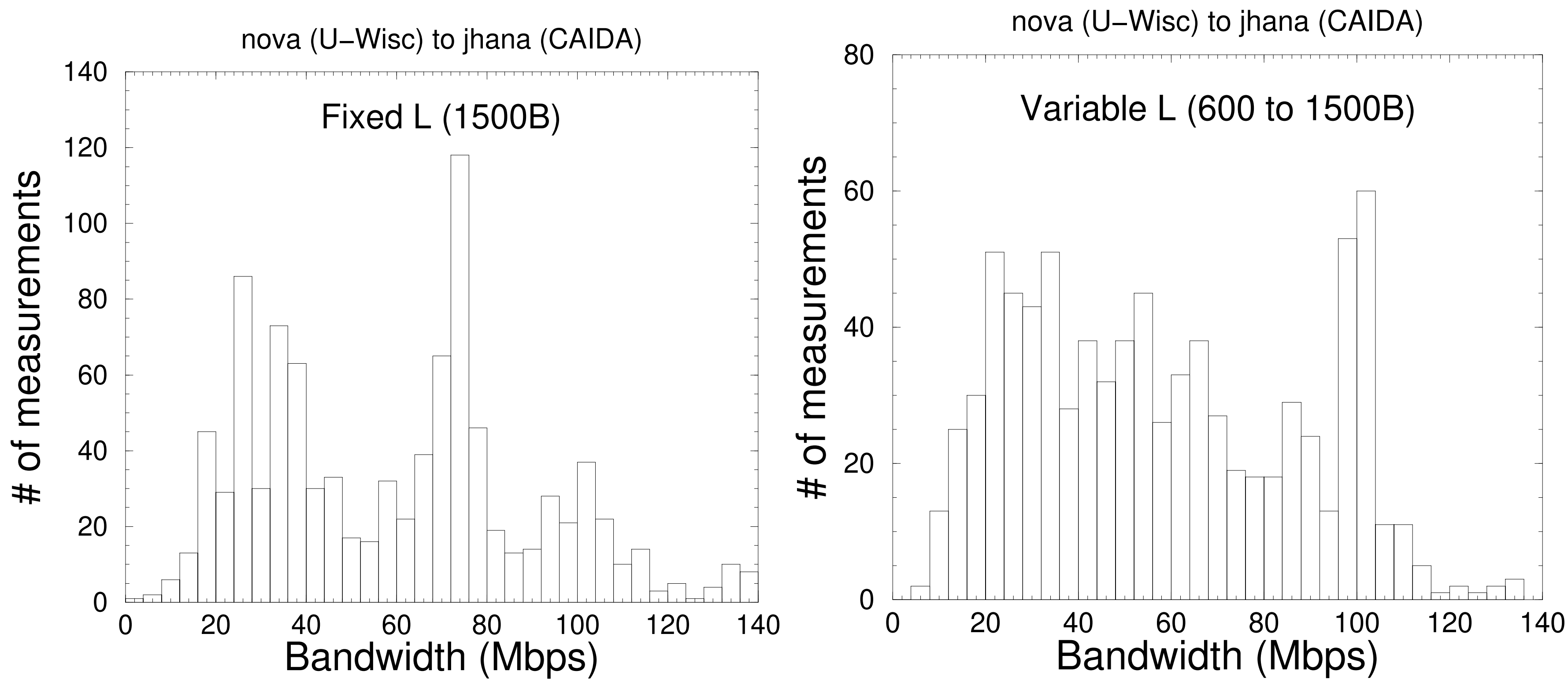
- Cross traffic interferes with packet pair more often in loaded paths



- Challenge: identify mode that corresponds to capacity from multimodal distribution

Effect of packet pair size

- Variable packet sizes make distribution ‘less multimodal’



- Reason: cross traffic packets tend to have certain sizes (e.g., 40B, 1500B)

Pathrate estimation methodology

Phase I:

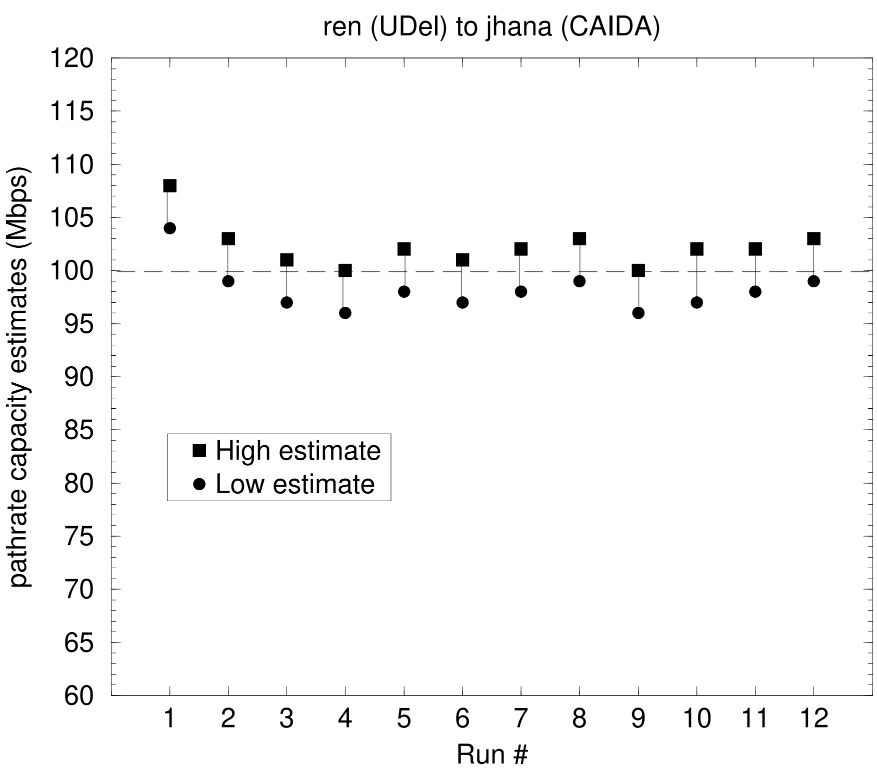
- Perform 1000 packet pair measurements
- Packet size varies between 500 and 1500 bytes
- Estimate local modes in Phase I: $\mathcal{M} = \{m_1, m_2, \ldots m_M\}$

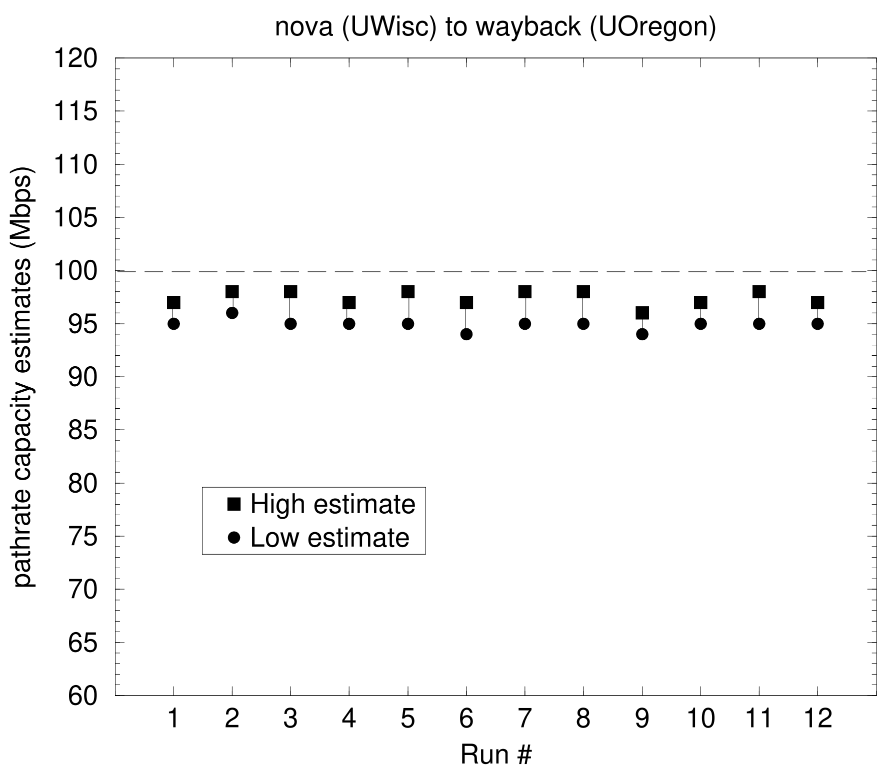
Phase II:

- Perform 500 packet train measurements to measure Average Dispersion Rate (ADR)
- Reject Phase I modes which are less than ADR
- Capacity is 'strongest' and 'narrowest' among remaining modes

Pathrate results

- Fairly accurate in paths that range from 100kbps to 1Gbps (4 orders of magnitude)





- Large errors are more common in heavily loaded paths

Available bandwidth estimation

and

Pathload

(published at SIGCOMM 2002 and Transactions in Networking)

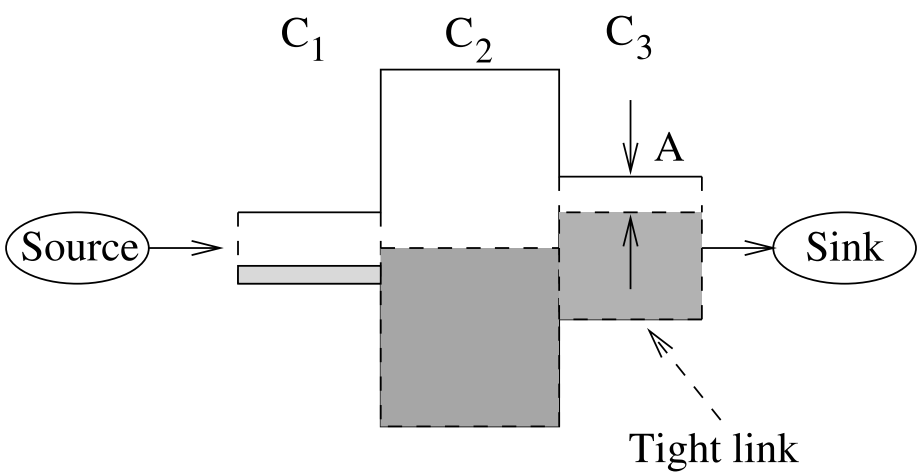
Definition of avail-bw

- u_i: Average utilization of link i in a time interval of length τ $(0 \leq u_i \leq 1)$

- Avail-bw of link i: $\quad A_i = C_i\,(1 - u_i)$

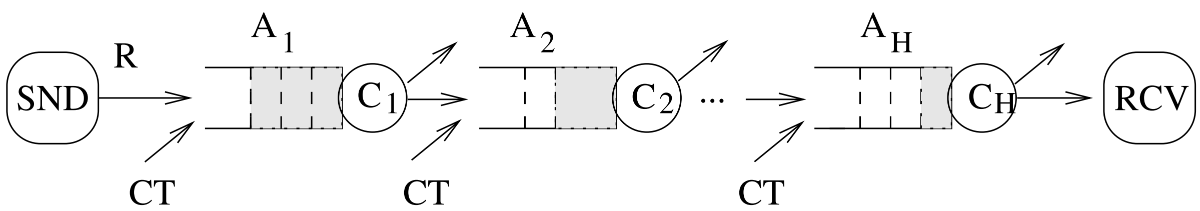
  End-to-end avail-bw: $\quad A = \min_{i=0\ldots H} A_i = \min_{i=0\ldots H} C_i\,(1 - u_i)$

