are raw bandwidth testbeds worth the investment?

Matt Mathis, PSC/NLANR
mathis@psc.edu
www.psc.edu

kc claffy, UCSD/SDSC/CAIDA
kc@caida.org
www.caida.org

raw b/w network research testbeds

- historically under used, starting w/ gigabits in the early 80's
- relatively few papers considering the tax dollars invested
- hard to identify indirect results
  - (improvements in real products)
- poor utilization even when connected to production campus nets
  - TCP in the field is lame
- good PR in some circles

typical non-net-researcher feedback

- basically unhappy
- experience really poor local performance
- view testbed as a waste
- unhappy to hear about "solved" problems when theirs are not
- users complaints can reach congress

typical testbed results/solns

mostly single point solutions:

- hardware typically not ready for products
- non–standard s/w or configs, e.g. larger than std MTU
- non–general optimizations
- Moore law advances suggest that next generation general solution will overtake point solution

hero numbers are not cost–effective

persistently unsolved research problems

- TCP dynamics
  - we do not understand TCP
  - simulation of unproven relevance
- routing dynamics
  - instability, load–balancing, propagation, bugs
- SLA articulation
  - we do not have a calculus to describe performance
- measurement
  - we do not (know how to) measure real traffic

none are needed (possible) on a raw bandwidth testbed

conclusion

another raw bandwidth testbed is not going to help the network research most needed now

the real problem today is not filling yet another faster empty link, but understanding traffic dynamics on real infrastructure

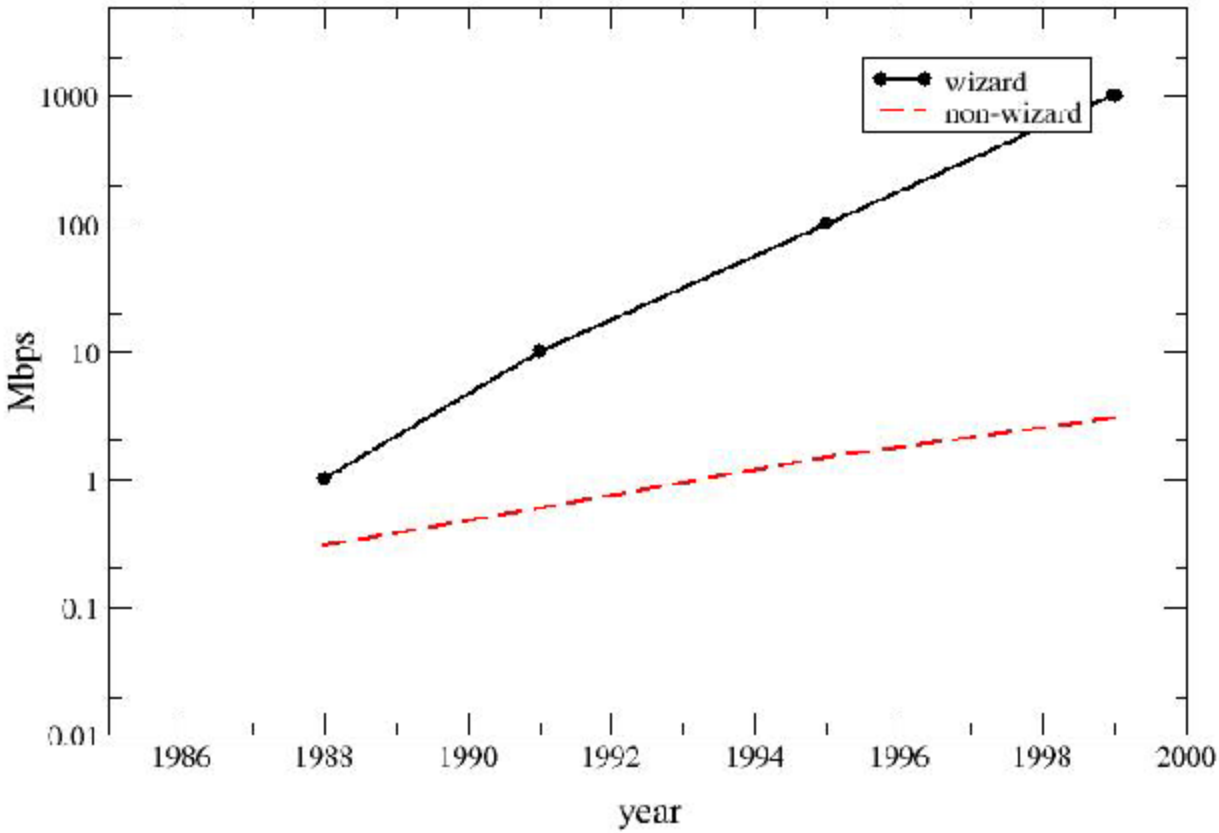
wasted headroom can never be filled until we understand congestion

part 1 end

What is the Important Problem?

