ESnet
ENERGY SCIENCES NETWORK

FAIL-transfer: Removing the Mystery of Network Performance from Scientific Data Movement

Jason Zurawski – zurawski@es.net

Science Engagement Engineer, ESnet

Lawrence Berkeley National Laboratory

XSEDE Campus Champions Webinar
August 20th 2014

U.S. DEPARTMENT OF ENERGY
Office of Science

BERKELEY LAB

Outline

- Introduction & Motivation
- Network Support for Science
- Data Mobility Expectations & Realities
- Preparing the Campus
- Conclusions

ESnet at a Glance

● Universities
● DOE laboratories

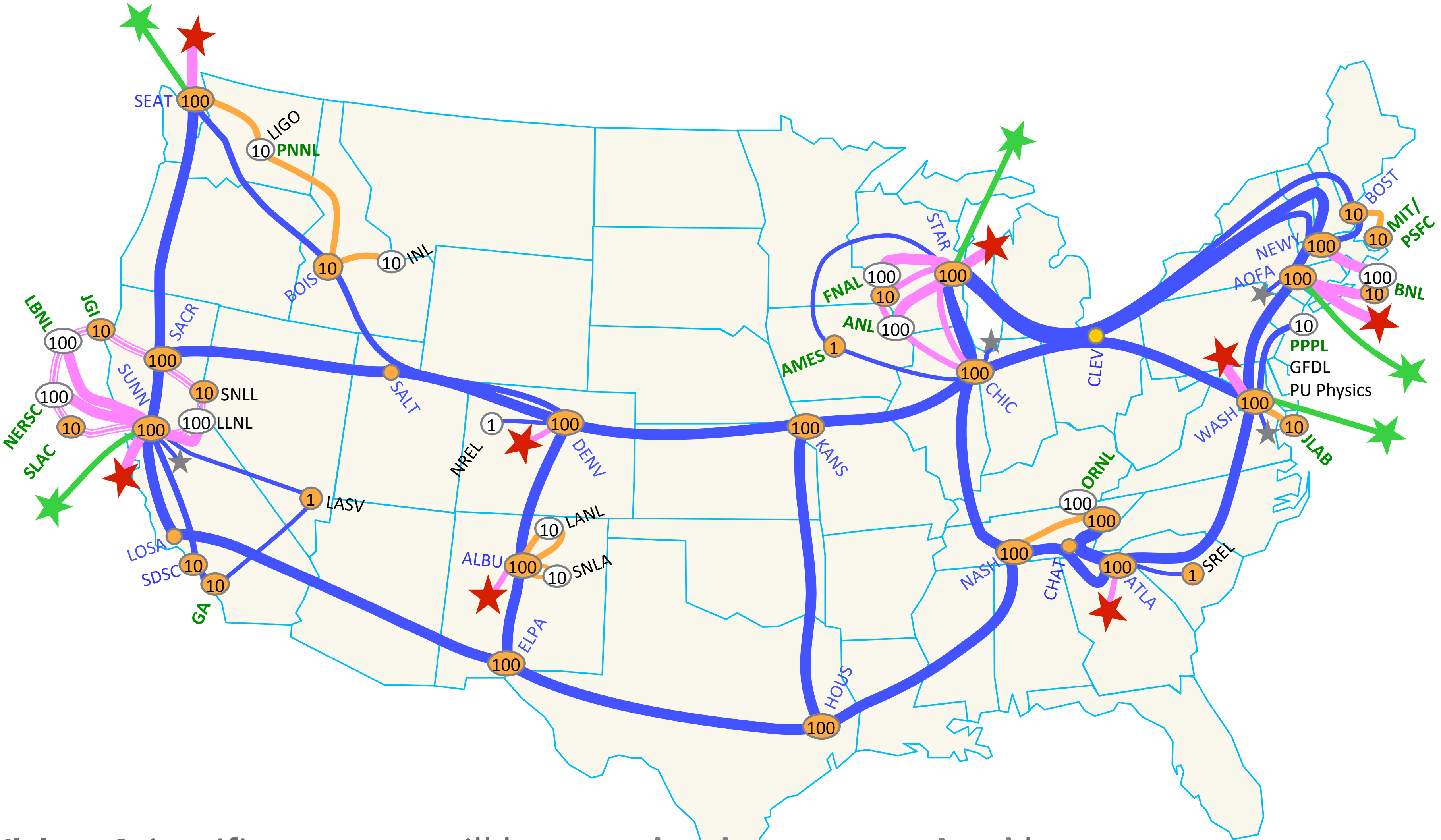
The Office of Science supports:

- 27,000 Ph.D.s, graduate students, undergraduates, engineers, and technicians
- 26,000 users of open-access facilities
- 300 leading academic institutions
- 17 DOE laboratories

- High-speed national network, optimized for DOE science missions:
  - connecting 40 labs, plants and facilities with >100 networks
  - $32.6M in FY14, 42FTE
  - older than commercial Internet, growing twice as fast
- $62M ARRA grant for 100G upgrade:
  - transition to new era of optical networking
  - world's first 100G network at continental scale
- Culture of urgency:
  - 4 awards in past 3 years
  - R&D100 in FY13
  - "5 out of 5" for customer satisfaction in last review
  - ***Dedicated staff to support the mission of science***

ESnet

Network as ~~Infrastructure~~ *Instrument*



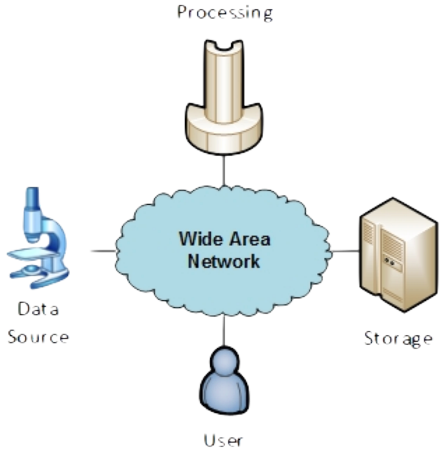
Vision: Scientific progress will be **completely unconstrained** by the physical location of instruments, people, computational resources, or data.

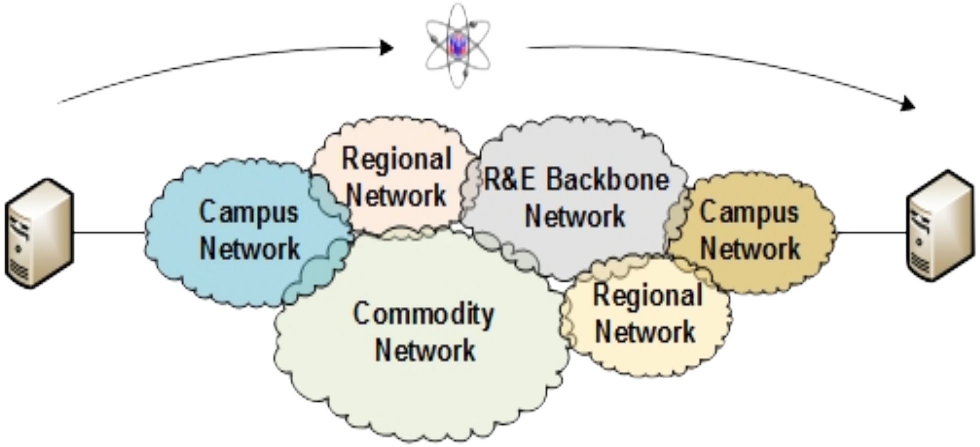
Outline

- Introduction & Motivation
- Network Support for Science
- Data Mobility Expectations & Realities
- Preparing the Campus
- Conclusions

The R&E Community

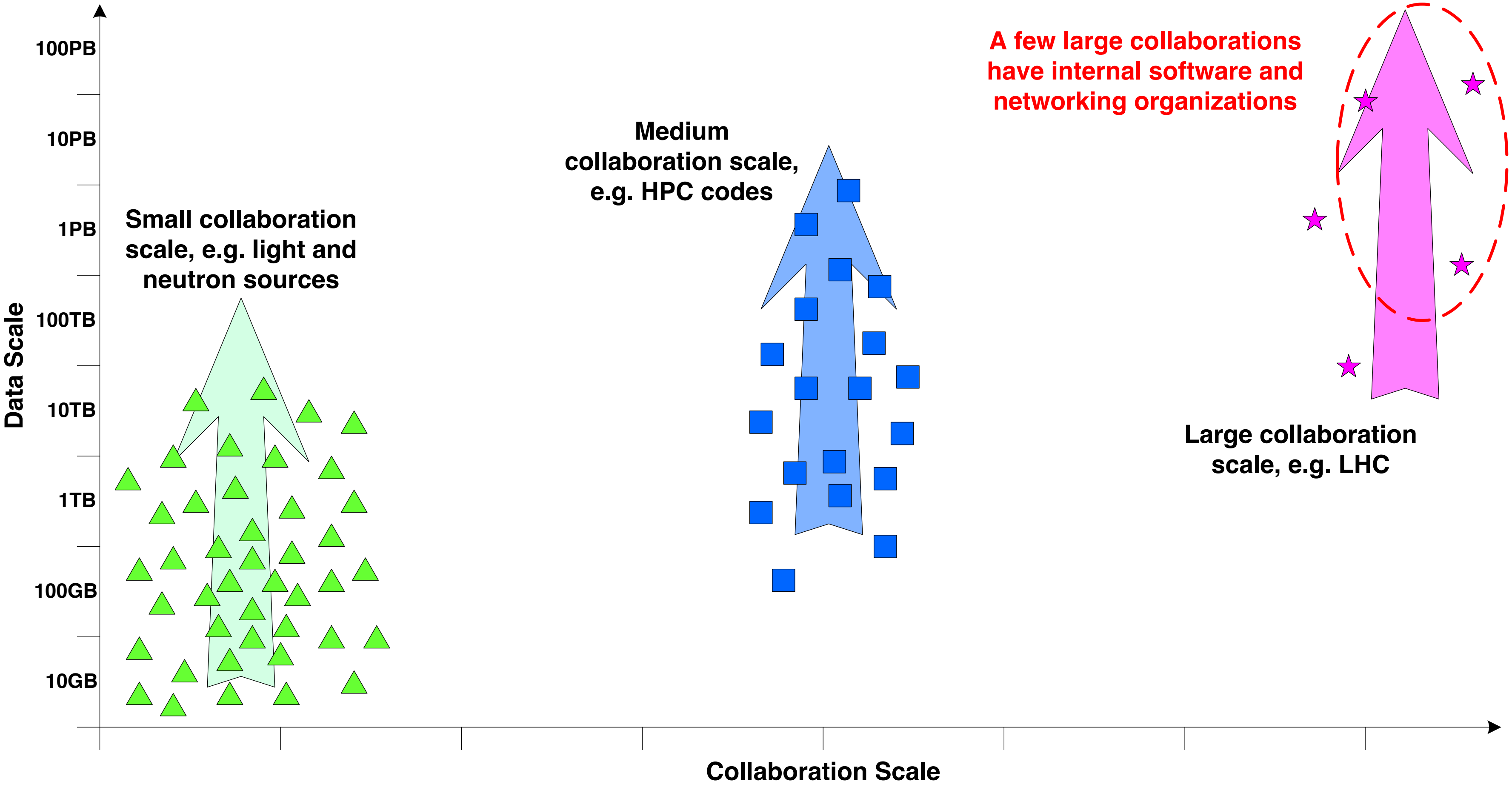
- The global Research & Education network ecosystem is comprised of hundreds of international, national, regional and local-scale resources – each independently owned and operated.
- This complex, heterogeneous set of networks ***must*** operate seamlessly from “end to end” to support science and research collaborations that are distributed globally.