- Time interval length τ: averaging timescale



- Avail-bw is limited by *tight link*

Estimation technique: Self-Loading Periodic Streams



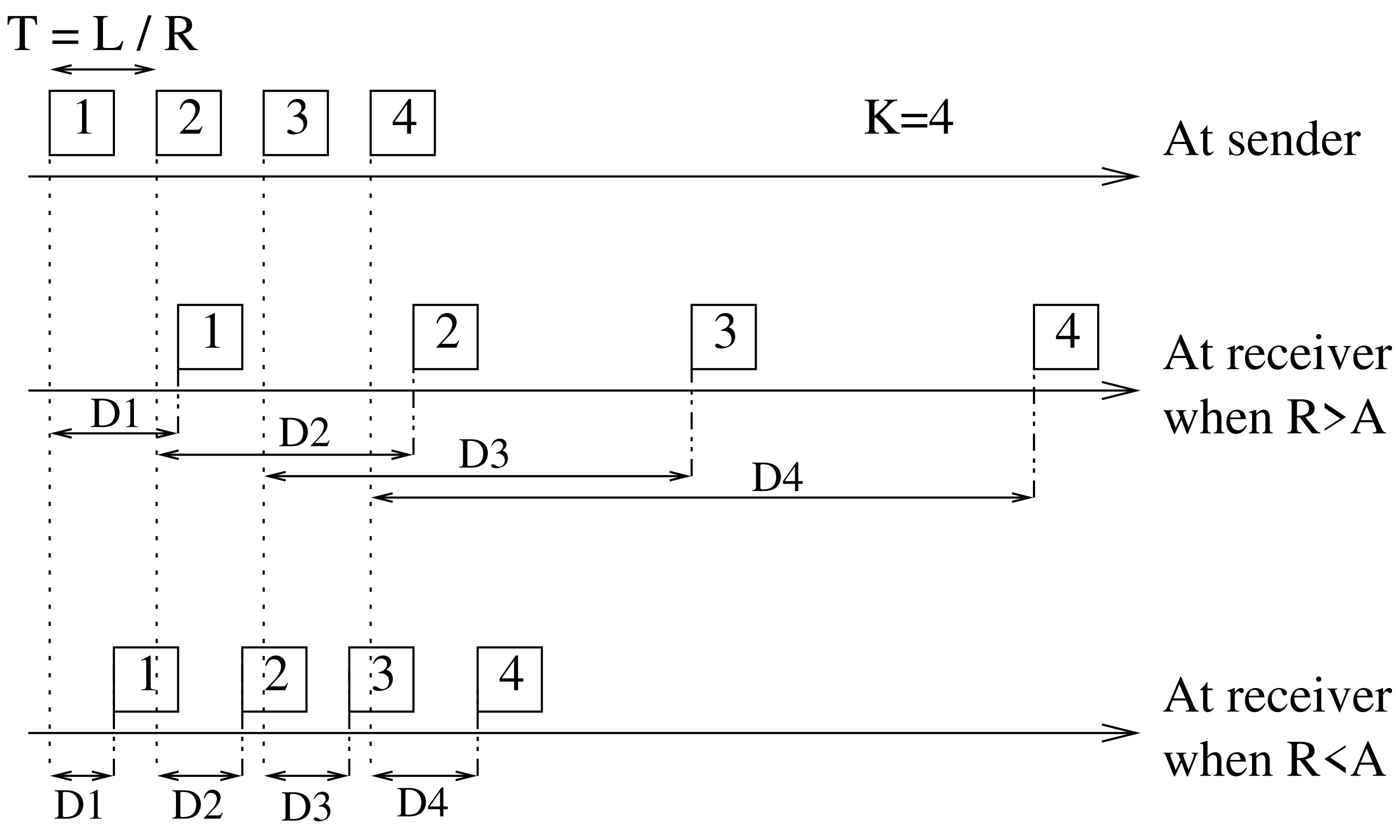
- SND sends a periodic UDP packet stream of rate R
- Stream characteristics: K packets, size L, period T, rate $R = L/T$
- Measured One-Way Delay (OWD): $D^k = T^{RCV}_{arrive} - T^{SND}_{send}$
- OWD variation: $\Delta D^k = D^{k+1} - D^k$ (independent of clock offset)
- With a stationary & fluid model for the cross traffic, and FIFO queues:

$$\textit{If } R > A = \min A_i, \textit{ then } \Delta D^k > 0 \textit{ for } k = 1, \ldots, K-1$$

$$\textit{Else, } \Delta D^k = 0 \textit{ for } k = 1, \ldots, K-1$$

Illustration of basic idea

- Periodic stream: K packets, period T, packet size L, rate: $R = L/T$



Iterative rate adjustment to measure A

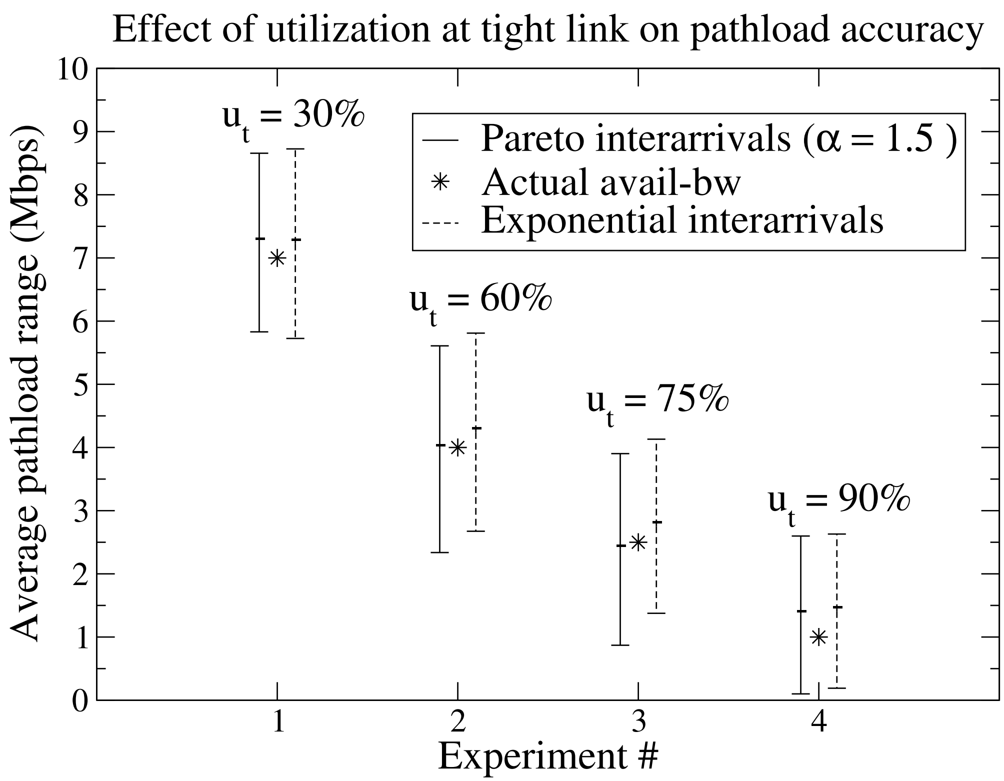
1. Source: Send n-th periodic stream with rate $R(n)$
2. Receiver: Measure delays D^k for $k = 1 \ldots K$
3. Receiver: Check for increasing delay trend, notify source
4. Source:

$$\begin{aligned} \text{If increasing delays } \ (R(n) > A), R^{max} &= R(n); \\ \text{If non-increasing delays } \ (R(n) < A), R^{min} &= R(n); \\ R(n+1) &= (R^{max} + R^{min})/2; \end{aligned} \tag{1}$$

- Exit when $R^{max} - R^{min} \leq \omega$ (ω: estimate resolution)

Pathload results

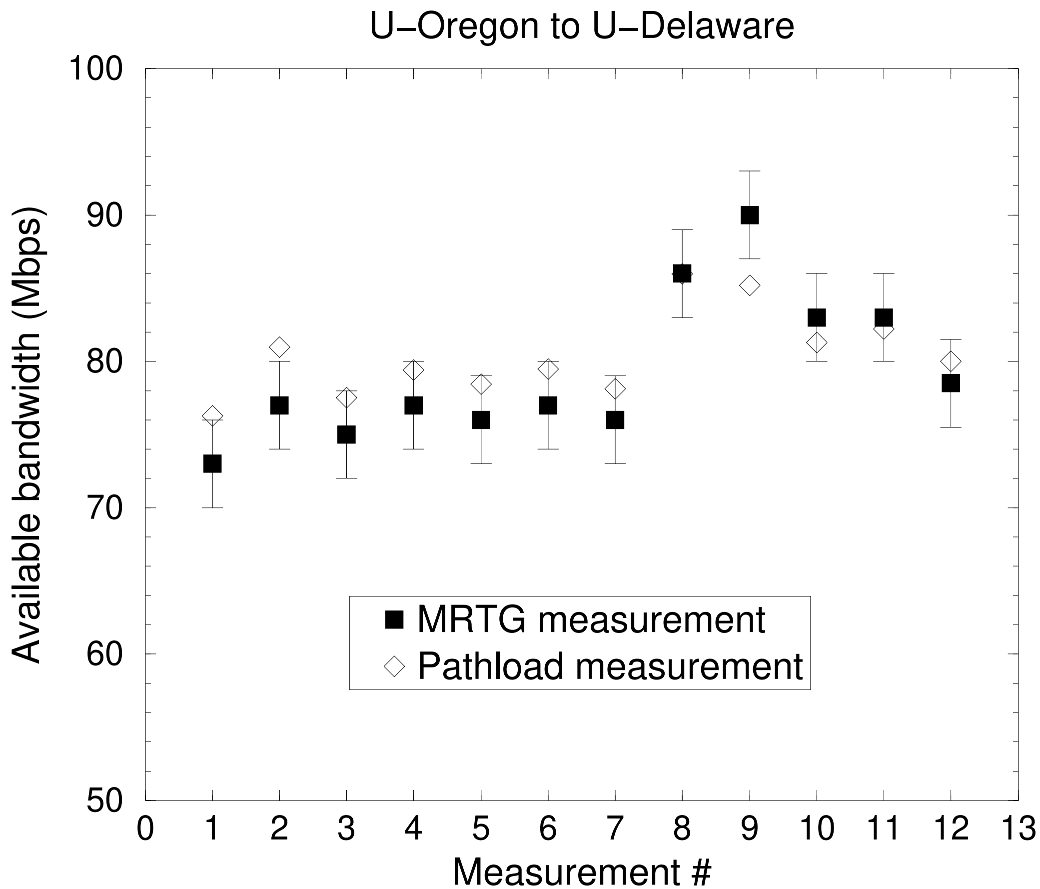
- Pathload uses the SLOPS estimation methodology
- Pathload range estimates variation range of available bandwidth



- Center of Pathload range: good estimate of average avail-bw

Experimental results

- Tight link: U-Oregon GigaPoP link (C=155Mbps), ω=3Mbps
- Compare Pathload estimate (average of consecutive runs for 5 mins) with 5-min average avail-bw from MRTG readings