The Wizard Gap

(ratio has gone from 3:1 to 300:1 in last decade)



The Wizard Gap

(TCP over a long path)

Year	Wizards	Non–wizards	Ratio
1988	1 Mb/s	300 kb/s	3:1
1991	10 Mb/s		
1995	100 Mb/s		
1999	1 Gb/s	3 Mb/s	300:1

Non–wizards are not happy

More hero numbers are likely to be bad politics

Web100 is attacking this gap

The Web100 project

Key components:

- Better instrumentation within TCP
  - If TCP is slow, just ask TCP why
- Autotune TCP/IP
  - Require less expertise from the users

See: www.web100.org

Impact

- Reduce stack bottlenecks
- Indirectly fix paths
- Indirectly fix applications

Danger – new load levels

Any single pair of (cheap) workstation can congest any OC–3 link.

Any single pair of (expensive) workstations can congest any OC–12 link.

10+ TB/s loads in the core?
100k users * 100 Mb/s

Pandemic congestion

Why not before Web100?

Lame TCP hides path problems

Lame paths hide TCP problems

For nearly everyone debugging TCP is a random walk in the dark

Lame TCP + hidden path problems smooth the traffic and limit peak loads

This should change in about 5 years

Problems

No deployed queue management
No deployed QoS
Pervasive broken link layers
No models for traffic sharing
Shared gigabit problem
No SLA quantification
New load levels

No deployed queue management

With drop tail routers, TCP controls against queue full

This causes huge delays and/or delay variance

Have observed single tuned flows causing 1.2 s RTT

- Zero current users have well tuned flows!

No deployed QoS

TCP requires queues for correct operation
- Web100 will cause queues

Most UDP prefers not to have queues

Why do real time applications work at all today?

Is web100 going to break all delay sensitive applications?

Pervasive broken link layers

(Poor behavior under sustained laminar packet flows)

Queueing problems

- Insufficient queues
- Policing without shaping

Coupling between flows

- Channel acquisition in CSMA/CD Ethernet, wireless, etc

No models for traffic sharing

(How do transient flows impact large flows?)

Matt's first TCP question (1991):
- Half T3 NSF net (22 Mb/s) with 10 Mb/s load
- Best possible FTP was 5 Mb/s
- Where was the missing 5 Mb/s?

No theory either
- (the mice and the elephant problem)

Akin to turbulence

We do not even know the dimensionality of the problem space

Shared gigabit problem

Can a 500 Mb/s application + 500 Mb/s "background" traffic share a 1 Gb/s link?

Can six 100 Mb/s applications + 500 Mb/s "background" traffic equitably share a 1 Gb/s link?

No SLA quantification

How would you write a (multi–provider) service level agreement for a commodity service to support specific data rates to a large number of sites? With specific latency requirements?

Can the next NGI be just common SLA language, say for 500 Mb/s between all R1 university's and research labs?

New load levels

Will ubiquitous well tuned TCP crush the net?

The common theme

We do not understand...

how traffic interacts with other traffic when the net is full

how traffic interacts with links when the net is full

fully utilized networks

We do not understand congestion!
and it has already been much longer than 5 years!

We need a traffic dynamics testbed

Study how traffic interacts with other traffic and underlying infrastructure

Which will have the longest impact?

- Implementing the first 10 Gb/s application?
- Getting an application to fill a 1 GB/s link with 500 Mb/s background traffic

Solving the second problem will create the demand for industry to solve the first

part 2 end

Traffic Dynamics Testbeds

A Traffic Dynamics Testbed

To study how traffic interacts with other traffic and lower layers

Requires carefully managed experiments where innocent traffic is routed over research infrastructure

The Basic Tension

Is the net for the users

or the network researchers?

Traffic dynamics research requires that the network be acceptable to both

- Users want stable, reliable network properties
- Researchers want to change things

Special Requirements

- Parallel standard "production like" infrastructure
  - Must offer similar properties and performance
- Full capacity Interconnects w/ standard infrastructure
- Fast IP routing knife switches to move traffic back and forth between the TB and standard paths.
- An AUP that permits (requires) momentary prime time service interruptions.

Normal requirements

- Lots of modularity, patch panels etc
  - support tinkering with different gear
- Easily programmed variable sub–rate on the links
  - to create and study bottlenecks

Example Experiment Scenario

- Load custom microcode into the TB
- Run test applications
- Re-route (most) I2/NGI/etc traffic over the TB
- Run test applications + real traffic
- Adjust link rates down
  - introduces some congestion
- Rerun test applications + real traffic
- Fast emergency cutback to standard paths

Note that step 1 (custom microcode) is more difficult at high rates

Lost vBNS opportunity

Proposed vBNS AUP: Campuses could use vBNS for whatever they pleased as long as they purchased sufficient commodity connectivity to withstand prime time down time on the vBNS

Conclusion

A testbed that can not easily support the above experiment is not going to help the network research most needed now

The real problem today is not filling yet another faster empty link, but making full use of existing headroom in the current infrastructure