- Data mobility is required; there is no liquid market for HPC resources (people use what they can get – DOE, XSEDE, NOAA, etc. etc.)
  - To stay competitive, we must learn the science, and support it
  - This may mean making sure your network, and the networks of overs, are functional

Understanding Data Trends

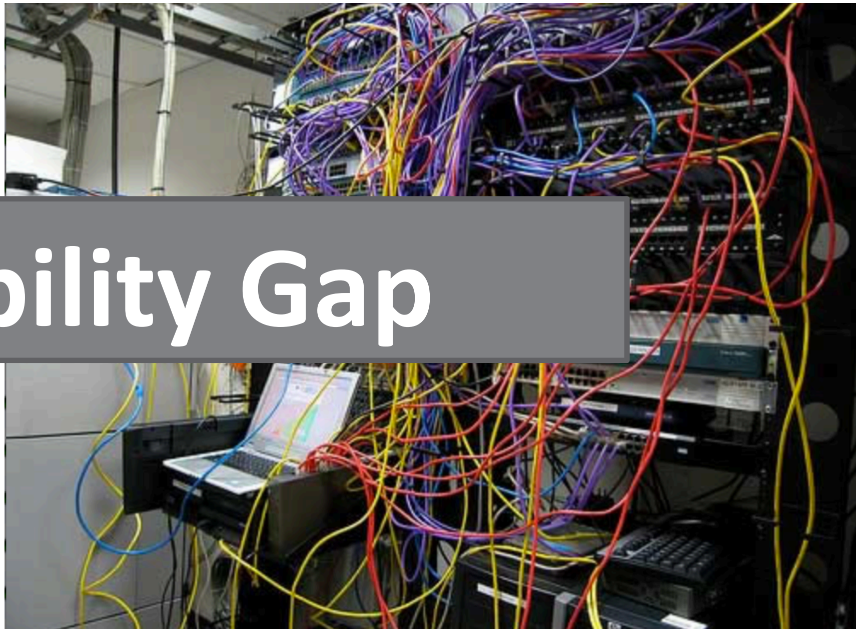


http://www.es.net/about/science-requirements/network-requirements-reviews/

Challenges to Network Adoption

- Causes of performance issues are complicated for users.
- Lack of communication and collaboration between the CIO's office and researchers on campus
- Lack of IT collaborat
- User's performance expectations are low ("The network is too slow", "I tried it and it didn't work").
- Cultural change is hard ("we've always shipped disks!").
- Scientists want to do science not IT support

The Capability Gap



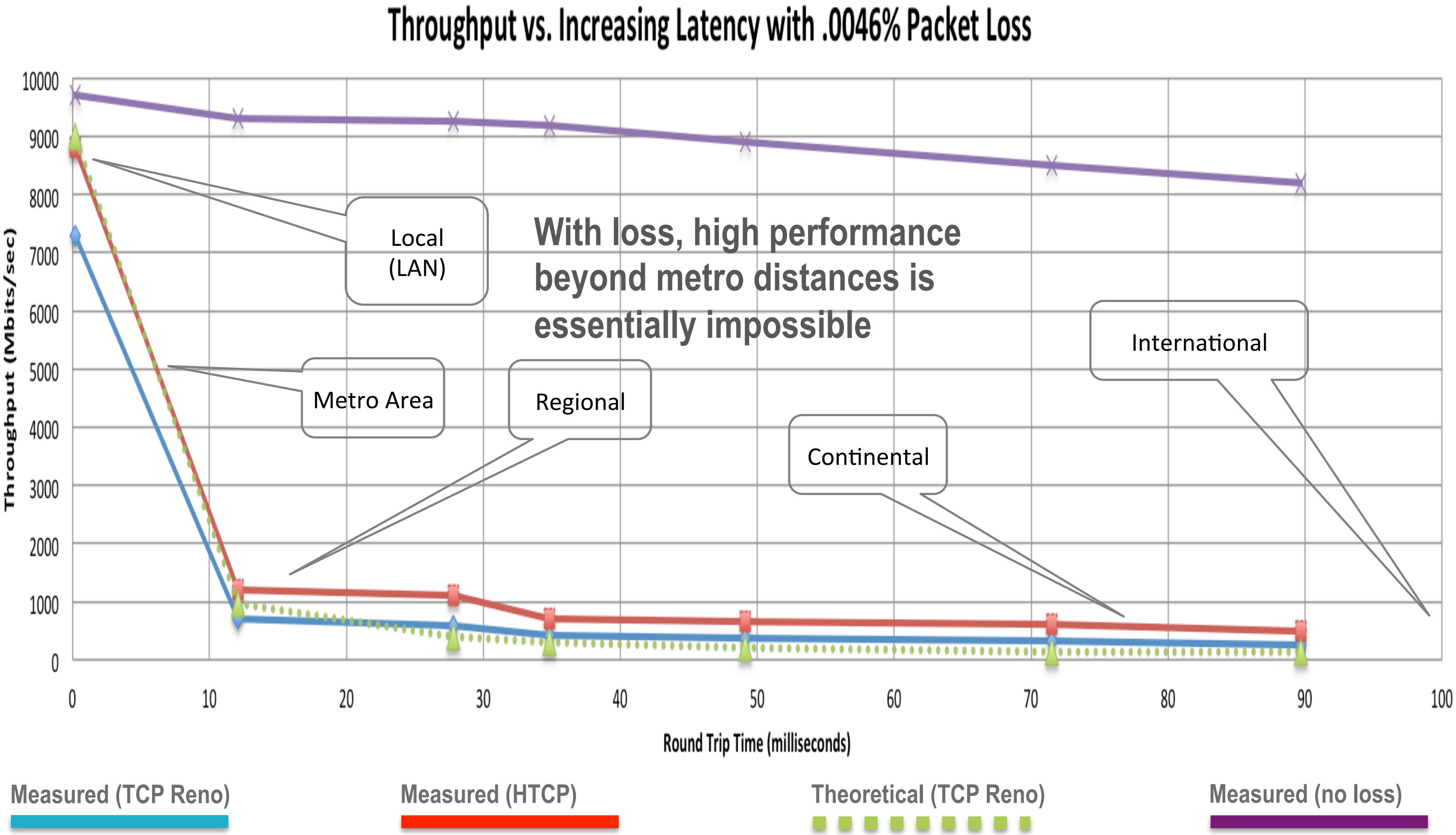
Lets Talk Performance …

"In any large system, there is always something broken."

Jon Postel

- Modern networks are occasionally designed to be *one-size-fits-most*
- e.g. if you have ever heard the phrase "converged network", the design is to facilitate CIA (Confidentiality, Integrity, Availability)
  - This is not bad for protecting the HVAC system from hackers.
- Its all TCP
  - Bulk data movement is a common thread (move the data from the microscope, to the storage, to the processing, to the people – and they are all sitting in different facilities)
  - This fails when TCP suffers due to path problems (***ANYWHERE*** in the path)
  - its easier to work with TCP than to fix it (20+ years of trying…)
- TCP suffers the most from unpredictability; Packet loss/delays are the enemy
  - Small buffers on the network gear and hosts
  - Incorrect application choice
  - Packet disruption caused by overzealous security
  - Congestion from herds of mice
- It all starts with knowing your users, and knowing your network

A small amount of packet loss makes a huge difference in TCP performance



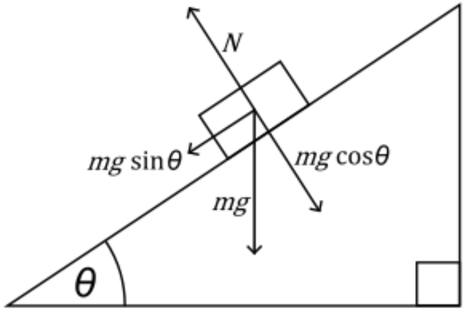
With loss, high performance beyond metro distances is essentially impossible

The Science DMZ in 1 Slide

Consists of **three key components**, all required:

- ***"Friction free" network path***
  - Highly capable network devices (wire-speed, deep queues)
  - Virtual circuit connectivity option
  - Security policy and enforcement specific to science workflows
  - Located at or near site perimeter if possible
- ***Dedicated, high-performance Data Transfer Nodes (DTNs)***
  - Hardware, operating system, libraries all optimized for transfer
  - Includes optimized data transfer tools such as Globus and GridFTP
- ***Performance measurement/test node***
  - perfSONAR
- ***Education & Engagement w/ End Users***

© 2013 Globus



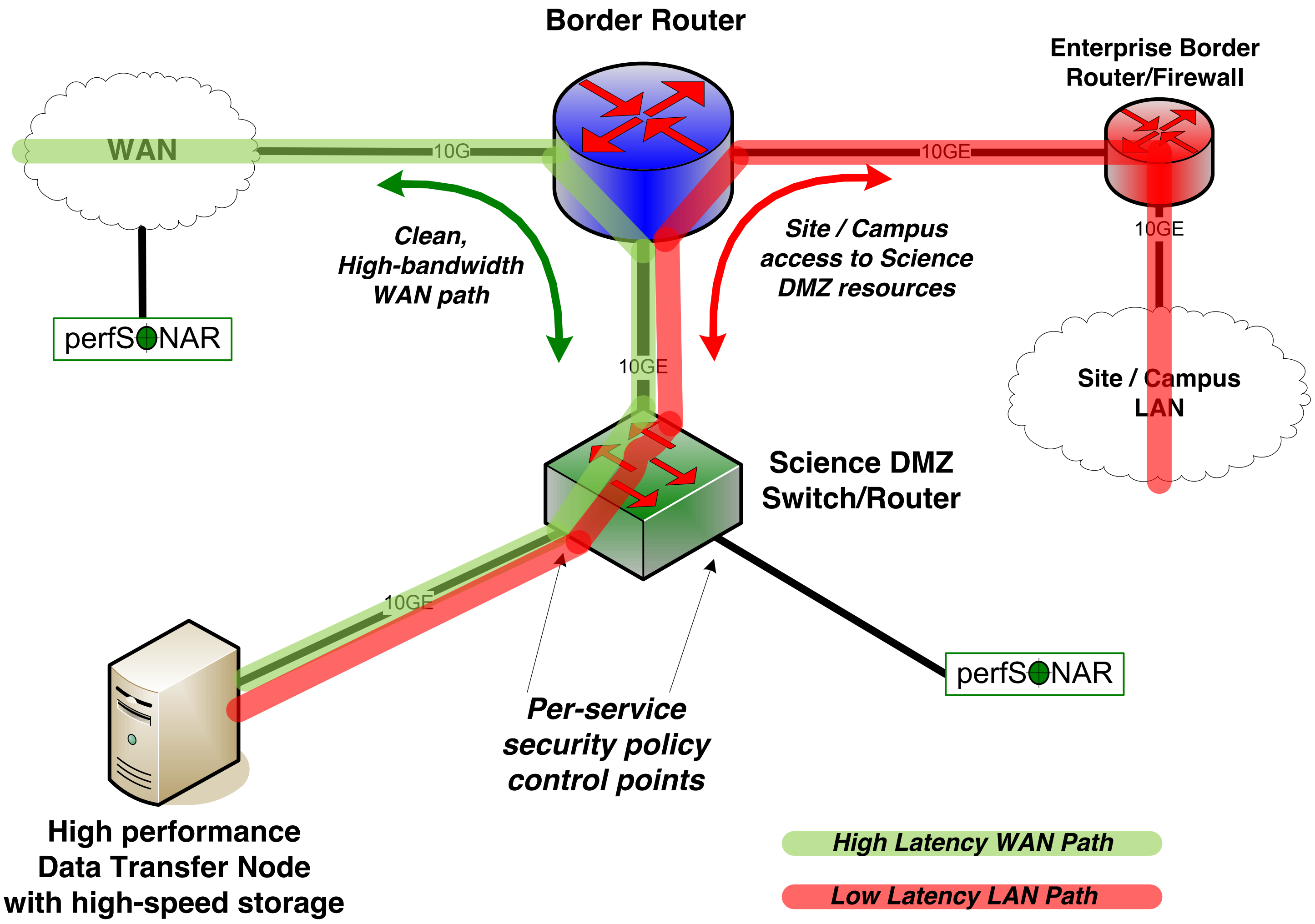
© 2013 Wikipedia

perfSONAR

Details at http://fasterdata.es.net/science-dmz/

ESnet

The Abstract Science DMZ


Border Router
Enterprise Border Router/Firewall
WAN
10G
10GE
Clean, High-bandwidth WAN path
Site / Campus access to Science DMZ resources
10GE
perfSONAR
10GE
Site / Campus LAN
Science DMZ Switch/Router
10GE
perfSONAR
Per-service security policy control points
High performance Data Transfer Node with high-speed storage
High Latency WAN Path
Low Latency LAN Path