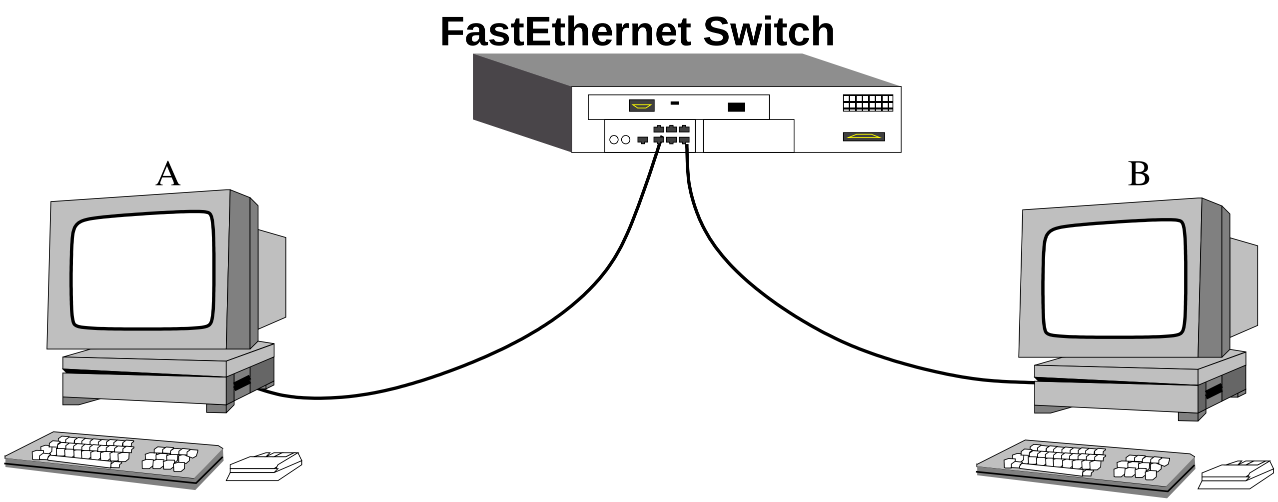
Per-hop capacity estimation

and

layer-2 effects

(published at INFOCOM 2003)

A single-hop path



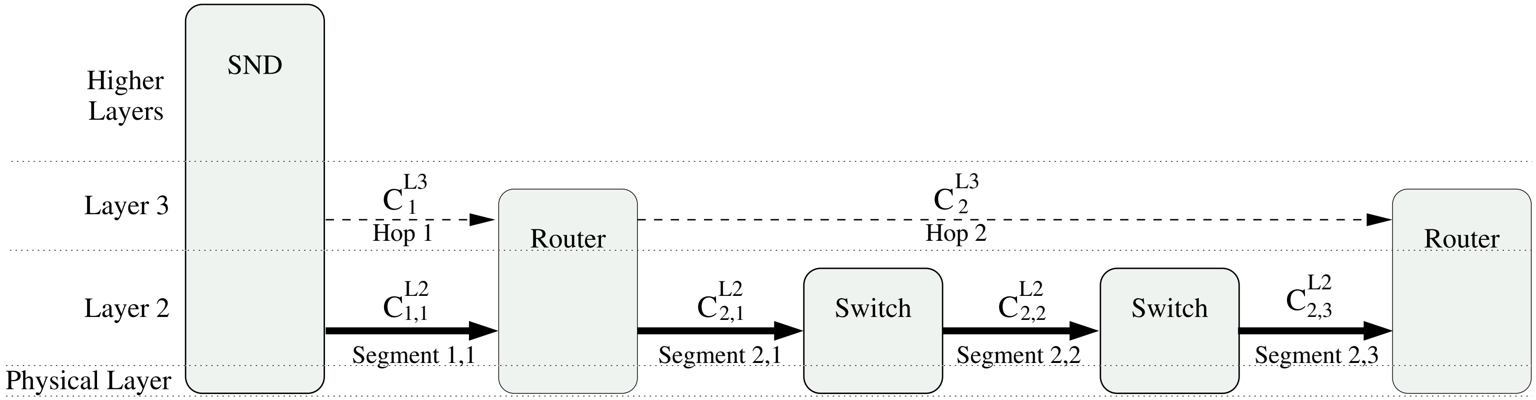
- •A and B both have Fast Ethernet network interface cards
- •What is the capacity from A to B ?

Estimated capacity

Tool	Capacity estimate
pathchar	49.0±1.5Mbps
clink	47.5±1.0Mbps
pchar	47.0±1.0Mbps
pathrate	97.5±0.5Mbps
bprobe	95.5±2.0Mbps

- What went wrong?
  - ⋆ The tool or <u>the methodology used</u> ?

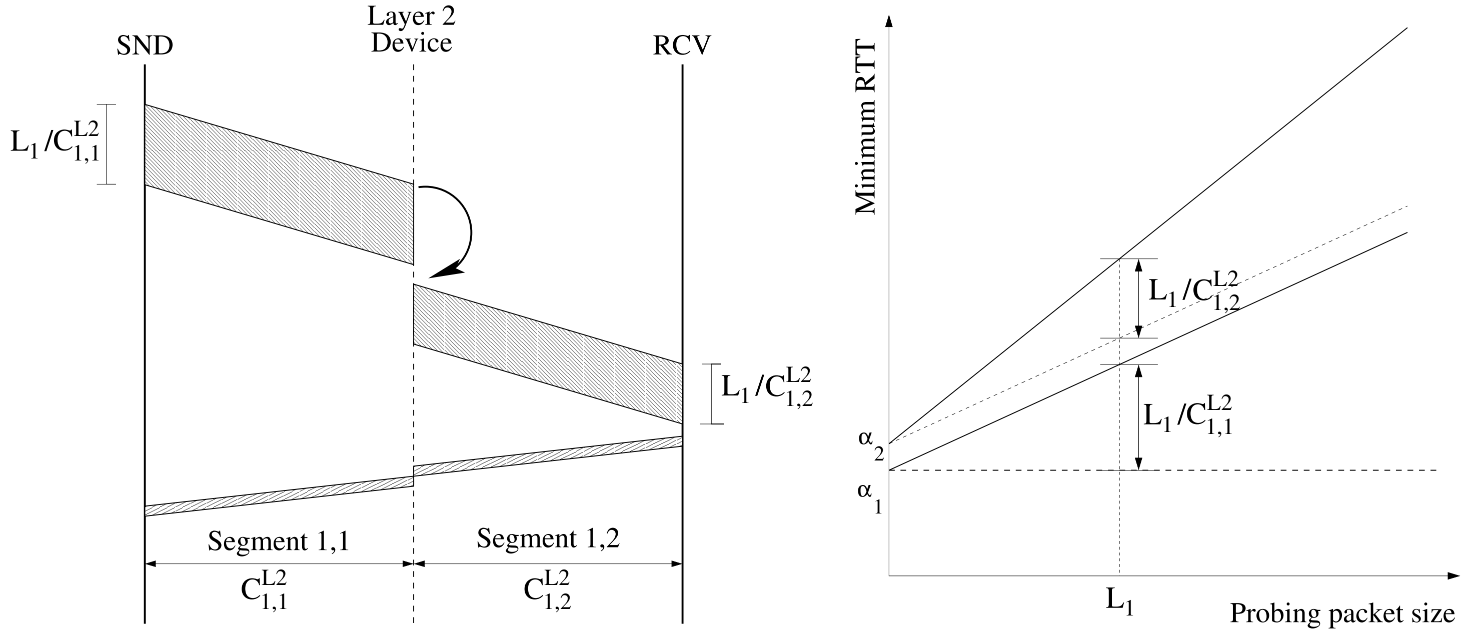
Links : Layer3 (L3) vs Layer2 (L2)



- Hop : Link at layer 3
- Segment : Link at layer 2
- If an L3 hop has intermediate L2 devices
  - ⋆ It will have more than one L2 segment
  - ⋆ Different L2 segments may have different capacities
  - ⋆ Capacity of i^{th} L3 hop that consists of M_i L2 segments

$$C_i^{L3} = \min_{j=1 \ldots M_i} \{C_{i,j}^{L2}\}$$

L2 store-and-forward devices & serialization delay



$$\beta_1 = \frac{1}{C^{L2}_{1,1}} + \frac{1}{C^{L2}_{1,2}}$$

Estimated capacity: $\hat{C}^{L3}_1 = \dfrac{1}{\frac{1}{C^{L2}_{1,1}} + \frac{1}{C^{L2}_{1,2}}} \leq C^{L3}_1$ (correct capacity)

In general, $\boxed{\hat{C}^{L3}_I = \dfrac{1}{\sum_{j=1\ldots M_I} \frac{1}{C^{L2}_{I,j}}}}$

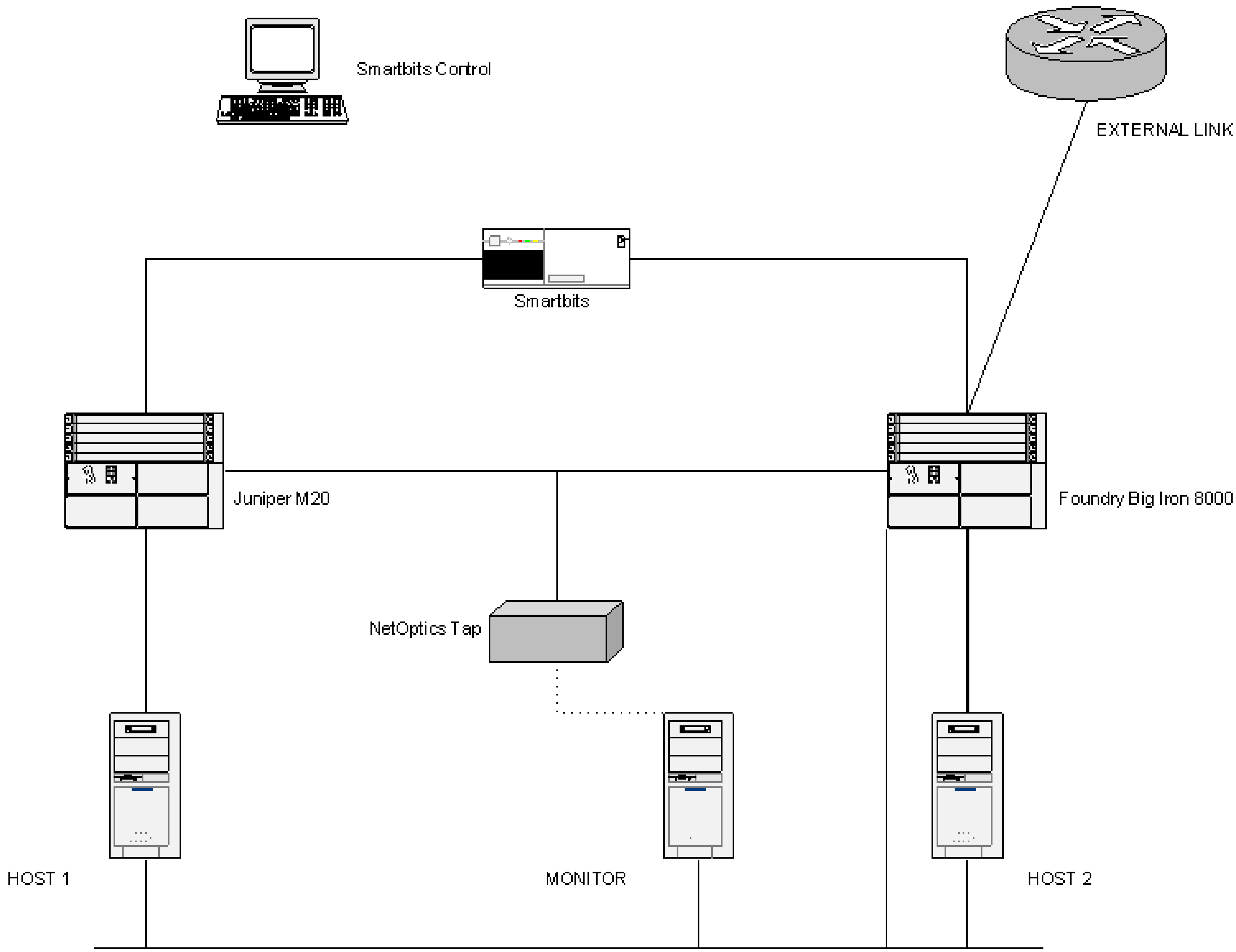
Effect of L2 devices on per-hop capacity estimation