But … It's Not Just the Network

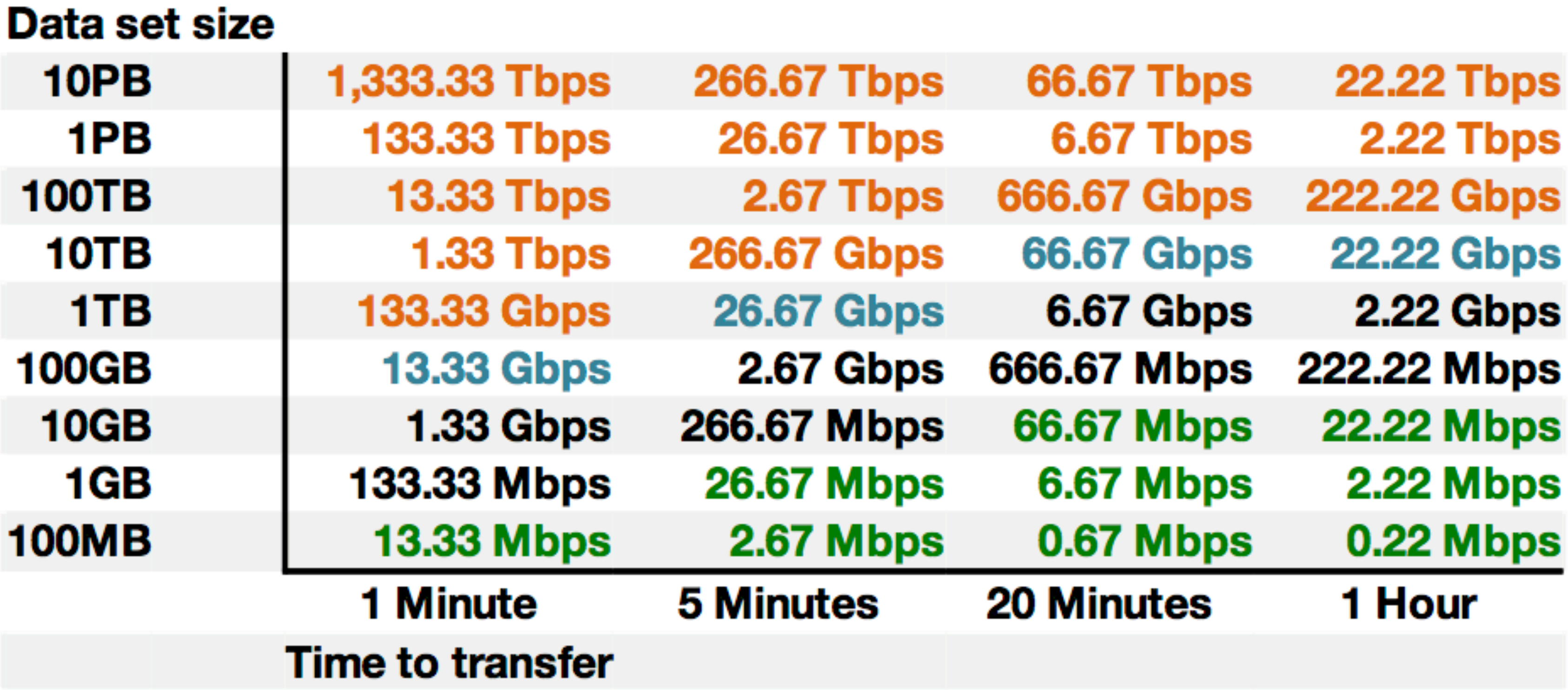
- Perhaps you are saying to yourself "I have no control over parts of my campus, let alone the 5 networks that sit between me and my collaborators"
  - Agree to disagree – network are like ogres, and ogres are like onions; both stink, and have layers*
  - Significant gains are possible in isolated areas of the OSI Stack
- Things "you" control:
  - Choice of data movement applications (say no to SCP and RSYNC)
  - Configuration of local gear (hosts, network devices)
  - Placement and configuration of diagnostic tools, e.g. perfSONAR
  - Use of the diagnostic tools
- Things that need some help:
  - Configuration of remote gear
  - Addressing issues when the diagnostic tools alarm
  - Getting someone to "care"

Google it

Outline

- Introduction & Motivation
- Network Support for Science
- Data Mobility Expectations & Realities
- Preparing the Campus
- Conclusions

Sample Data Transfer Rates



Data set size	1 Minute	5 Minutes	20 Minutes	1 Hour
10PB	1,333.33 Tbps	266.67 Tbps	66.67 Tbps	22.22 Tbps
1PB	133.33 Tbps	26.67 Tbps	6.67 Tbps	2.22 Tbps
100TB	13.33 Tbps	2.67 Tbps	666.67 Gbps	222.22 Gbps
10TB	1.33 Tbps	266.67 Gbps	66.67 Gbps	22.22 Gbps
1TB	133.33 Gbps	26.67 Gbps	6.67 Gbps	2.22 Gbps
100GB	13.33 Gbps	2.67 Gbps	666.67 Mbps	222.22 Mbps
10GB	1.33 Gbps	266.67 Mbps	66.67 Mbps	22.22 Mbps
1GB	133.33 Mbps	26.67 Mbps	6.67 Mbps	2.22 Mbps
100MB	13.33 Mbps	2.67 Mbps	0.67 Mbps	0.22 Mbps

Time to transfer

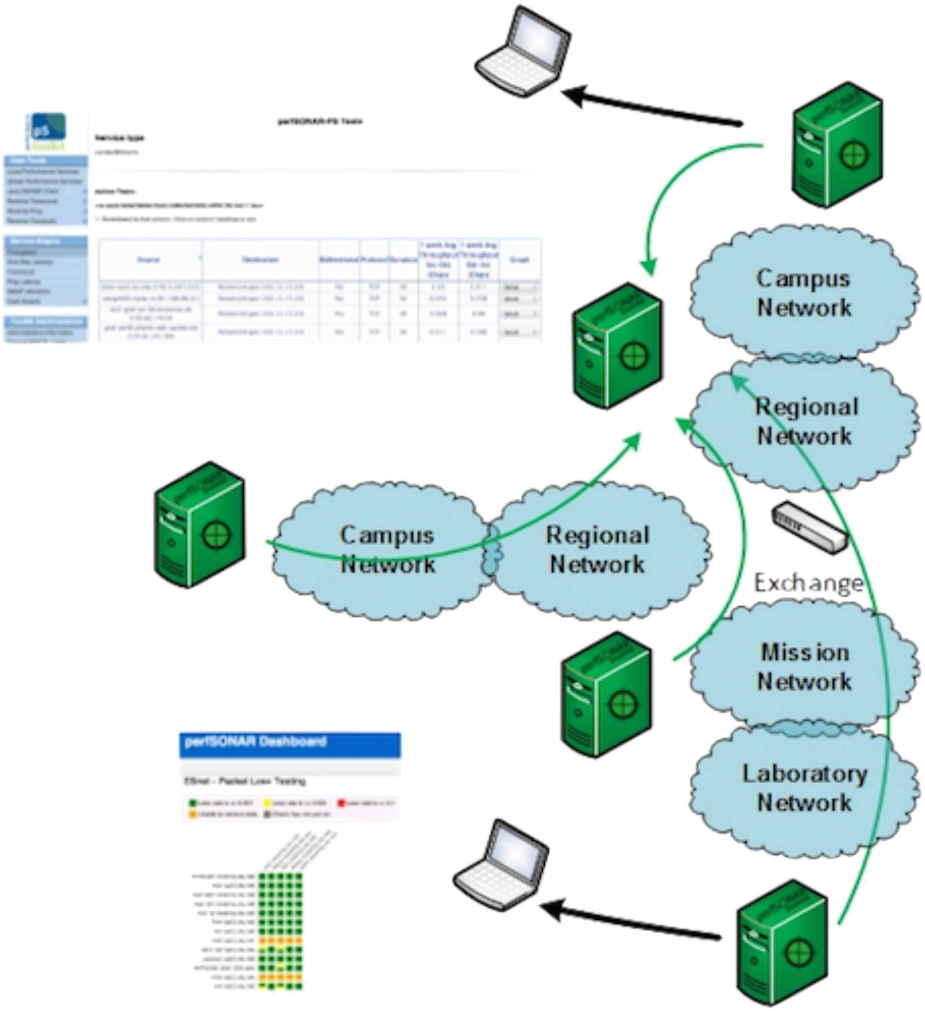
This table available at:

http://fasterdata.es.net/fasterdata-home/requirements-and-expectations/

Simulating Performance

- It's infeasible to perform at-scale data movement all the time – as we see in other forms of science, we need to rely on simulations
- Network performance comes down to a couple of key metrics:
  - Throughput (e.g. "how much can I get out of the network")
  - Latency (time it takes to get to/from a destination)
  - Packet loss/duplication/ordering (for some sampling of packets, do they all make it to the other side without serious abnormalities occurring?)
  - Network utilization (the opposite of "throughput" for a moment in time)
- We can get many of these from a selection of active and passive measurement tools – enter the perfSONAR Toolkit

Toolkit Use Case



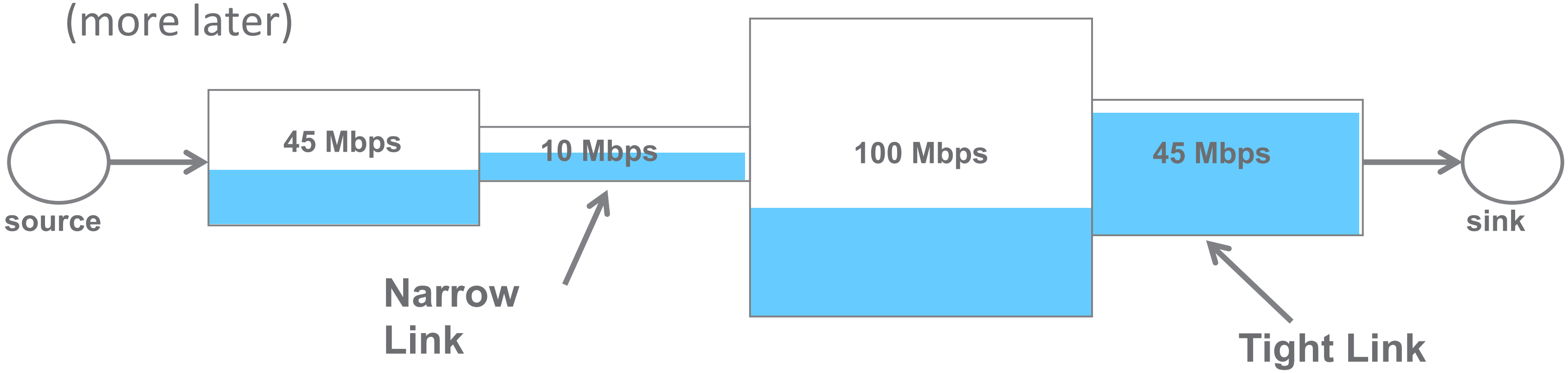
- The general use case is to establish some set of tests to other locations/ facilities
  - Sometimes you establish GUIs on top of this – XSEDE has one
- To answer the what/why questions:
  - Regular testing with select tools helps to establish patterns – how much bandwidth we would see during the course of the day – or when packet loss appears
  - We do this to ‘points of interest’ to see how well a real activity (e.g. Globus transfer) would do.
- If performance is ‘bad’, don’t expect much from the data movement tool

Its All About the Buffers

- A prequel – The Bandwidth Delay Product
  - The amount of “in flight” data allowed for a TCP connection (BDP = bandwidth * round trip time)
  - Example: 1Gb/s cross country, ~100ms
    - 1,000,000,000 b/s * .1 s = 100,000,000 bits
    - 100,000,000 / 8 = 12,500,000 bytes
    - 12,500,000 bytes / (1024*1024) ~ 12MB
  - Major OSs default to a base of 64k.
    - For those playing at home, the maximum throughput with a TCP window of 64 KByte for RTTs:
      - 10ms = 50Mbps
      - 50ms = 10Mbps
      - 100ms = 5Mbps
    - Autotuning does help by growing the window when needed. Do make this work properly, the host needs tuning: https://fasterdata.es.net/host-tuning/
- Ignore the math aspect, its really just about making sure there is memory to catch packets. As the speed increases, there are more packets. If there is not memory, we drop them, and that makes TCP sad.
  - Memory on hosts, and network gear

What BWCTL Tells Us

- Lets start by describing throughput, which is vague.
  - Capacity: link speed
    - Narrow Link: link with the lowest capacity along a path
    - Capacity of the end-to-end path = capacity of the narrow link
  - Utilized bandwidth: current traffic load
  - Available bandwidth: capacity – utilized bandwidth
    - Tight Link: link with the least available bandwidth in a path
  - Achievable bandwidth: includes protocol and host issues (e.g. BDP!)
- All of this is "memory to memory", e.g. we are not involving a spinning disk (more later)



(Shaded portion shows background traffic)

What BWCTL Tells Us

- BWCTL gives us a number – a number from the iperf2/iperf3/nuttcp tools

```
[zurawski@wash-pt1 ~]$ bwctl -T iperf -f m -t 10 -i 2 -c sunn-pt1.es.net
bwctl: 83 seconds until test results available

RECEIVER START
bwctl: exec_line: /usr/bin/iperf -B 198.129.254.58 -s -f m -m -p 5136 -t 10 -i 2.000000
bwctl: run_tool: tester: iperf
bwctl: run_tool: receiver: 198.129.254.58
bwctl: run_tool: sender: 198.124.238.34
bwctl: start_tool: 3598657357.738868
------------------------------------------------------------
Server listening on TCP port 5136
Binding to local address 198.129.254.58
TCP window size: 0.08 MByte (default)
------------------------------------------------------------
[ 16] local 198.129.254.58 port 5136 connected with 198.124.238.34 port 5136
[ ID] Interval       Transfer     Bandwidth
[ 16]  0.0- 2.0 sec  90.4 MBytes   379 Mbits/sec
[ 16]  2.0- 4.0 sec   689 MBytes  2891 Mbits/sec
[ 16]  4.0- 6.0 sec   684 MBytes  2867 Mbits/sec
[ 16]  6.0- 8.0 sec   691 MBytes  2897 Mbits/sec
[ 16]  8.0-10.0 sec   691 MBytes  2898 Mbits/sec
[ 16]  0.0-10.0 sec  2853 MBytes  2386 Mbits/sec
[ 16] MSS size 8948 bytes (MTU 8988 bytes, unknown interface)
bwctl: stop_tool: 3598657390.668028

RECEIVER END
```

N.B. This is what perfSONAR Graphs – the average of the complete test

What BWCTL Tells Us

- Iperf2 is not the tool you are looking for, hello iperf3

```
[zurawski@wash-pt1 ~]$ bwctl -T iperf3 -f m -t 10 -i 2 -c sunn-pt1.es.net
bwctl: 55 seconds until test results available

SENDER START
bwctl: run_tool: tester: iperf3
bwctl: run_tool: receiver: 198.129.254.58
bwctl: run_tool: sender: 198.124.238.34
bwctl: start_tool: 3598657653.219168
Test initialized
Running client
Connecting to host 198.129.254.58, port 5001
[ 17] local 198.124.238.34 port 34277 connected to 198.129.254.58 port 5001
[ ID] Interval           Transfer     Bandwidth       Retransmits
[ 17]   0.00-2.00   sec   430 MBytes  1.80 Gbits/sec  2
[ 17]   2.00-4.00   sec   680 MBytes  2.85 Gbits/sec  0
[ 17]   4.00-6.00   sec   669 MBytes  2.80 Gbits/sec  0
[ 17]   6.00-8.00   sec   670 MBytes  2.81 Gbits/sec  0
[ 17]   8.00-10.00  sec   680 MBytes  2.85 Gbits/sec  0
[ ID] Interval           Transfer     Bandwidth       Retransmits
      Sent
[ 17]   0.00-10.00  sec  3.06 GBytes  2.62 Gbits/sec  2
      Received
[ 17]   0.00-10.00  sec  3.06 GBytes  2.63 Gbits/sec

iperf Done.
bwctl: stop_tool: 3598657664.995604

SENDER END
```

N.B. This is what perfSONAR Graphs – the average of the complete test.

What BWCTL May Not be Telling Us

- Why kick iperf2 to the curb?
  - No notion of TCP retransmits – and you really want to have this to understand what is going on in a transfer (retransmits = a symptom of something dropping/corrupting/delaying packets)
  - CPU waster when you are doing UDP tests, e.g. it can't give you an accurate notion of network performance since it is host limited
  - Entering into non-supported territory (the best reason to switch)
- In general, there are other problems with a throughput tool we need to be concerned with – some are controllable and some aren't
  - Relies on the tuning of the host (e.g. did you follow http://fasterdata.es.net recommendations?)
  - Single number is not descriptive of what is really going on (e.g. was it 1Mbps because of my local host, local network, remote network, or remote host?)
  - Easy to test 'poorly' – lets get into that

What BWCTL May Not be Telling Us

- Fasterdata Tunings
  - Fasterdata recommends a set of tunings (https://fasterdata.es.net/host-tuning/) that are designed to increase the performance of a single COTS host, on a shared network infrastructure
  - What this means is that we don't recommend 'maximum' tuning
  - We are assuming (expecting? hoping?) the host can do parallel TCP streams via the data transfer application (e.g. Globus)
  - Because of that you don't want to assign upwards of 256M of kernel memory to a single TCP socket – a sensible amount is 32M/64M, and if you have 4 streams you are getting the benefits of 128M/256M (enough for a 10G cross country flow)
  - We also strive for good citizenship – its very possible for a single 10G machine to get 9.9Gbps TCP, we see this often. If its on a shared infrastructure, there is benefit to downtuning buffers.
- Can you ignore the above? Sure – overtune as you see fit, ***KNOW YOUR NETWORK, USERS, AND USE CASES***
- What does this do to perfSONAR testing?

What BWCTL May Not be Telling Us

- Regular Testing Setup
  - If we don't 'max tune', and run a 20/30 second single streamed TCP test (defaults for the toolkit) we are not going to see 9.9Gbps.
  - Think critically: TCP ramp up takes 1-5 seconds (depending on latency), and any tiny blip of congestion will cut TCP performance in half.
  - It is ***common*** (and in my mind - expected) to see regular testing values on clean networks range between 1Gbps and 5Gbps, latency dependent
  - Performance has two ranges – really crappy, and expected (where expected has a lot of headroom). You will know when its really crappy (trust me).
- Diagnostic Suggestions
  - You can max out BWCTL in this capacity
  - Run long tests (-T 60), with multiple streams (-P 4), and large windows (-W 128M); go crazy
  - It is also ***VERY COMMON*** that doing so will produce different results than your regular testing. It's a different set of test parameters, its not that the tools are deliberately lying.

What Happens When BWCTL Says "Crappy"

- Science does not live by throughput alone – mainly because if its low you need to understand why.

```
[zurawski@wash-pt1 ~]$ bwctl -T nuttcp -f m -t 10 -i 2 -c sunn-pt1.es.net
bwctl: 41 seconds until test results available

SENDER START
bwctl: exec_line: /usr/bin/nuttcp -vv -p 5004 -i 2.000000 -T 10 -t 198.129.254.58
bwctl: run_tool: tester: nuttcp
bwctl: run_tool: receiver: 198.129.254.58
bwctl: run_tool: sender: 198.124.238.34
bwctl: start_tool: 3598658394.807831
nuttcp-t: v7.1.6: socket
nuttcp-t: buflen=65536, nstream=1, port=5004 tcp -> 198.129.254.58
nuttcp-t: time limit = 10.00 seconds
nuttcp-t: connect to 198.129.254.58 with mss=8948, RTT=62.440 ms
nuttcp-t: send window size = 98720, receive window size = 87380
nuttcp-t: available send window = 74040, available receive window = 65535
nuttcp-r: v7.1.6: socket
nuttcp-r: buflen=65536, nstream=1, port=5004 tcp
nuttcp-r: interval reporting every 2.00 seconds
nuttcp-r: accept from 198.124.238.34
nuttcp-r: send window size = 98720, receive window size = 87380
nuttcp-r: available send window = 74040, available receive window = 65535
    6.3125 MB /   2.00 sec =   26.4759 Mbps     27 retrans
    3.5625 MB /   2.00 sec =   14.9423 Mbps      4 retrans
    3.8125 MB /   2.00 sec =   15.9906 Mbps      7 retrans
    4.8125 MB /   2.00 sec =   20.1853 Mbps     13 retrans
    6.0000 MB /   2.00 sec =   25.1659 Mbps      7 retrans
nuttcp-t: 25.5066 MB in 10.00 real seconds = 2611.85 KB/sec = 21.3963 Mbps
nuttcp-t: 25.5066 MB in 0.01 CPU seconds = 1741480.37 KB/cpu sec
nuttcp-t: retrans = 58
nuttcp-t: 409 I/O calls, msec/call = 25.04, calls/sec = 40.90
nuttcp-t: 0.0user 0.0sys 0:10real 0% 0i+0d 768maxrss 0+2pf 51+3csw

nuttcp-r: 25.5066 MB in 10.30 real seconds = 2537.03 KB/sec = 20.7833 Mbps
nuttcp-r: 25.5066 MB in 0.02 CPU seconds = 1044874.29 KB/cpu sec
nuttcp-r: 787 I/O calls, msec/call = 13.40, calls/sec = 76.44
nuttcp-r: 0.0user 0.0sys 0:10real 0% 0i+0d 770maxrss 0+4pf 382+0csw
bwctl: stop_tool: 3598658417.214024
```

N.B. This is what perfSONAR Graphs – the average of the complete test.

What OWAMP Tells Us

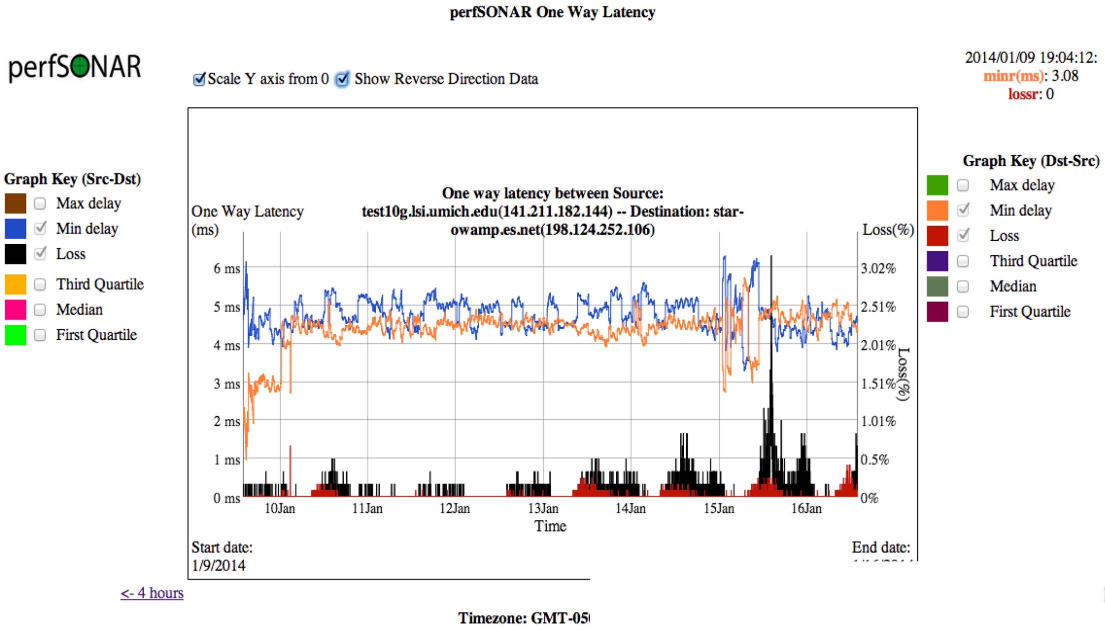
- OWAMP is designed to tell us when small packets (~50B in size, UDP based) have perturbation when sent end to end.

```
[zurawski@wash-owamp ~]$ owping sunn-owamp.es.net
Approximately 12.6 seconds until results available

--- owping statistics from [wash-owamp.es.net]:8852 to [sunn-owamp.es.net]:8837 ---
SID:  c681fe4ed67f1f0908224c341a2b83f3
first:     2014-01-13T18:27:22.032
last: 2014-01-13T18:27:32.904
100 sent, 12 lost (12.000%), 0 duplicates
one-way delay min/median/max = 31.1/31.1/31.3 ms, (err=0.00502 ms)
one-way jitter = nan ms (P95-P50)
Hops = 7 (consistently)
no reordering


--- owping statistics from [sunn-owamp.es.net]:9182 to [wash-owamp.es.net]:8893 ---
SID:  c67cfc7ed67f1f09531c87cf38381bb6
first:     2014-01-13T18:27:21.993
last: 2014-01-13T18:27:33.785
100 sent, 0 lost (0.000%), 0 duplicates
one-way delay min/median/max = 31.4/31.5/31.5 ms, (err=0.00502 ms)
one-way jitter = 0 ms (P95-P50)
Hops = 7 (consistently)
no reordering
```

What OWAMP Tells Us

- OWAMP is a necessity in regular testing – if you aren't using this you need to be
  - Queuing often occurs in a single direction (think what everyone is doing at noon on a college campus)
  - Packet loss (and how often/how much occurs over time) is more valuable than throughput
  - If your router is going to drop a 50B UDP packet, it is most certainly going to drop a 15000B/9000B TCP packet
- Overlaying data
  - Compare your throughput results against your OWAMP – do you see patterns?
  - Alarm on each, if you are alarming (and we hope you are alarming …)

What OWAMP Tells Us



perfSONAR One Way Latency

perfSONAR

☑ Scale Y axis from 0 ☑ Show Reverse Direction Data

2014/01/09 19:04:12:
minr(ms): 3.08
lossr: 0

One way latency between Source: test10g.lsi.umich.edu(141.211.182.144) -- Destination: star-owamp.es.net(198.124.252.106)

Start date:
1/9/2014

End date:

Graph Key (Src-Dst)

🟫	□	Max delay
🟦	☑	Min delay
⬛	☑	Loss
🟨	□	Third Quartile
🟥	□	Median
🟩	□	First Quartile