- L2 devices cannot be detected by VPS technique
  - ⋆ do not decrease TTL field
  - ⋆ do not generate ICMP packets
- Store-and-forwarding affects capacity estimate
  - ⋆ increase RTT proportional to the packet size
  - ⋆ change relation between β and capacity

CAIDA/$CalIT^2$ bandwidth estimation test lab

high speed isolated test network with
traffic generation
and data measurement capabilities

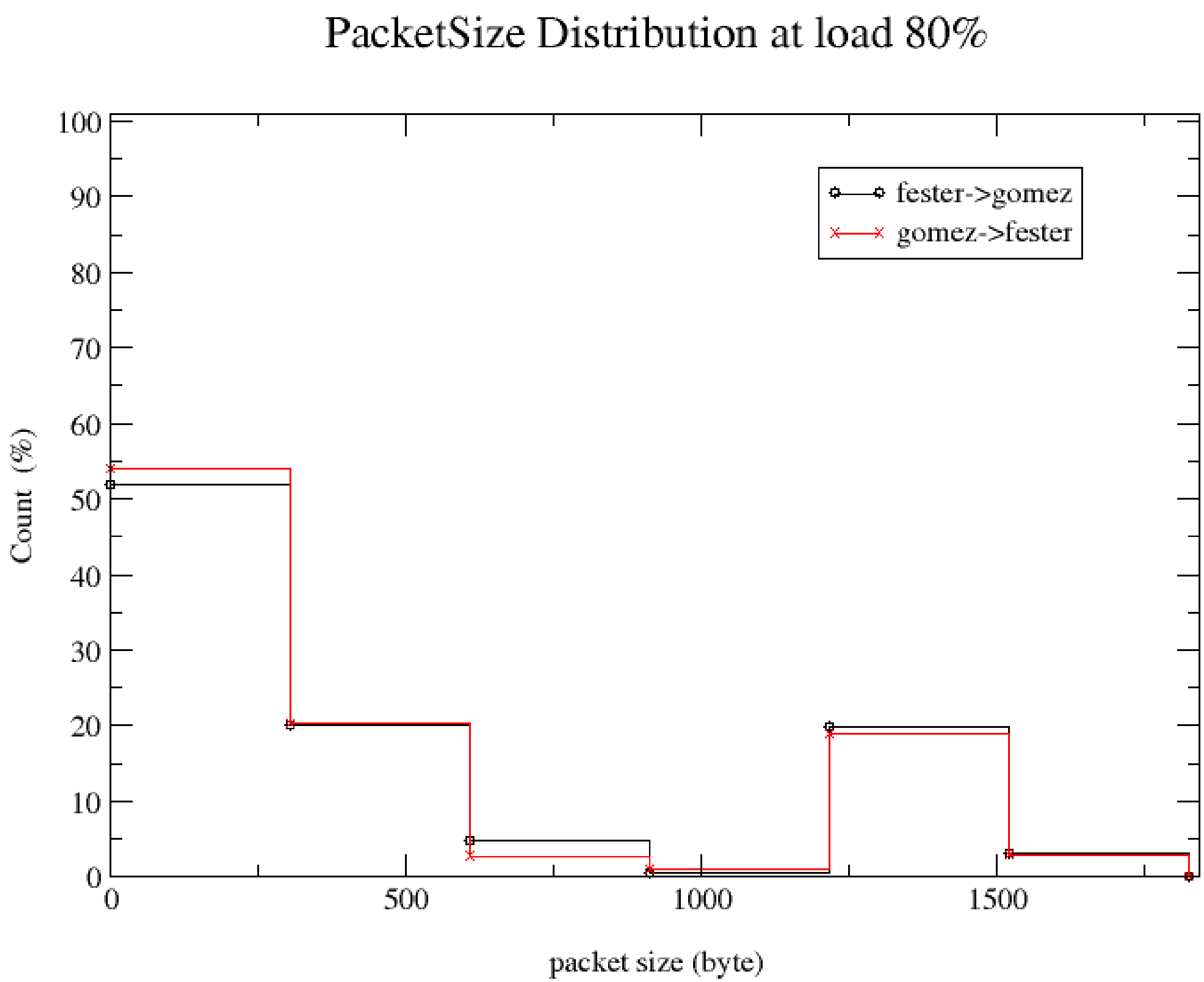
Bandwidth Estimation Test Lab Components



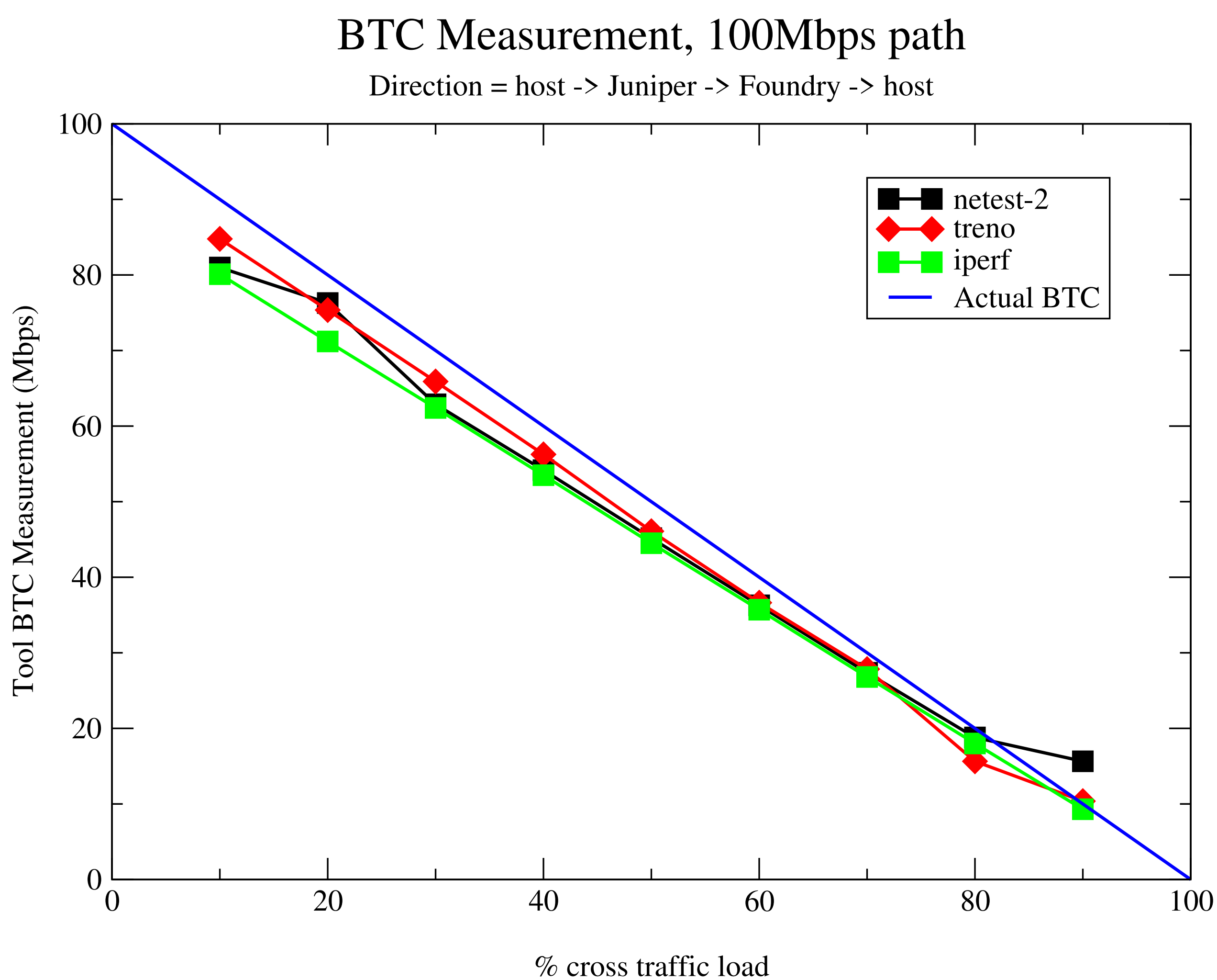
CAIDA/$CalIT^2$ Bandwidth Estimation Experiments

- Intel Pro/1000SC GigEther end hosts (gomez; fester) 9K MTU
- 9 or more identical runs of each tool from host gomez -> fester
- 9 or more identical runs of each tool from host fester -> gomez
- SmartFlow GUI to set cross-traffic load
- Perl scripts run tools and store output
- run iperf as reference indicator of TCP response to cross-traffic
- ran ABw manually (src sender, reflector, src receiver)