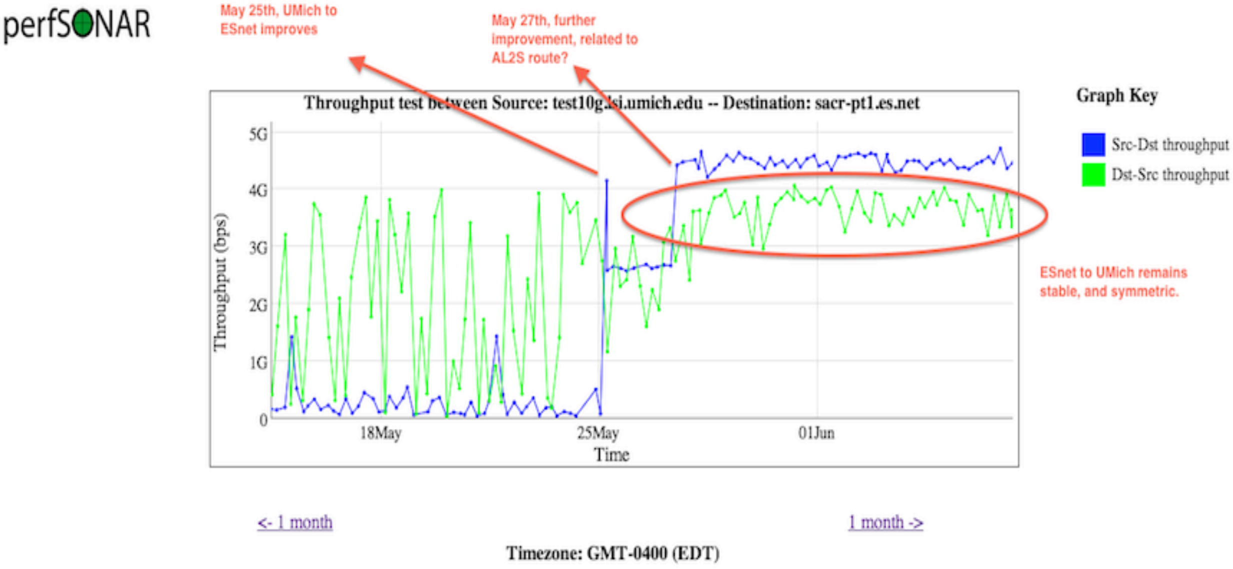
Graph Key (Dst-Src)

🟩	□	Max delay
🟧	☑	Min delay
🟥	☑	Loss
🟪	□	Third Quartile
🟩	□	Median
🟥	□	First Quartile

<- 4 hours

Timezone: GMT-05

perfSONAR BWCTL Graph



perfSONAR

May 25th, UMich to ESnet improves

May 27th, further improvement, related to AL2S route?

ESnet to UMich remains stable, and symmetric.

Throughput test between Source: test10g.lsi.umich.edu -- Destination: sacr-pt1.es.net

Graph Key

🟦	Src-Dst throughput
🟩	Dst-Src throughput

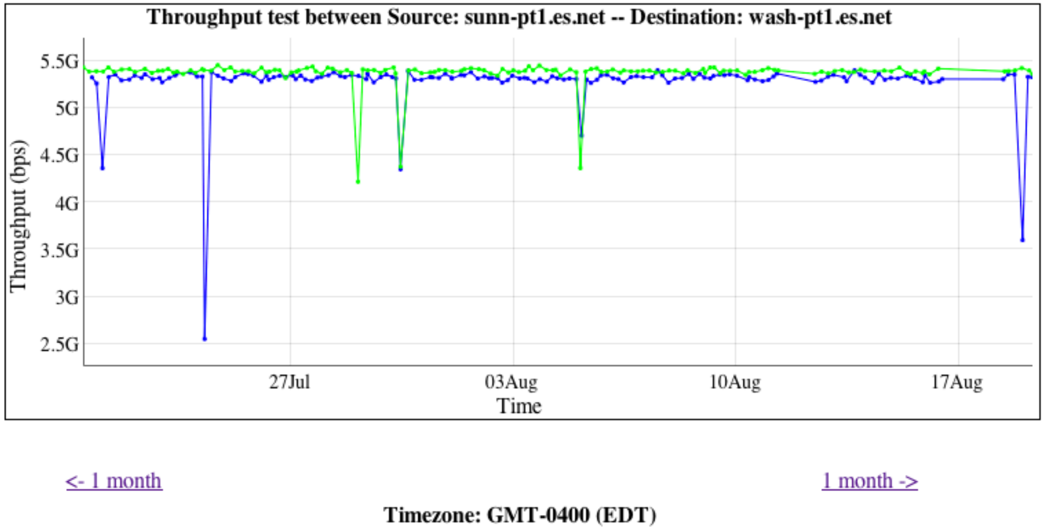
<- 1 month

1 month ->

Timezone: GMT-0400 (EDT)

Common Pitfalls – “it should be higher!”

- There have been some expectation management problems with the tools that we have seen (in XSEDE and elsewhere)
  - Some feel that if they have 10G, they will get all of it
  - Some may not understand the makeup of the test
  - Some may not know what they should be getting
- Lets start with an ESnet to ESnet test, between very well tuned and recent pieces of hardware
- 5Gbps is “awesome” for:
  - A 20 second test
  - 60ms Latency
  - ***Homogenous*** servers
  - Using fasterdata tunings
  - On a shared infrastructure



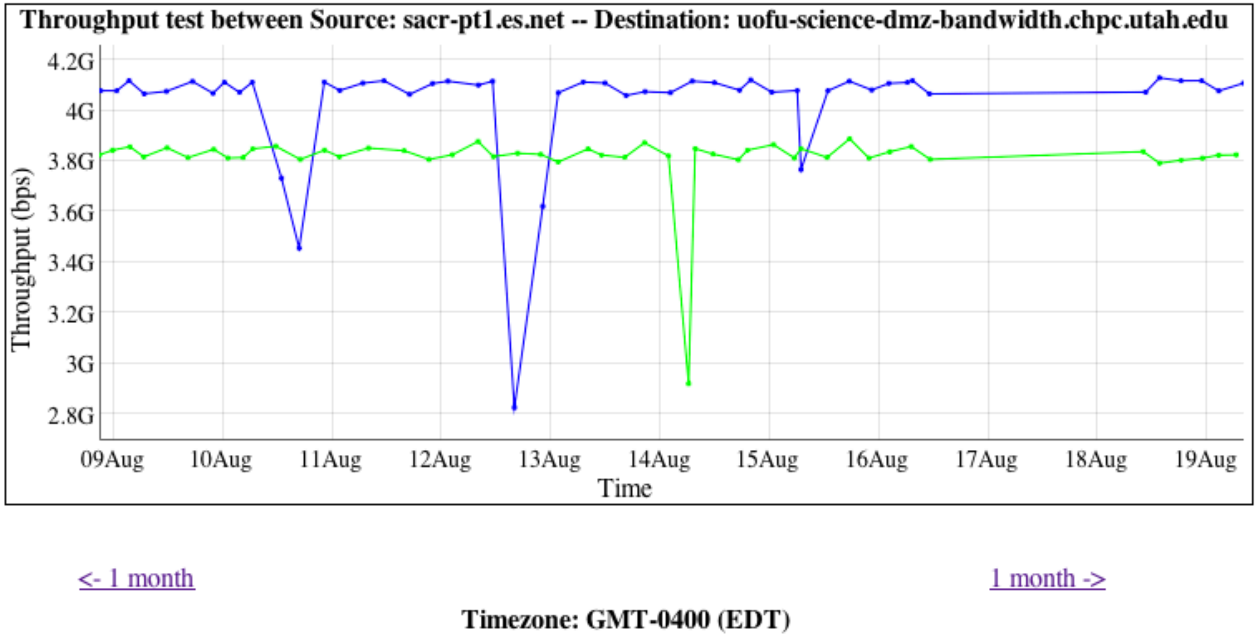
<- 1 month

1 month ->

Timezone: GMT-0400 (EDT)

Common Pitfalls – "it should be higher!"

- Another example, ESnet (Sacremento CA) to Utah, ~20ms of latency

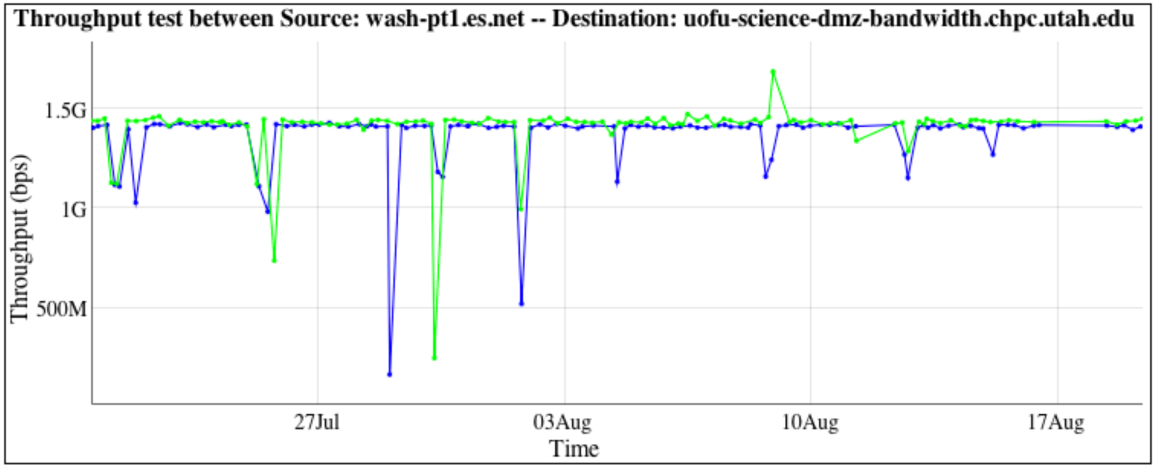


Timezone: GMT-0400 (EDT)

- Is it 5Gbps? No, but still outstanding given the environment:
  - 20 second test
  - ***Heterogeneous*** hosts
  - Possibly different configurations (e.g. similar tunings of the OS, but not exact in terms of things like BIOS, NIC, etc.)
  - Different congestion levels on the ends

Common Pitfalls – "it should be higher!"

- Similar example, ESnet (Washington DC) to Utah, ~50ms of latency



<- 1 month

1 month ->

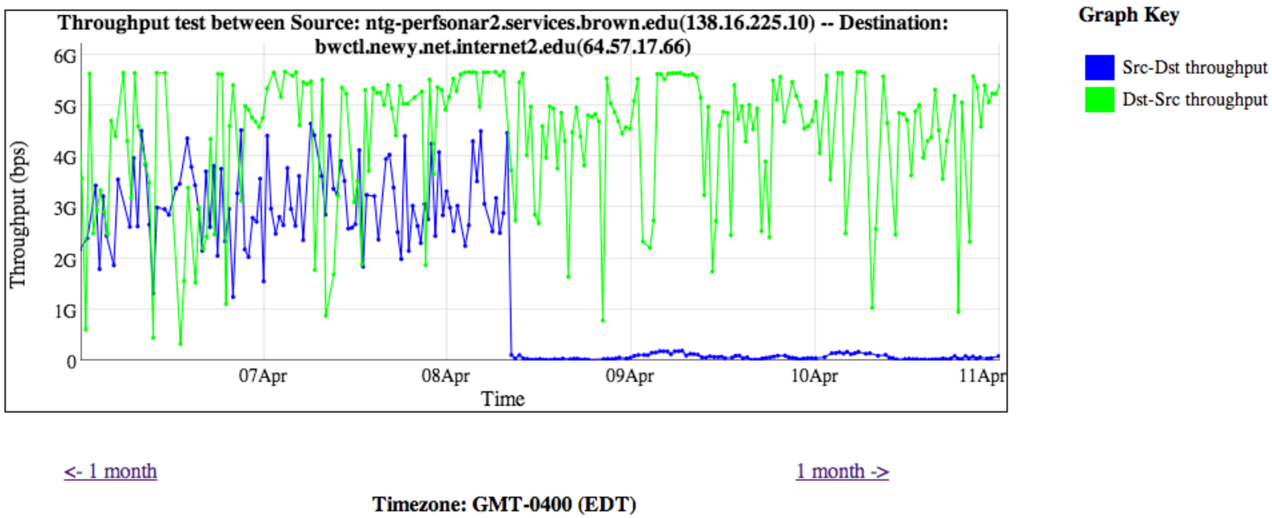
Timezone: GMT-0400 (EDT)

- Is it 5Gbps? No. Should it be? No! Could it be higher? Sure, run a different diagnostic test.
  - Longer latency – still same length of test (20 sec)
  - ***Heterogeneous*** hosts
  - Possibly different configurations (e.g. similar tunings of the OS, but not exact in terms of things like BIOS, NIC, etc.)
  - Different congestion levels on the ends
- Takeaway – you will know bad performance when you see it. This is consistent and jives with the environment.

ESnet

Common Pitfalls – "it should be higher!"