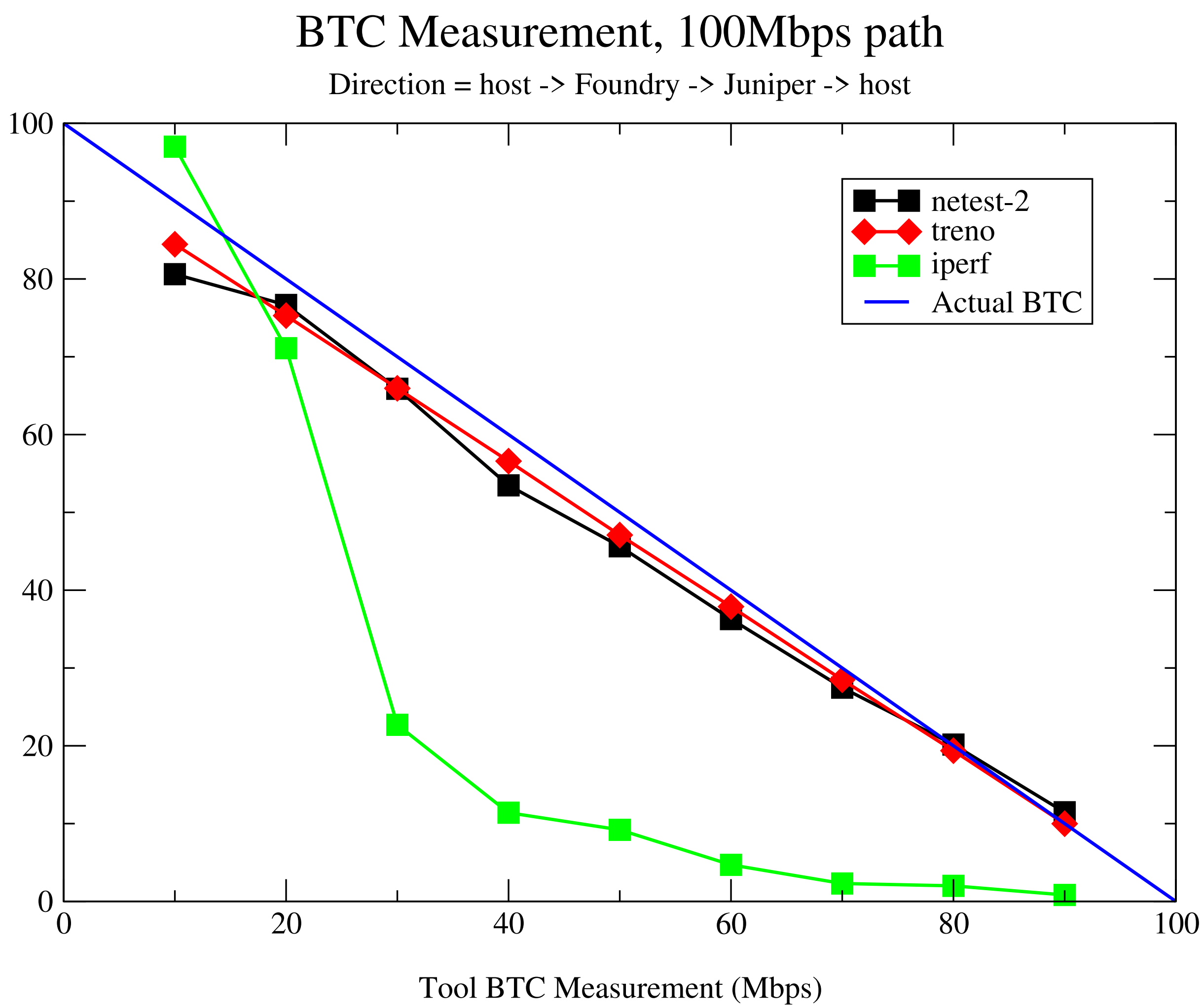
Reproducible Cross-Traffic



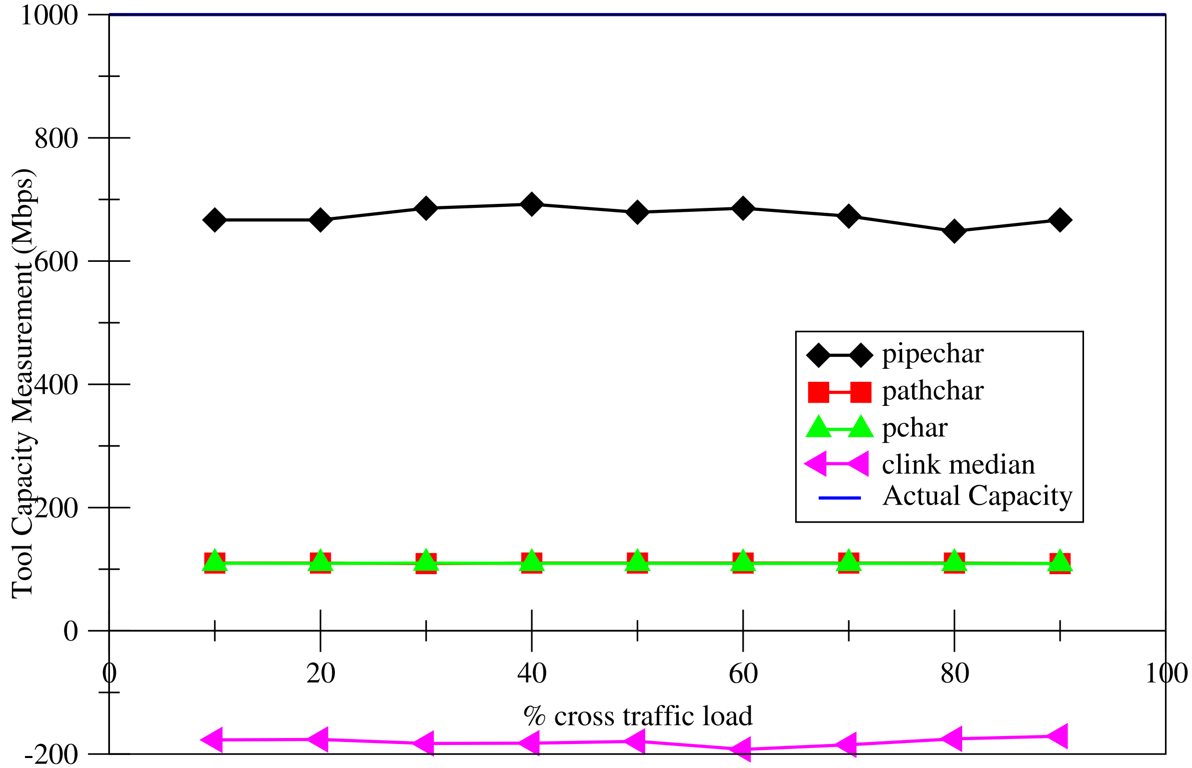
netest2, iperf, and treno respond similarly to cross-traffic



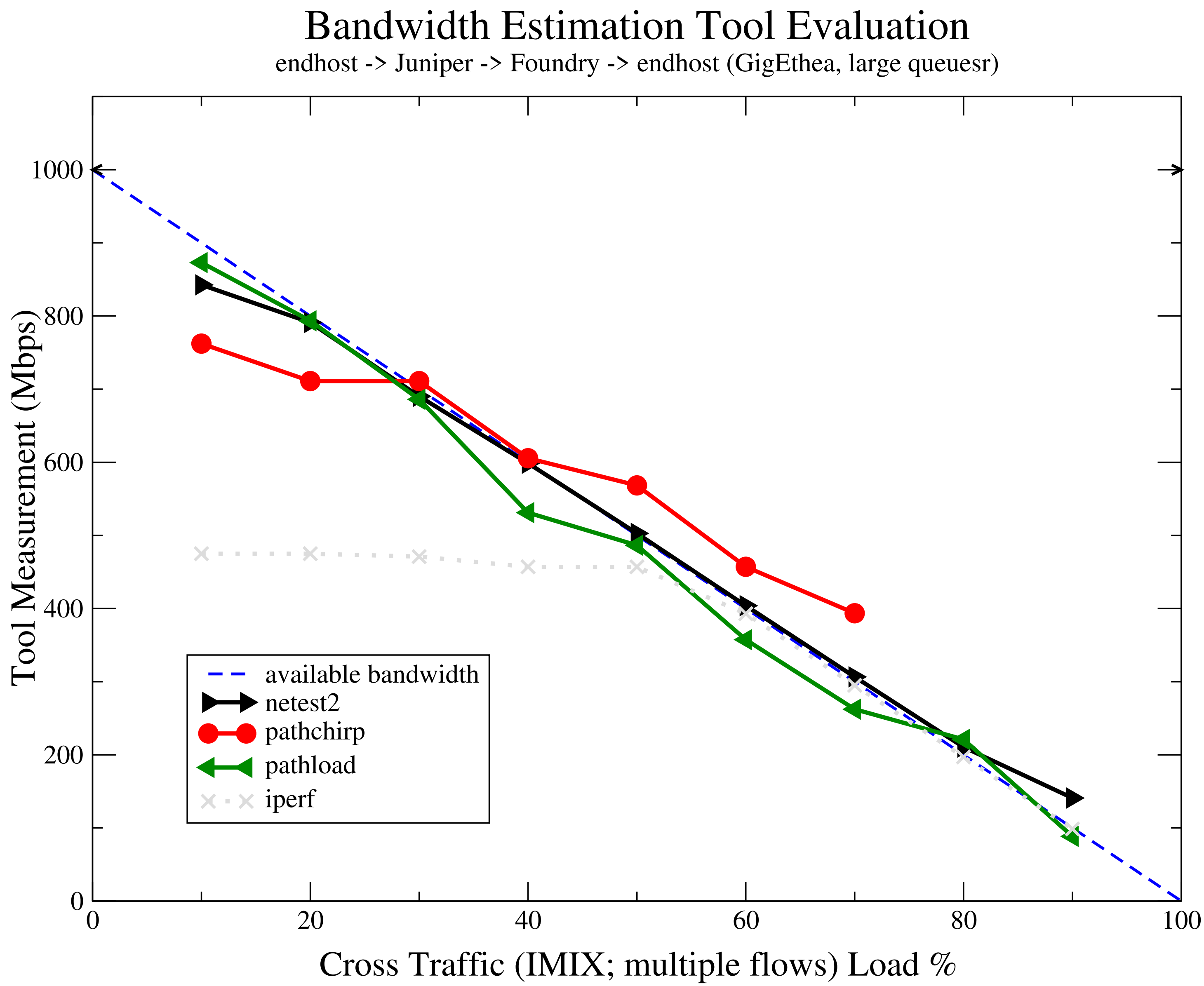
same path; opposite direction; smaller queues



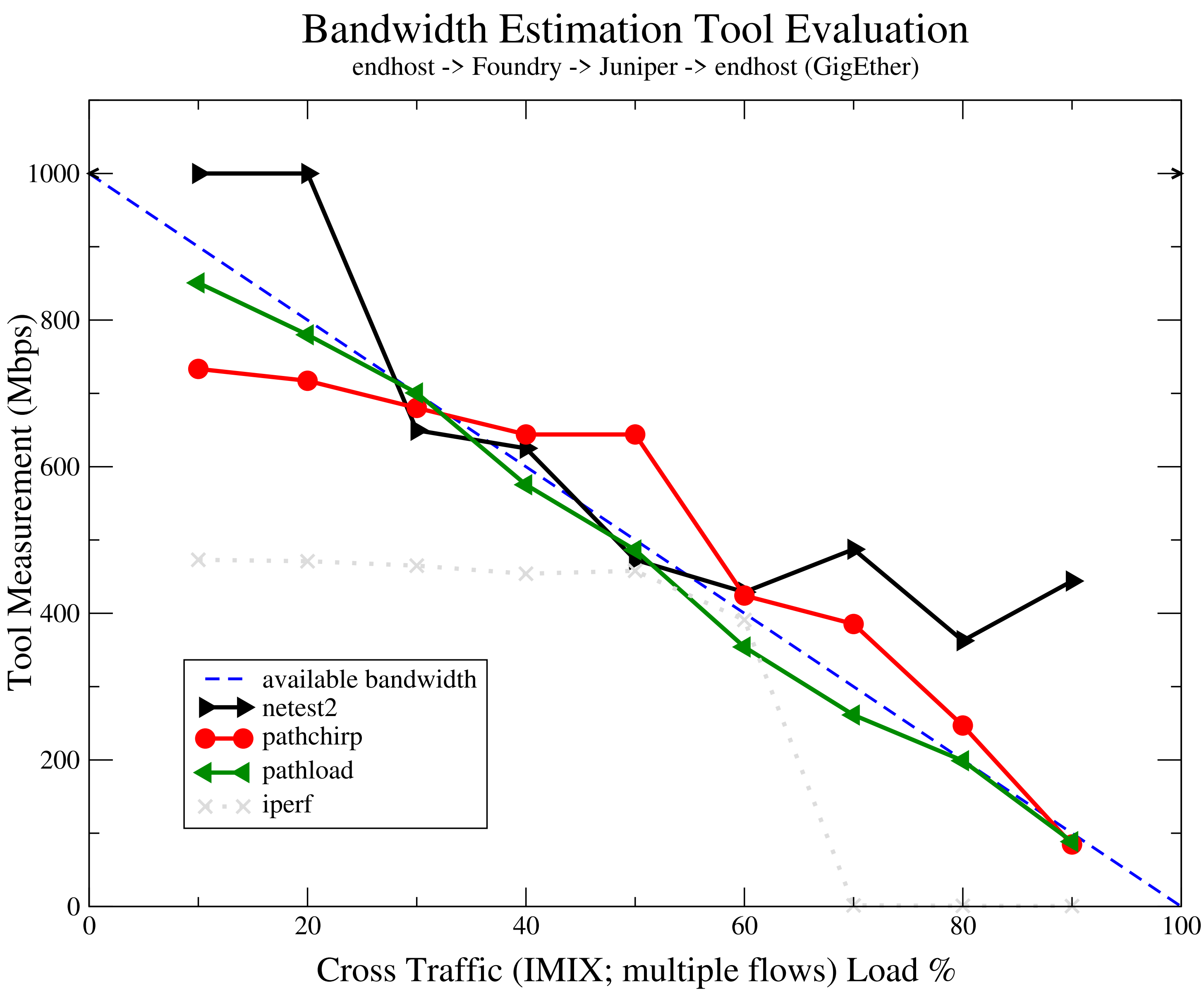
pathchar, pchar, pipechar and clink fail at GigEther