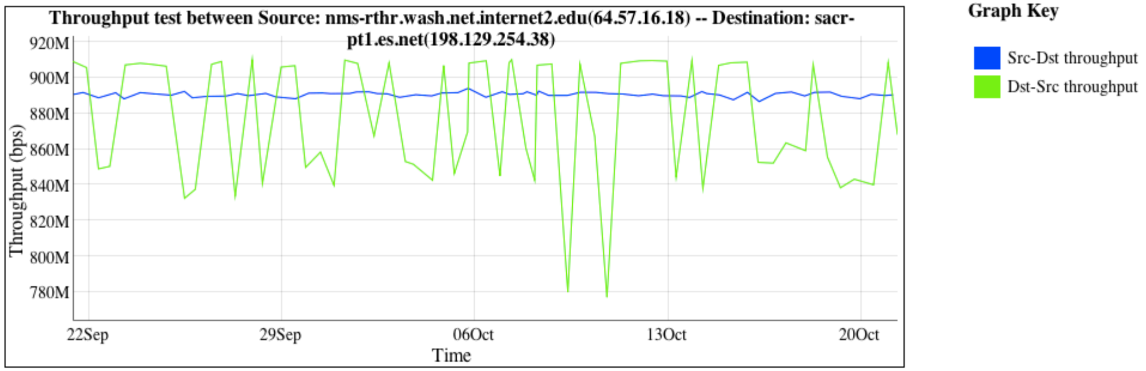
- Another Example – the 1st half of the graph is perfectly normal
  - Latency of 10-20ms (TCP needs time to ramp up)
  - Machine placed in network core of one of the networks – congestion is a fact of life
  - Single stream TCP for 20 seconds
- The 2nd half is not (e.g. packet loss caused a precipitous drop)



- ***You will know it, when you see it.***

Common Pitfalls – “the tool is unpredictable”

- Sometimes this happens:



<- 1 month

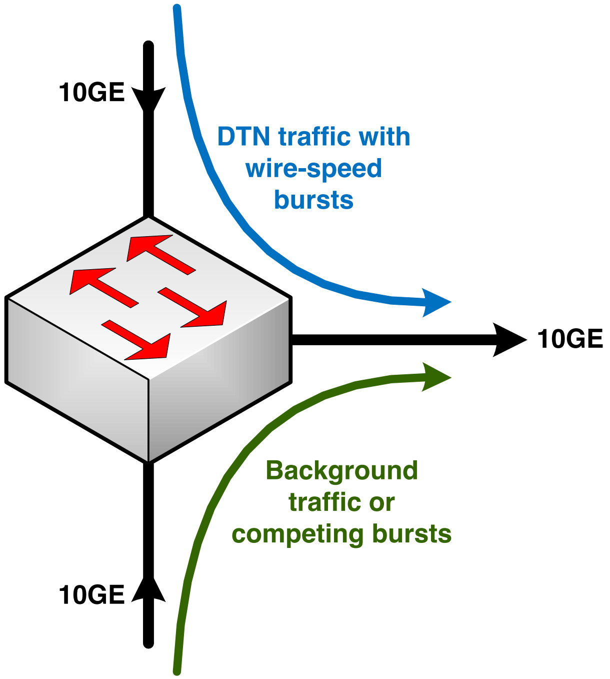
1 month ->

Timezone: GMT-0400 (EDT)

- Is it a “problem”? Yes and no.
- Cause: this is called “overdriving” and is common. A 10G host and a 1G host are testing to each other
  - 1G to 10G is smooth and expected (~900Mbps, Blue)
  - 10G to 1G is choppy (variable between 900Mbps and 700Mbps, Green)

Common Pitfalls – "the tool is unpredictable"

- A NIC doesn't stream packets out at some average rate - it's a binary operation:
  - Send (e.g. @ max rate) vs. not send (e.g. nothing)
- 10G of traffic needs buffering to support it along the path. A 10G switch/router can handle it. So could another 10G host (if both are tuned of course)
- A 1G NIC is designed to hold bursts of 1G. Sure, they can be tuned to expect more, but may not have enough physical memory
  - Ditto for switches in the path
- At some point things 'downstep' to a slower speed, that drops packets on the ground, and TCP reacts like it were any other loss event.


10GE
DTN traffic with wire-speed bursts
10GE
Background traffic or competing bursts
10GE

Common Pitfalls – "GridFTP is much worse than BWCTL!"

- And now we come to our frienemy, the disk
- perfSONAR tests are memory to memory for a reason:
  - Remove the host from the equation as much as we can
  - Unify tunings of the OS and tools
  - May sometimes need to get picky about the BIOS, motherboard, system bus, NIC and driver – but a good baseline is possible without all that
- Learning to use disks correctly:
  - You ***DO**** need to care about tunings, all the way down
  - Way too much to describe here, reading material: https://fasterdata.es.net/science-dmz/DTN/
  - In general, you need to worry about performance per spindle, you will learn to care about things like RAID to stripe data, and the RAID card performance to ensure it streams off the device to the hardware as fast as possible.
  - Realities from ESnet reference implementation:
    - memory to memory, 1 10GE NIC: 9.9 Gbps
    - disk to disk: 9.6 Gbps (1.2 GBytes/sec) using large files on all 3 disk partitions in parallel

Common Pitfalls – Summary

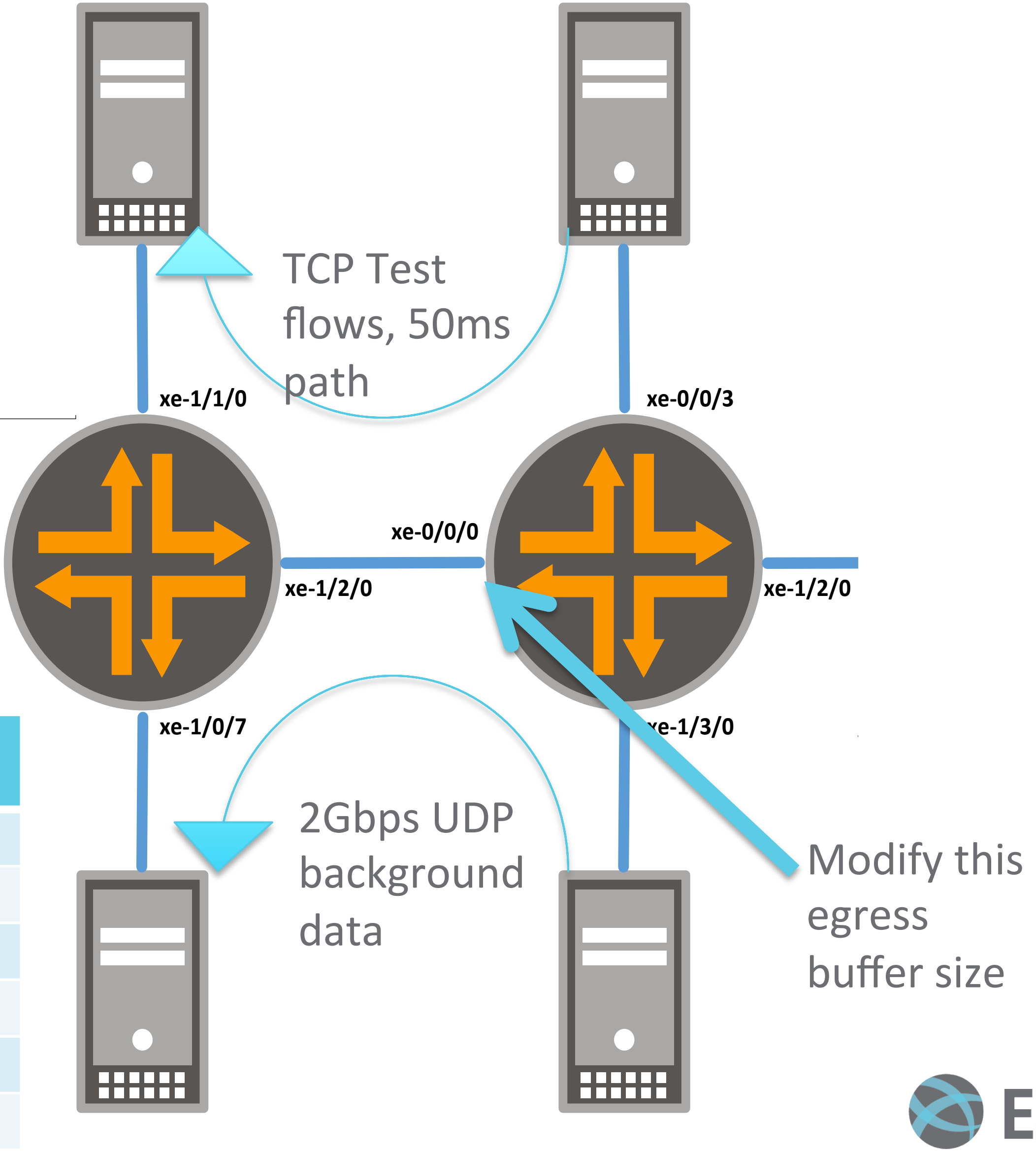
- When in doubt – test again!
  - Diagnostic tests are informative – and they should provide more insight into the regular stuff (still do regular testing, of course)
  - Be prepared to divide up a path as need be
- A poor carpenter blames his tools
  - The tools are only as good as the people using them, do it methodically
  - Trust the results – remember that they are giving you a number based on the entire environment
- If the site isn't using perfSONAR – step 1 is to get them to do so
  - http://www.perfsonar.net
- Get some help
  - To quote Blondie, "Call me, call me any, anytime"
  - engage@es.net

Outline

- Introduction & Motivation
- Network Support for Science
- Data Mobility Expectations & Realities
- Preparing the Campus
- Conclusions

ESnet

Small Bufffer = Science FAIL

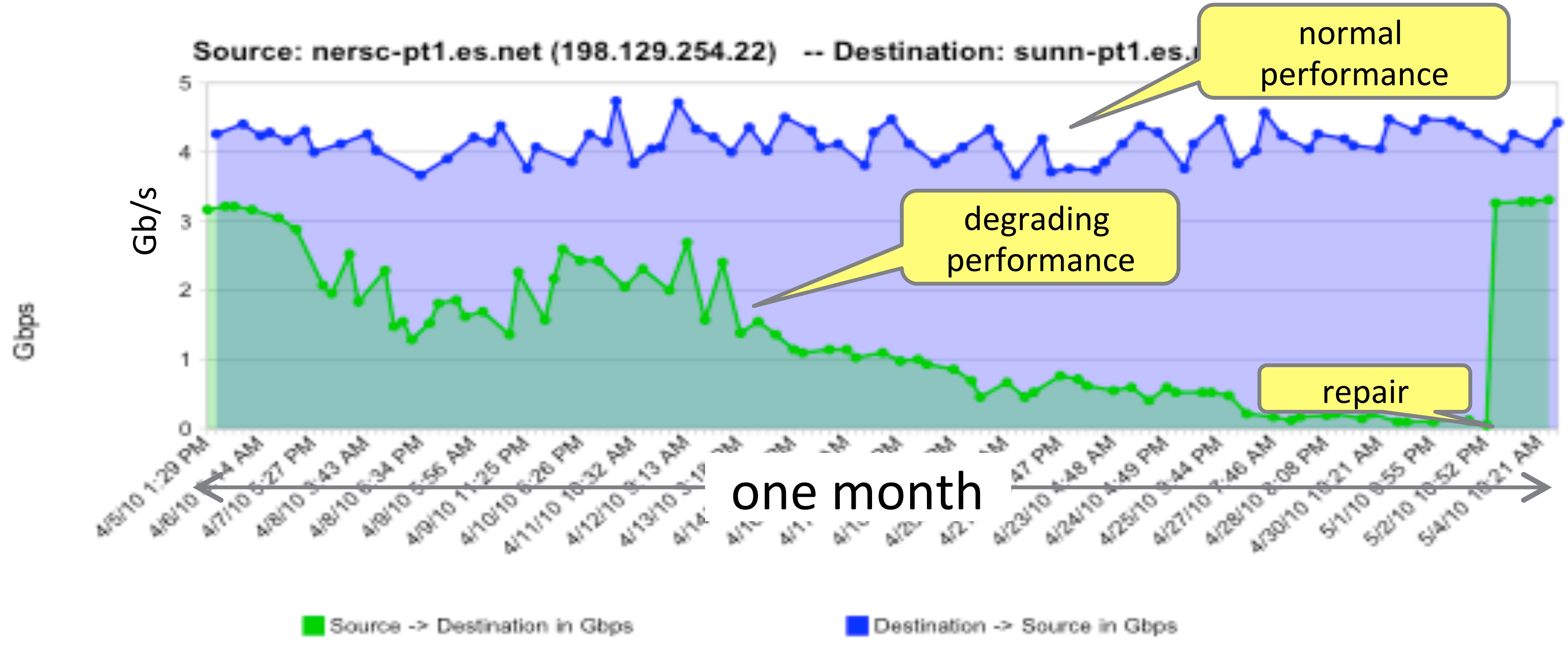

TCP Test flows, 50ms path
xe-1/1/0
xe-0/0/3
xe-0/0/0
xe-1/2/0
xe-1/2/0
xe-1/0/7
xe-1/3/0
2Gbps UDP background data
Modify this egress buffer size