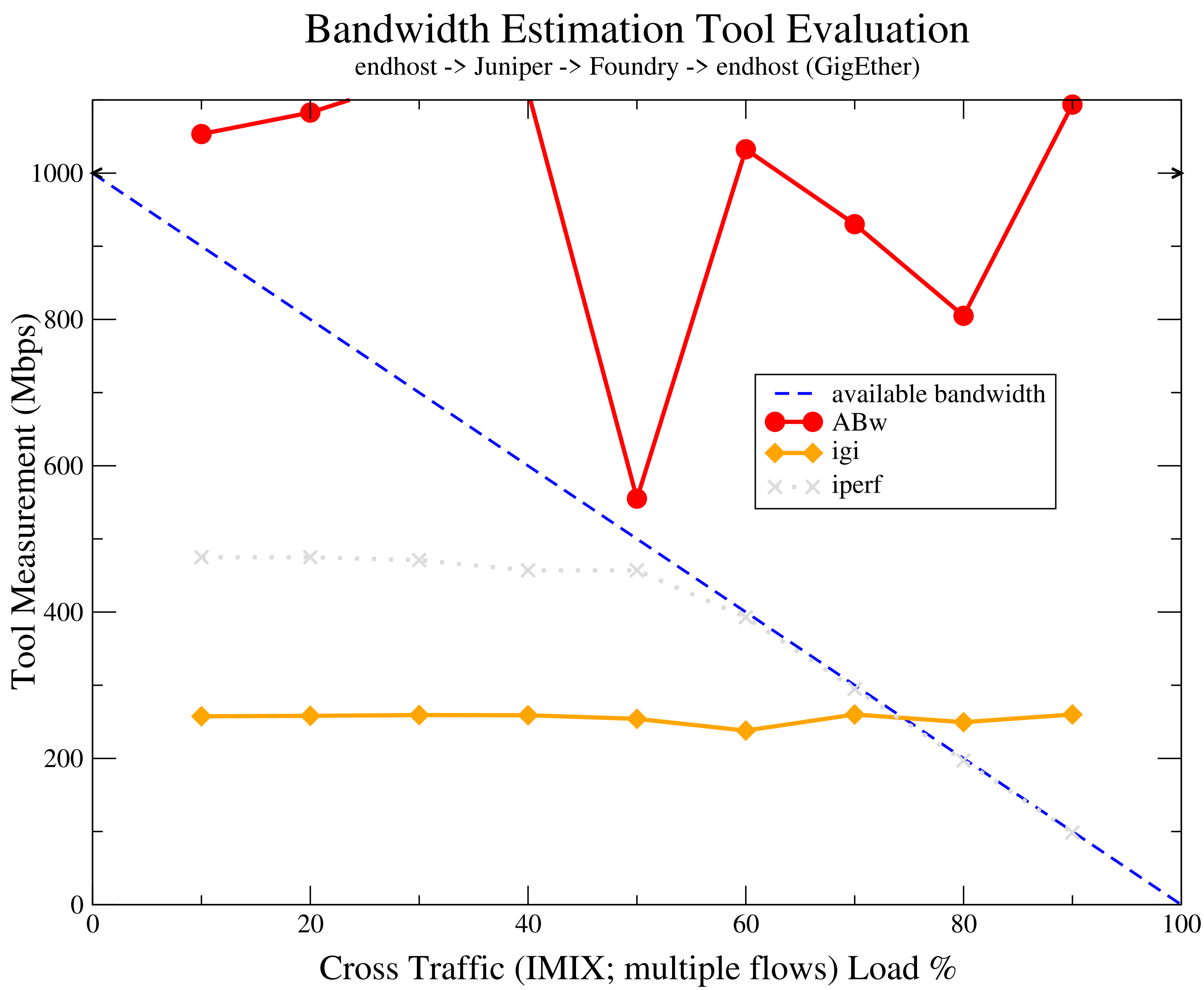
tools that measure available bandwidth



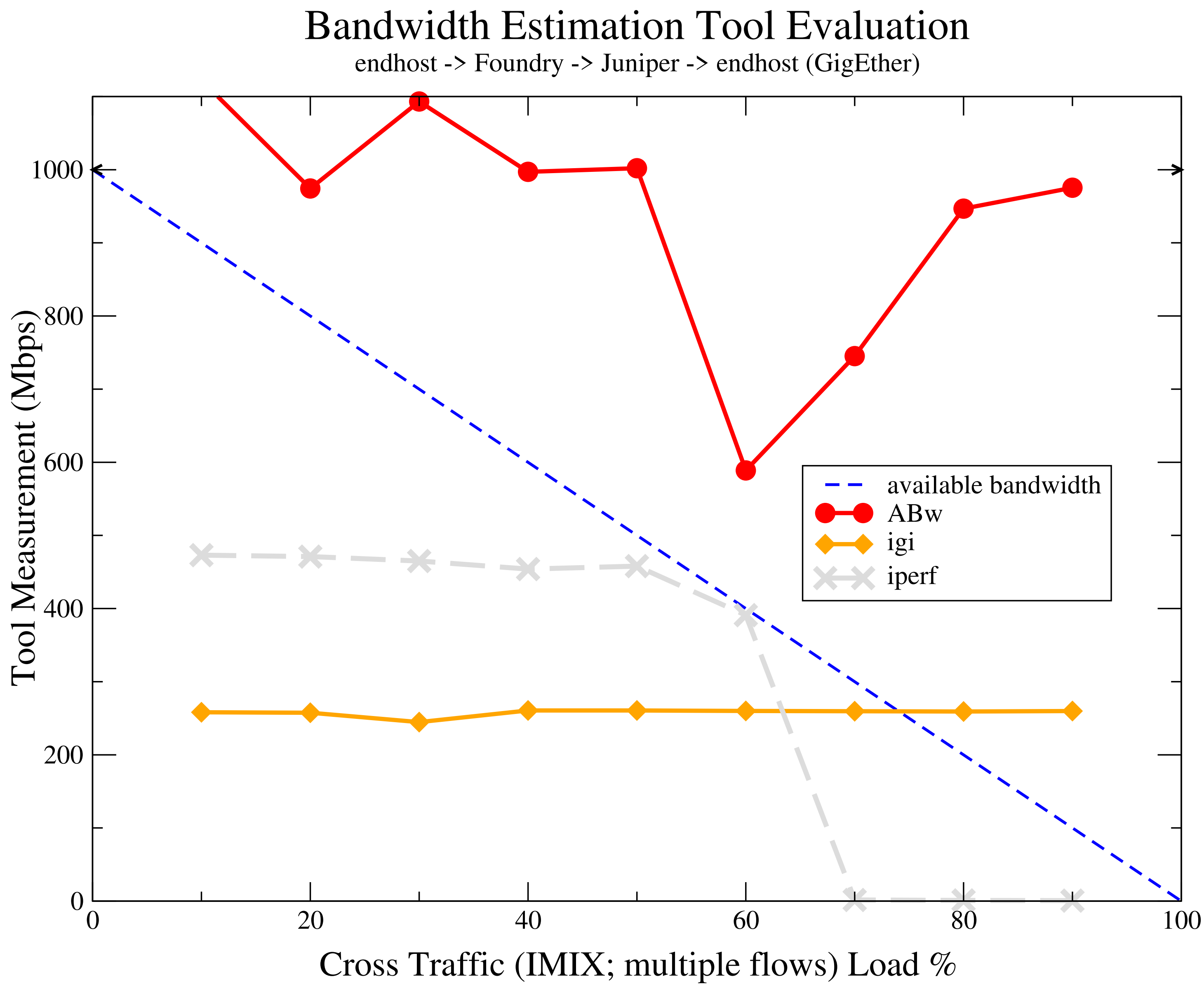
...in opposite directions, w/subtle differences



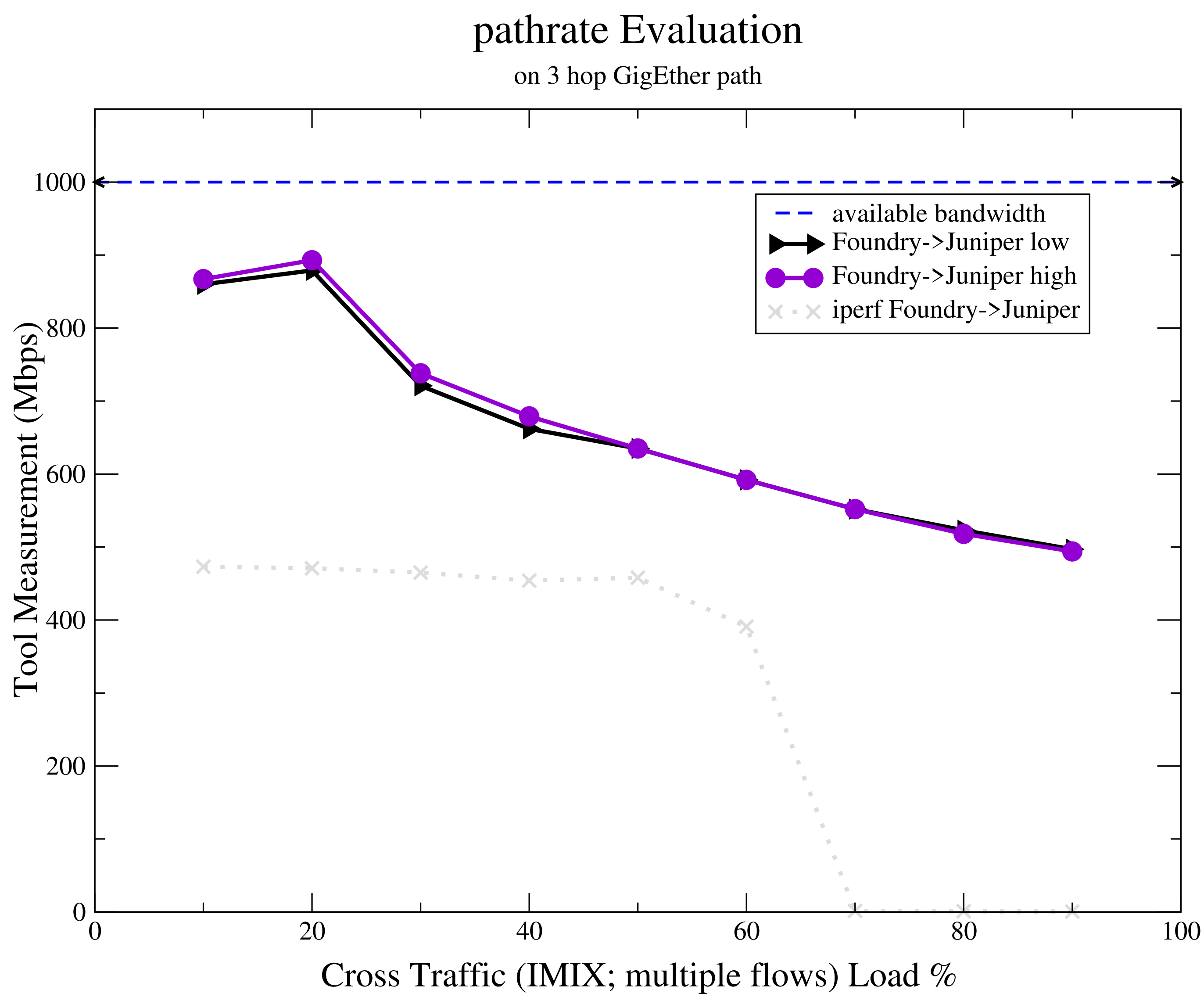
IGI and ABw do not (yet) measure available bw



...in either direction



capacity measurements GigEther Foundry->Juniper



Lessons learned

- E2E measurements are sensitive to L2 and L3 chars
- VPS tools (pathchar, and pchar, clink) underestimate layer 3 capacity when layer 2 devices are present
- for measurement of available bandwidth, all of our testing indicates that pathload is currently the most accurate
- pathrate's underestimation error increases at higher cross-traffic loads
- this project unique in emphasis on testing tools under a variety of realistic but reproducible conditions
- access to developed testbed translated into better tools
- systematic use of these tools in the real world is imperative to their evolution

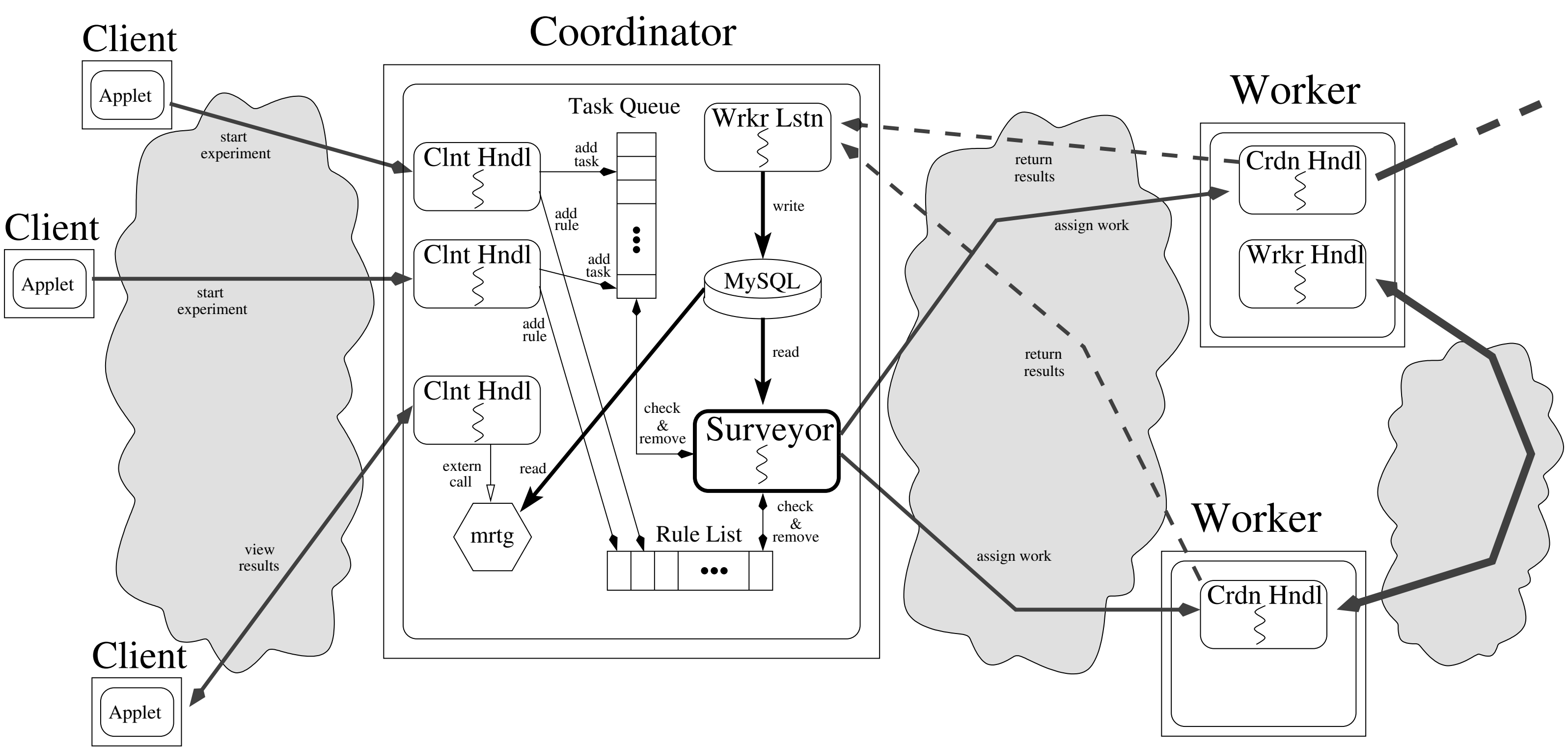
ANEMOS

Autonomous NEtwork MOnitoring System

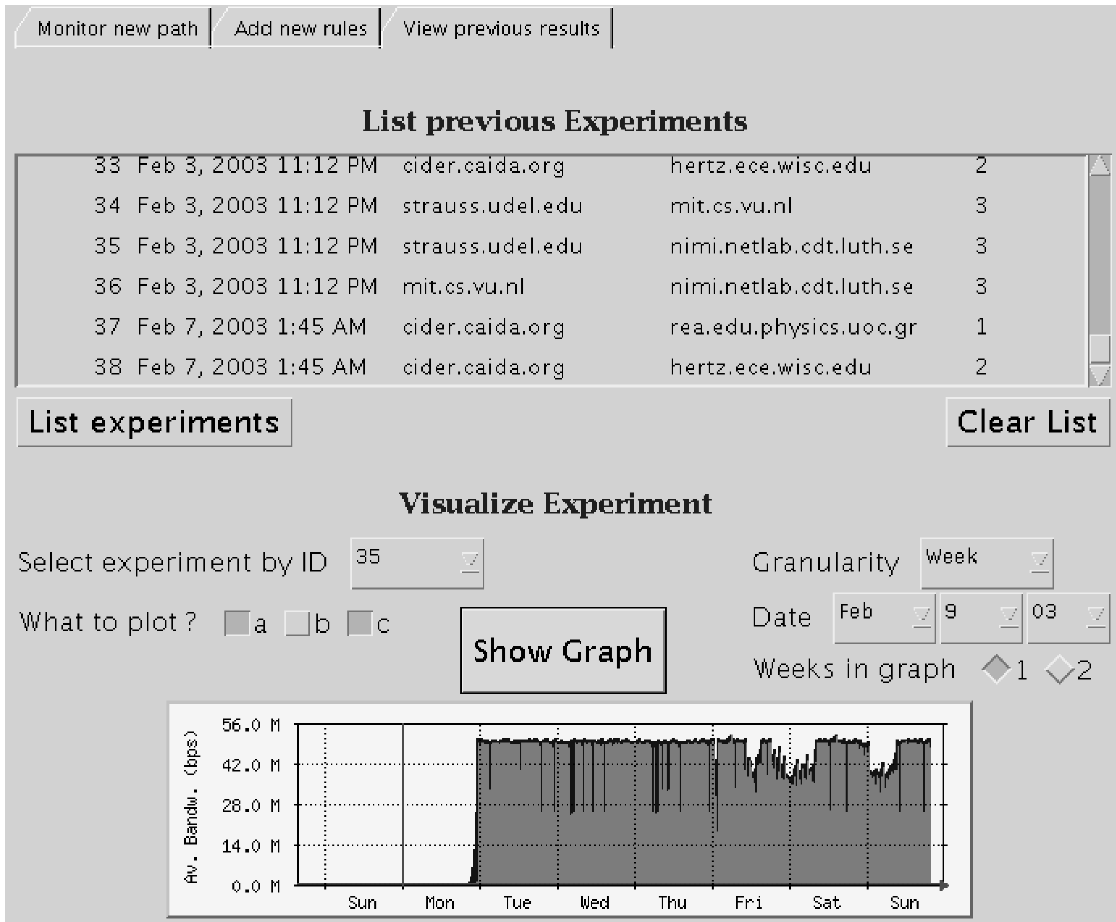
ANEMOS overview

- ANEMOS provides monitoring infrastructure for multiple paths
- Measurements can be visualized using MRTG
- Measurements can be archived using MySQL
- Sophisticated rules can automate data analysis
- Real-time analysis can detect anomalies
- ANEMOS currently integrates ping and pathload

System architecture



ANEMOS snapshot



ANEMOS rules

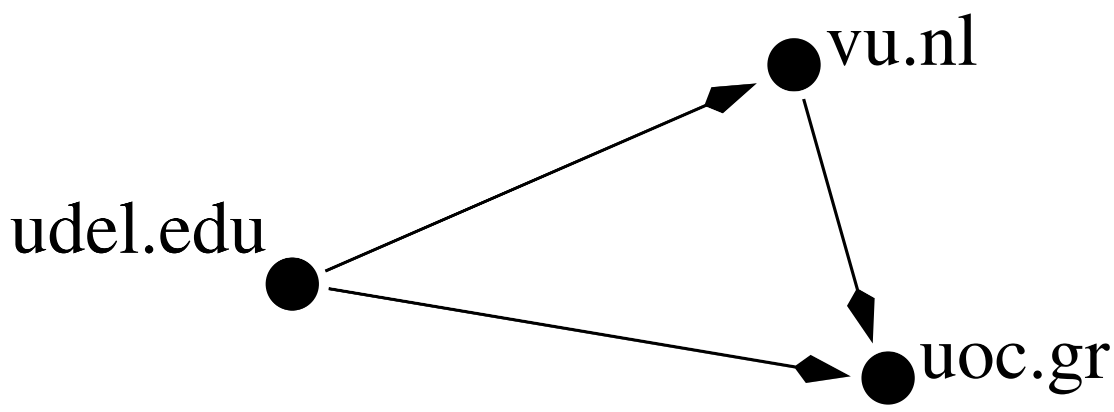
- Automatically detect changes in path performance
- Example:

$$\text{RTT}(\Delta T_1) > \text{RTT}(\Delta T_2) + 25\text{msec}$$
$$\text{AvailBW}(\Delta T_1) < 25\% \times \text{AvailBW}(\Delta T_2)$$
$$\text{LossRate}(\Delta T_1) > 5\%$$