30 Second test, 2 TCP streams

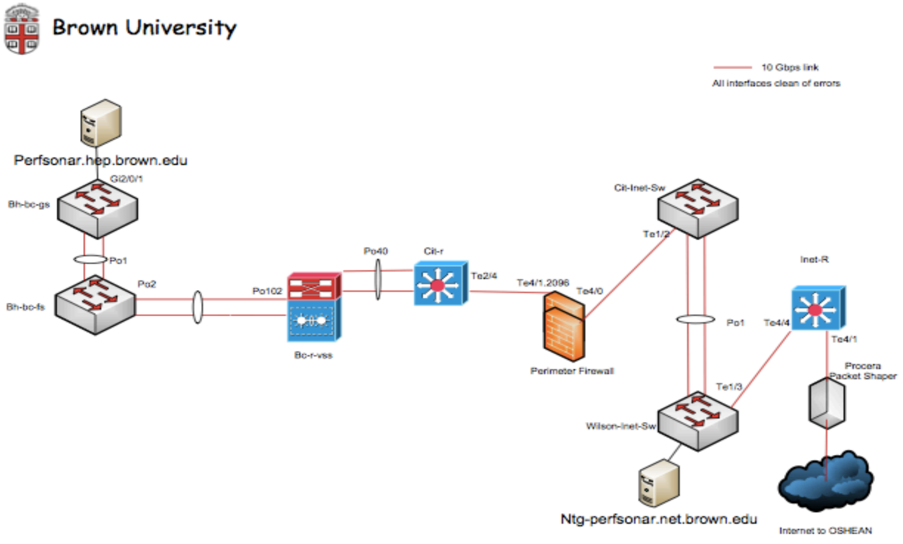
Buffer Size	Packets Dropped	TCP Throughput
120 MB	0	8Gbps
60 MB	0	8Gbps
36 MB	200	2Gbps
24 MB	205	2Gbps
12 MB	204	2Gbps
6 MB	207	2Gbps

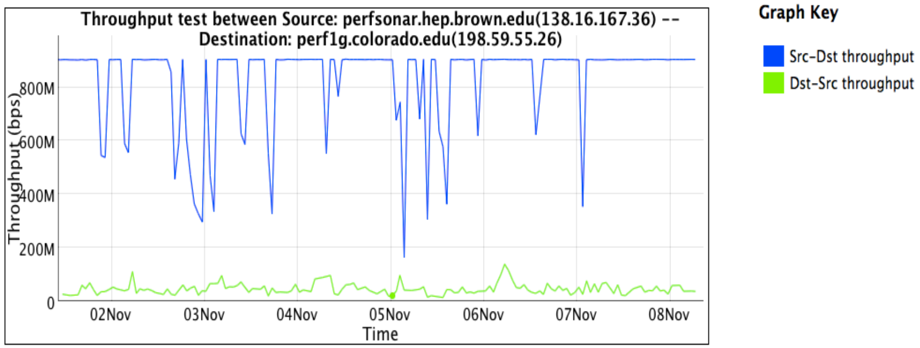
Infrastructure FAIL = Science FAIL

- perfSONAR is designed to pinpoint and identify soft failures to accelerate resolution.
- Example: Find and replace failing optics

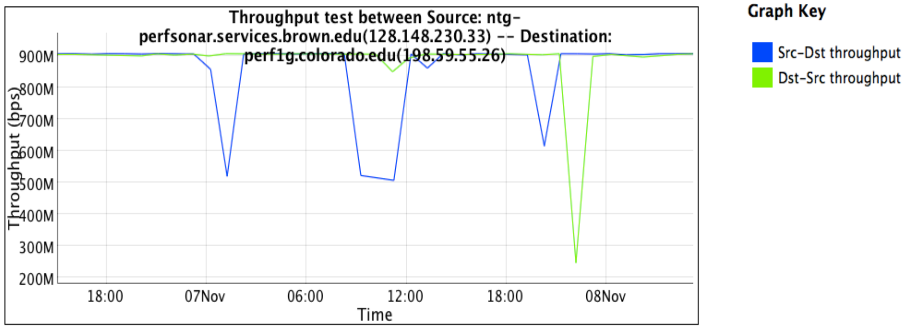


Overzealous Security = Science FAIL

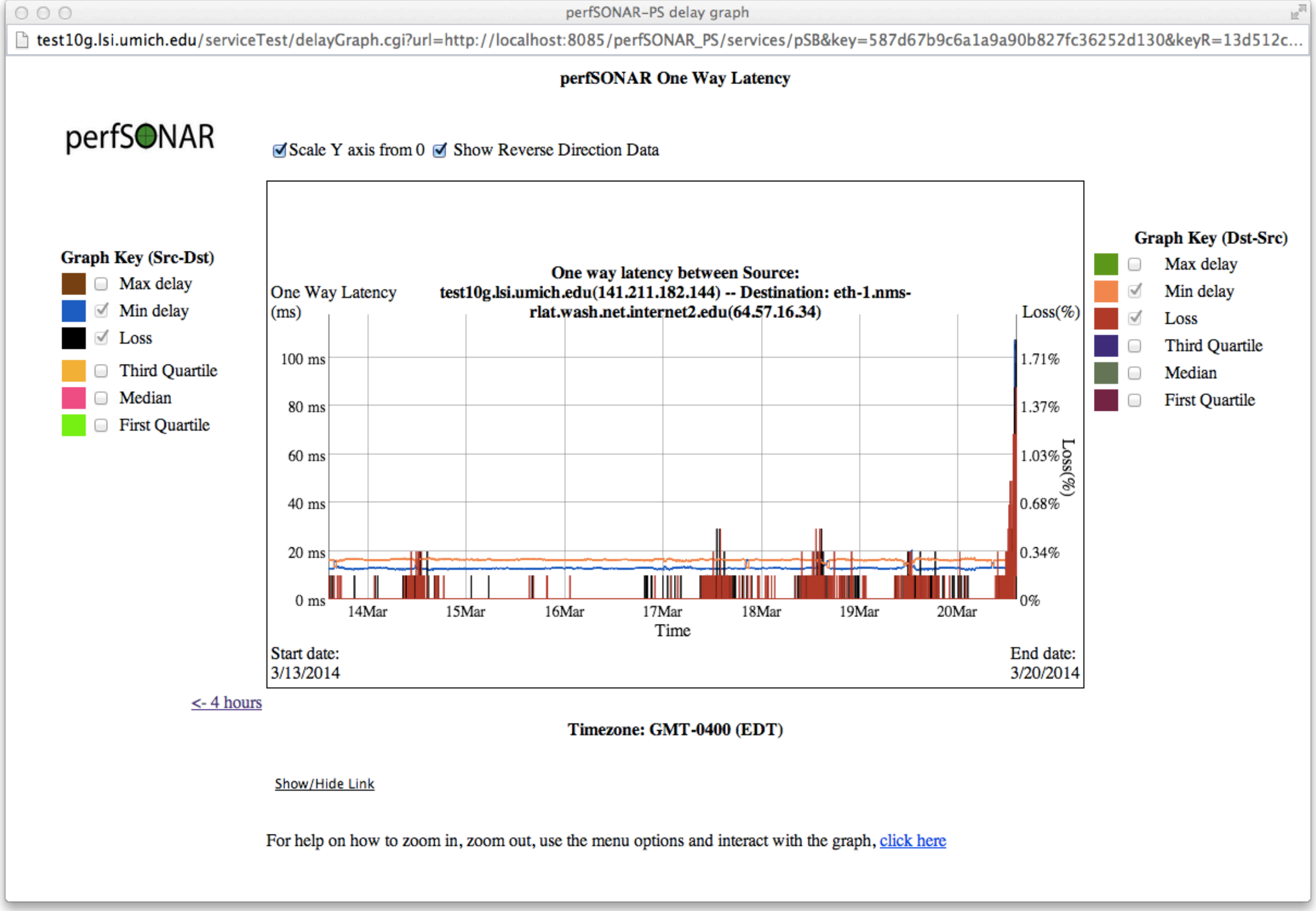



Throughput test between Source: perfsonar.hep.brown.edu(138.16.167.36) -- Destination: perf1g.colorado.edu(198.59.55.26)

Behind vs. in front of the firewall. Performance differences of an order of magnitude have a way of catching the attention of management.


Throughput test between Source: ntg-perfsonar.services.brown.edu(128.148.230.33) -- Destination: perf1g.colorado.edu(198.59.55.26)

Congestion = Science FAIL



perfSONAR-PS delay graph

test10g.lsi.umich.edu/serviceTest/delayGraph.cgi?url=http://localhost:8085/perfSONAR_PS/services/pSB&key=587d67b9c6a1a9a0b827fc36252d130&keyR=13d512c...

perfSONAR One Way Latency

☑ Scale Y axis from 0 ☑ Show Reverse Direction Data

Graph Key (Src-Dst)

🟫 □ Max delay
🟦 ☑ Min delay
⬛ ☑ Loss
🟧 □ Third Quartile
🟥 □ Median
🟩 □ First Quartile

Graph Key (Dst-Src)

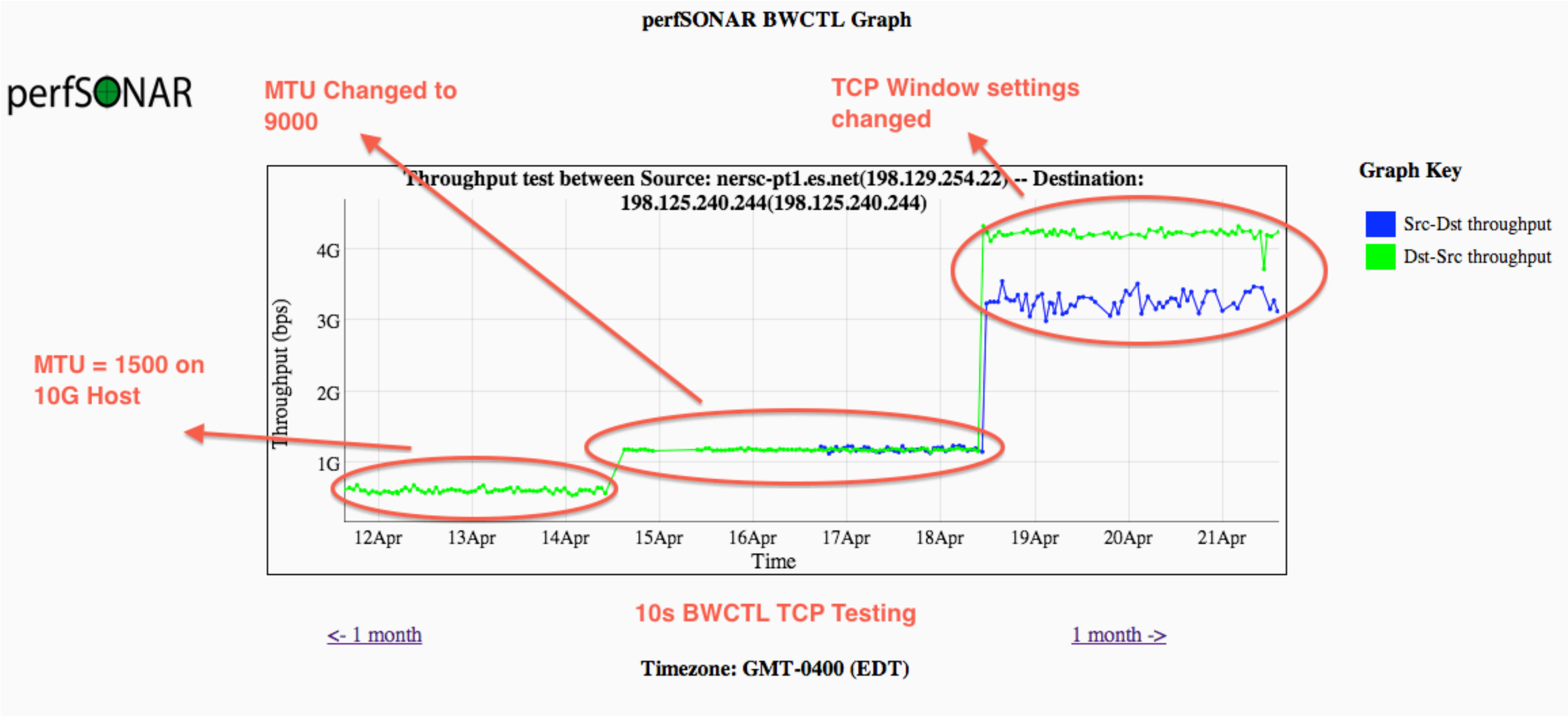
🟩 □ Max delay
🟧 ☑ Min delay
🟥 ☑ Loss
🟪 □ Third Quartile
🟩 □ Median
🟥 □ First Quartile

Timezone: GMT-0400 (EDT)

For help on how to zoom in, zoom out, use the menu options and interact with the graph, click here

- Networks are designed to be used after all – unless you engineer capacity and respect use cases