- Correlate characteristics of different paths
- Example:

$$\text{RTT}(\text{path}_1) > \text{RTT}(\text{path}_2) + 30\text{msec}$$
$$\text{AvailBW}(\text{path}_1) < 50\% \times \text{AvailBW}(\text{path}_2)$$
$$\text{LossRate}(\text{path}_1) > \text{LossRate}(\text{path}_2)$$

An ANEMOS application



- Overlay network routing
- Route-1: UDel directly to UOC
- Route-2: UDel to VU to UOC
- ANEMOS can detect route with lowest total RTT automatically

SOBAS

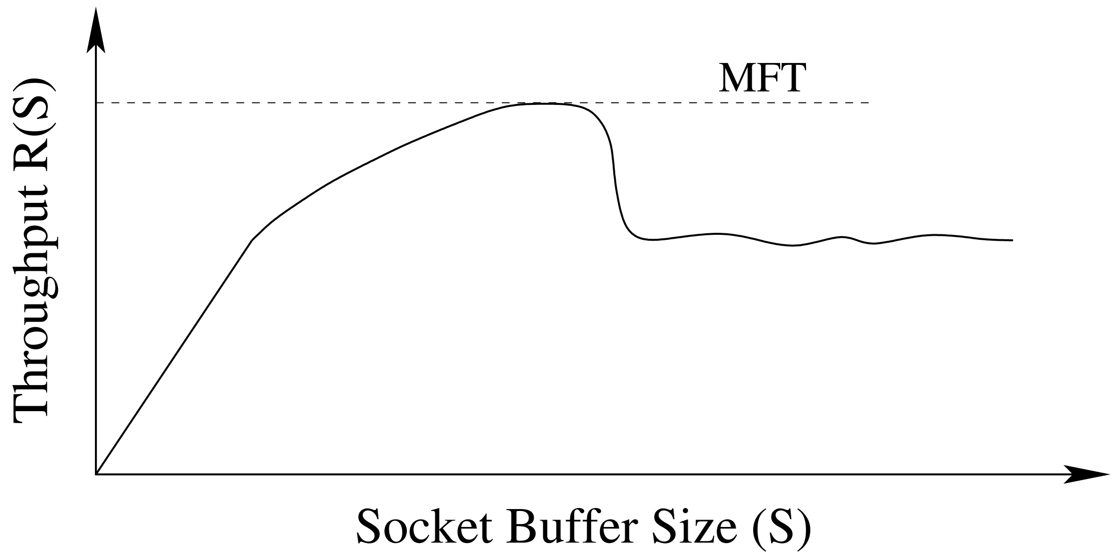
SOcket Buffer Auto-Sizing

SOBAS overview

- Application-layer mechanism to automatically adjust socket buffer size
- Objective: achieve Maximum Feasible Throughput (MFT)
- Does **not require:**
  - Changes to TCP
  - Prior knowledge of path properties
  - Prior knowledge of cross traffic characteristics
- Assumes **end-system support for:**
  - Dynamically changing send/receive socket buffer size
  - Large enough maximum socket buffer limit

Maximum Feasible Throughput (MFT)

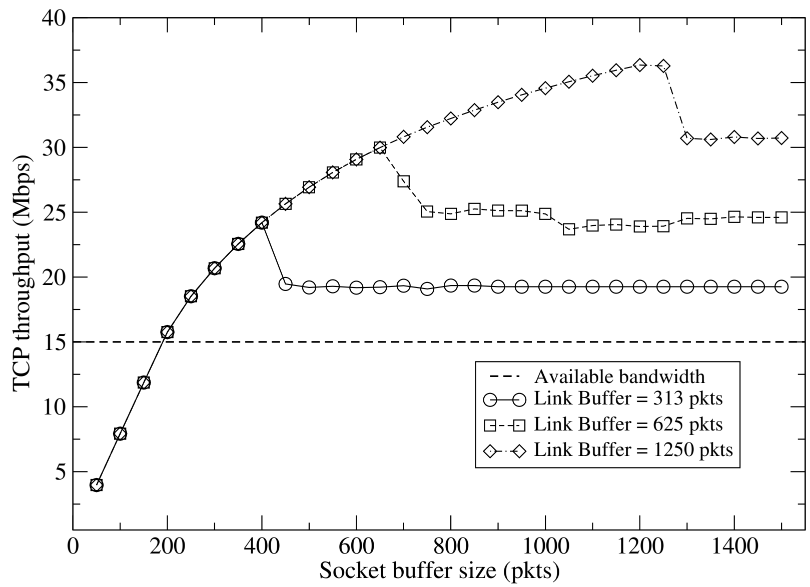
- Consider TCP throughput $R(S)$ as function of socket buffer size S



- For a given network path and cross traffic load:
  - MFT$= R(S)$ such that $\frac{\partial R}{\partial S} = 0$
  - SOBAS attempts to dynamically identify the *optimal* socket size that leads to MFT

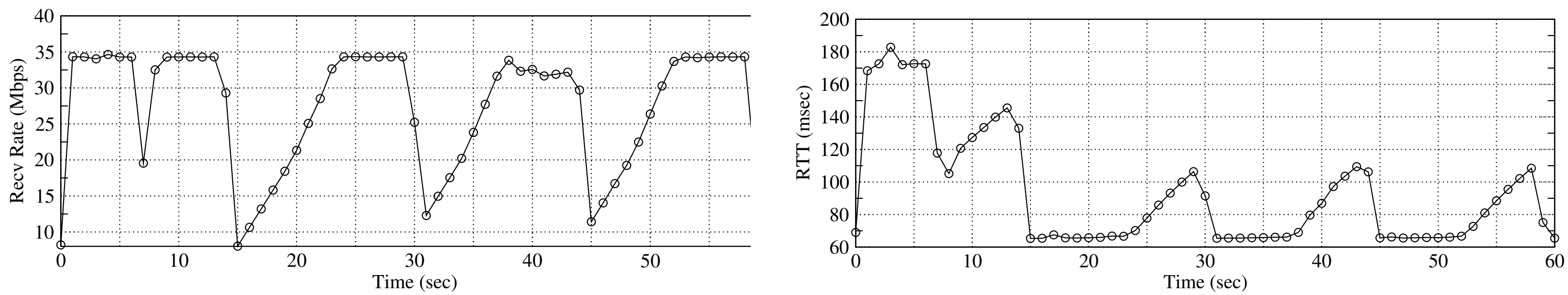
MFT in non-congested path

- Non-congested path (A=15Mbps), cross traffic: persistent TCP connections



- $MFT > A$ because of bandwidth sharing with cross traffic
- Large net buffers cause higher MFT (higher RTTs for cross traffic)

Basic idea in SOBAS

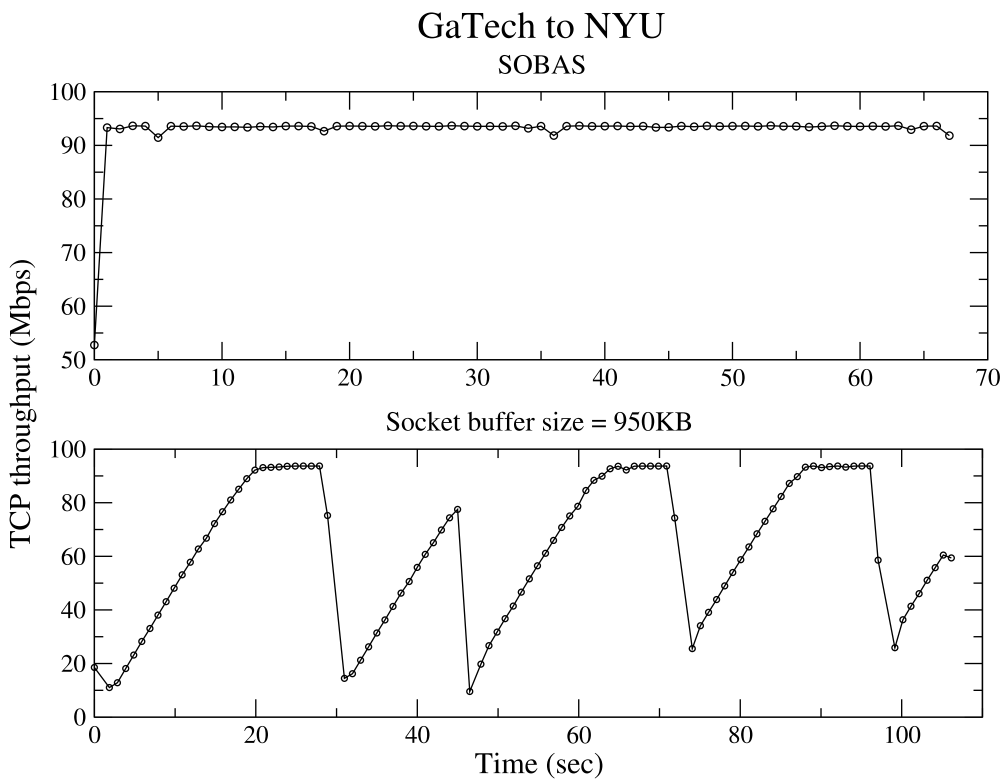


- Receive rate increases until MFT is reached: rate saturation point
  - After rate saturation, receive rate flattens
  - Further congestion window increases cause increased queues
  - Packet drops occur if congestion window exceeds $CT + B$ (BDP + LinkBuffer)

Basic idea in SOBAS (cont')

- Receiver measures goodput R of TCP transfer
- When rate is flattened, limit socket buffer size to $B_r = R \times RTT$
- Receiver measures RTT using periodic UDP ping packets to sender

SOBAS experimental result



- Optimal socket buffer size at maximum lossless receive-throughput

Looking forward

DOE technology integration

Long term: bandwidth estimation grid portal

Short term: workshop at SDSC 9-10 dec 2003

Long term: Emerging Research Areas

- Once we have a handle on bandwidth estimation, each of these is a whole research area:
  - Congestion control and TCP: automatic socket buffer sizing
  - Overlay networks: configure overlay routes
  - Content distribution networks: select best server
  - Streaming applications: adjust encoding rate
  - SLA and QoS verification: monitor path load
  - End-to-end admission control: check for sufficient bandwidth
  - Peer-to-peer networks: construct application-layer topology
  - Interdomain traffic engineering: select egress ISP

Short term: bandwidth workshop at SDSC

- CAIDA ISMA 9-10 december 2003
- between ‘invitation only’ & ‘open call for participation’
- announce september 2003
- growing number of bw-est researchers
- operational engineering folks, especially ESnet
- in collaboration with IRTF’s IMRG
- DOE-sponsored: cost of workshop is 1-2 page abstract/presentation
- steering committee Constantinos, kc, Mark Allman (NASA), tbd

Contact and further information

- dovrolis@cc.gatech.edu, Georgia Tech
- kc@caida.org, CAIDA/UCSD

- http://www.pathrate.org/
- http://www.caida.org/projects/bwest