Turning a FAILs into WINs



perfSONAR BWCTL Graph

MTU = 1500 on 10G Host

MTU Changed to 9000

TCP Window settings changed

10s BWCTL TCP Testing

<- 1 month

1 month ->

Timezone: GMT-0400 (EDT)

- 12 Step programs encourage you to admit your problems – and then work toward a solution
  - Tuning the network for science takes time

Outline

- Introduction & Motivation
- Network Support for Science
- Data Mobility Expectations & Realities
- Preparing the Campus
- Conclusions

Summary

- Data mobility problems are not going to go away
  - It is unlikely everyone will have a particle accelerator, a supercomputer, EBs of storage, and 100s of researchers all within 10ms of each other
- Capacity will increase, but is meaningless when problems exist
- Insight into the network comes from tools like perfSONAR – if you aren't using it, start. If you need help, ask.
  - If its useful to XSEDE – ask for more ☺
- To combat this:
  - We need to identify users and use cases
  - We need to clean up networks to support science – while still offering all the CIA components
  - We need to move toward the proper tools and procedures to accomplish goals
- This is a team effort – because you cannot fix network performance in a vacuum

The "Data Champion"

- I propose the creation of a new focus for XSEDE to go hand and hand with the campus champion: the "Data Champion"
  - A Lorax for Science (Scorax?) – someone to speak for the bits
- Basic idea:
  - Someone who knows the needs and workflows of the campus users
    - Doesn't ask "how much they need", asks "what do you need to do"
  - Someone who can translate the needs into a requirement for the campus IT/Regional IT/National IT powers
    - Translates "transfer light source data" into actionable network engineering considerations
  - Someone who is there to listen, and step in, when there is a problem
    - Coordinates with other engagement efforts at ESnet, regional networks, etc.
  - Stays relevant in technology and training (http://oinworkshop.com)
  - Stays connected (e.g. human networking) with the parties that are on the critical path to success. XSEDE to XSEDE happens, but so does XSEDE to DOE, etc.
    - Its all connected, and we need to make sure it all works
- Who is with me on this? Who can make this happen at XSEDE?

Conclusion

- Science is good, growing, and changing. Adapt with these factors
- Science support doesn't need to be hard
- For those that design the networks – consider the Science DMZ Approach for network architecture and security posture
- For those that use the networks – consider the Science DMZ approach for data movement hosts and tools
- Monitoring matters to everyone
- A little knowledge goes a long way – learn to understand the tools, and ask when you don't understand.

ESnet
ENERGY SCIENCES NETWORK

FAIL-transfer: Removing the Mystery of Network Performance from Scientific Data Movement

Jason Zurawski – zurawski@es.net

Science Engagement Engineer, ESnet

Lawrence Berkeley National Laboratory

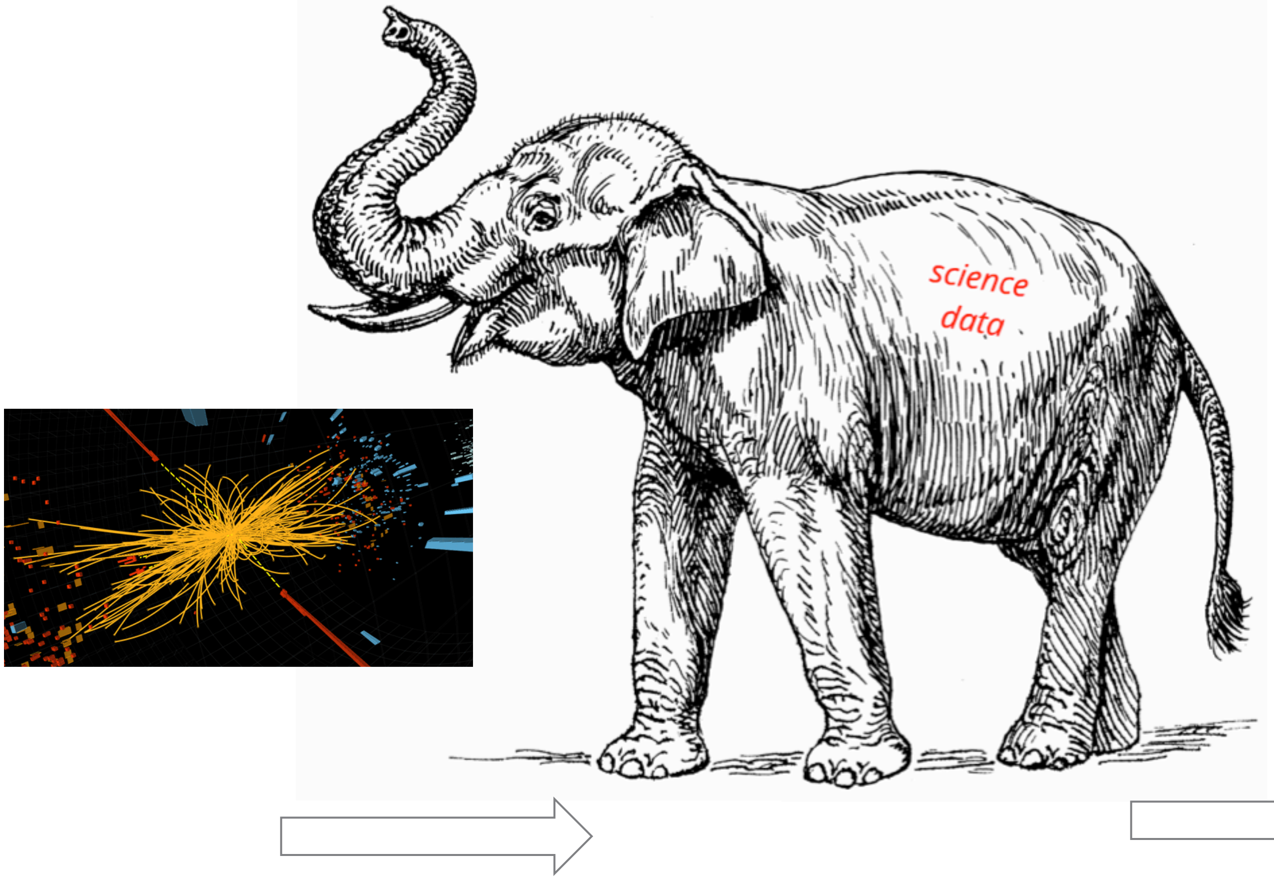
XSEDE Campus Champions Webinar
August 20th 2014

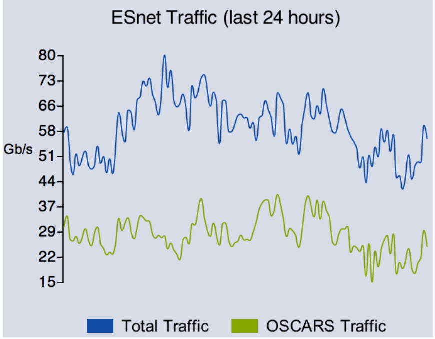
U.S. DEPARTMENT OF ENERGY
Office of Science

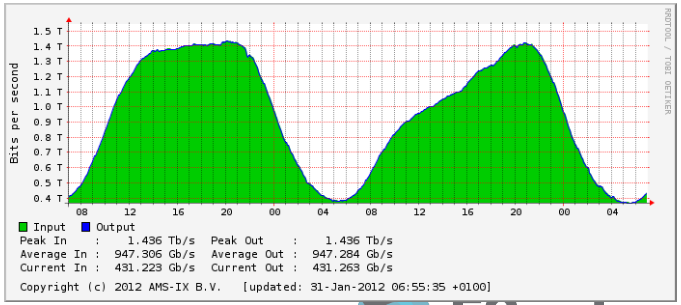
BERKELEY LAB

Extra Material

ESnet != Commercial Internet

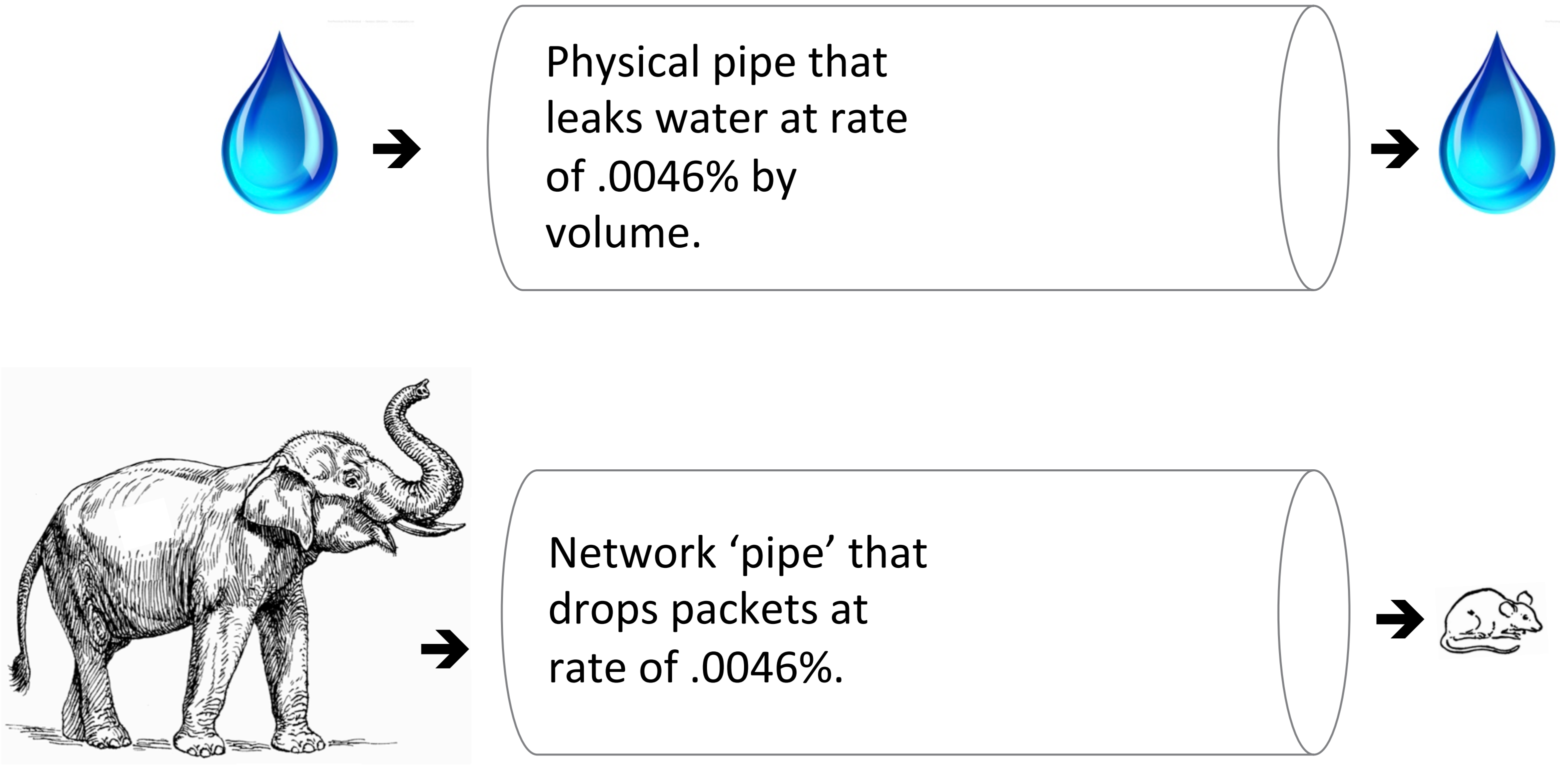






```
Peak In     :   1.436 Tb/s   Peak Out     :   1.436 Tb/s
Average In  : 947.306 Gb/s   Average Out  : 947.284 Gb/s
Current In  : 431.223 Gb/s   Current Out  : 431.263 Gb/s

Copyright (c) 2012 AMS-IX B.V.   [updated: 31-Jan-2012 06:55:35 +0100]
```

Elephant Flows Place Great Demands on Networks



Physical pipe that leaks water at rate of .0046% by volume.

Network ‘pipe’ that drops packets at rate of .0046%.

<u>Result</u>
99.9954% of water transferred, at “line rate.”

<u>Result</u>
100% of data transferred, *slowly*, at <<5% optimal speed.

essentially fixed

determined by speed of light

$$\frac{\text{maximum segment size}}{\text{round-trip time}} \times \frac{1}{\sqrt{\text{packet-loss rate}}}$$

Through careful engineering, we can minimize packet loss.

Assumptions: 10Gbps TCP flow, 80ms RTT.
See Eli Dart, Lauren Rotman, Brian Tierney, Mary Hester, and Jason Zurawski. The Science DMZ: A Network Design Pattern for Data-Intensive Science. In *Proceedings of the IEEE/ACM Annual SuperComputing Conference (SC13)*, Denver CO, 2013.