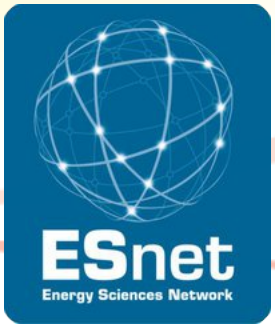


July 22nd 2013, XSEDE Network Performance Tutorial

Jason Zurawski – Internet2/ESnet

Kathy Benninger - Pittsburgh Supercomputing Center

Using XSEDE & General Measurement Tools

Outline

- Introduction
- Ping
- Traceroute/Tracepath
- NDT
- NPAD
- XSEDE maddash-gui Dashboard
- GridFTP

Underlying Assumption

- When problems exist, it's the network's fault!
- Corollary – what is a network really?
  - I contend that when we throw buzzwords around like 'the cloud', 'the network' stretches to our applications and fingers instead of ending at some piece of hardware.
- Easy to blame a resource, but where else could a problem be when transferring large data sets?
  - Host (Disk, CPU, Kernel, NIC Drivers)
  - Network Interface Cards
  - Routers/Switches, Routing and Configuration
  - Physical Infrastructure
  - Protocols
- The network is viewed as a single resource in many cases
  - Reality – complex series of components
  - Multiple vendors/technologies
  - Multiple configuration options
  - Crossing administrative domains

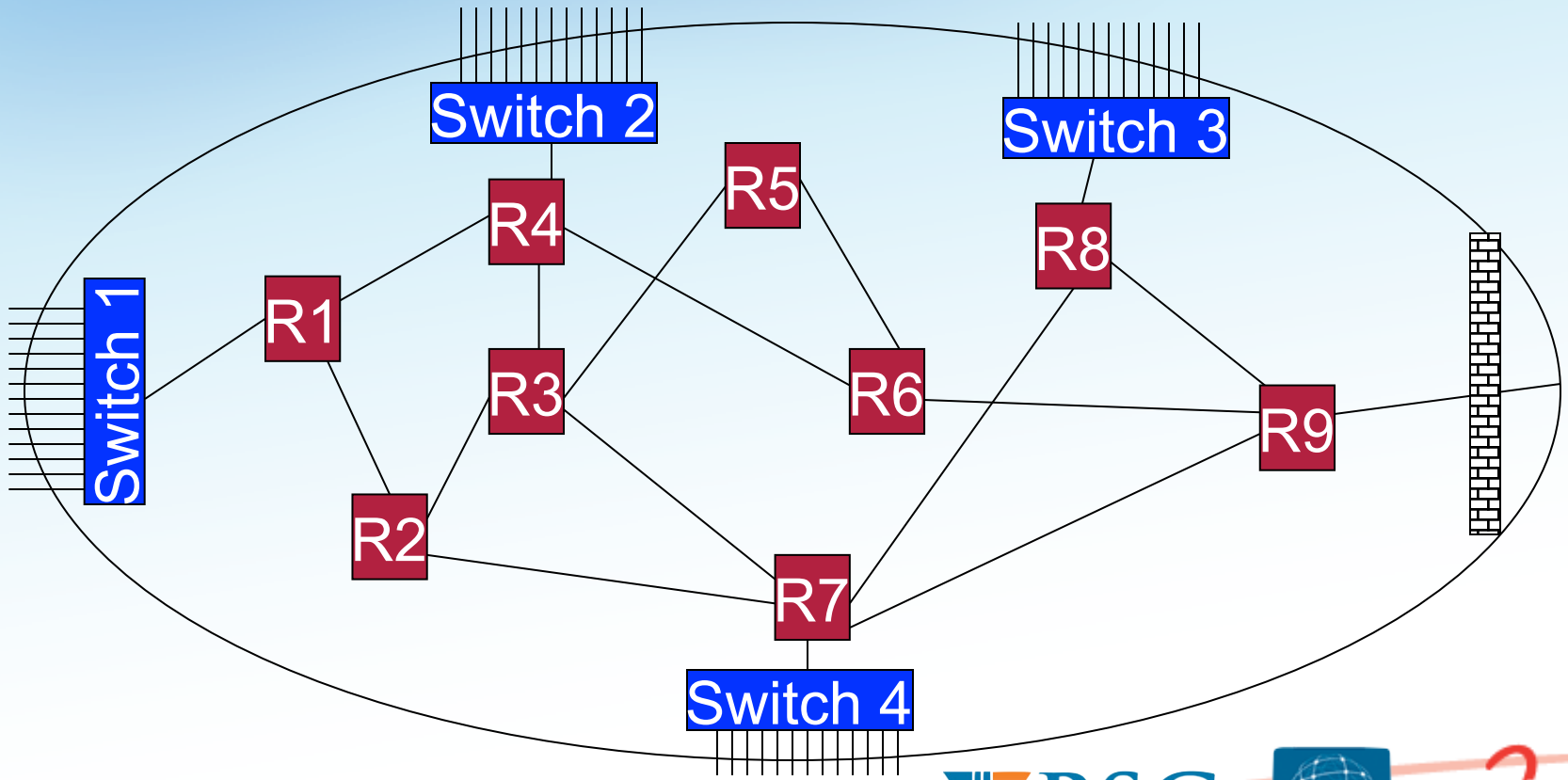
Network View (Layman's Terms)

Bob's Host

"The Internets"

Carol's Host

Network View (Actual)



J. Zurawski – zurawski@es.net & K. Benninger - benninger@psc.edu

Addressing a Performance Discrepancy

- What are the first steps to address problems related to network performance?
  - Try a Tool
- What tools are out there
  - Numerous
  - Different metrics (measurements) available
  - How to interpret the results?

Tools, Tools, Tools

- Ping
- Traceroute
- Iperf
- Tcpdump
- Tcptrace
- BWCTL
- NDT
- OWAMP
- AMP
- Advisor
- Thrulay
- Web100
- MonaLisa
- pathchar
- NPAD
- Pathdiag
- Surveyor
- Ethereal
- CoralReef
- MRTG
- Skitter
- Cflowd
- Cricket
- Net100
- Pathload
- Pathchrip
- MRTG
- Cacti
- Smokeping
- PingER
- FDT
- perfSONAR
- Nagios
- Ganglia
- Thurlay
- Etc. etc. etc.

Highlighting some Interesting Tools

- Focus on 2 Types of tools (for now)
  - Basic Diagnostics - Ping, Traceroute
- What about the others?
  - Try them out, learn how they work.
  - Most tools are designed to solve a specific problem and they may add value to your organization
- Integration of multiple solutions
  - Measurement frameworks integrate use of tools (operation, collecting results) along with analysis and presentation
  - perfSONAR

- ***Feel Free to try this on the VMs as I talk!***

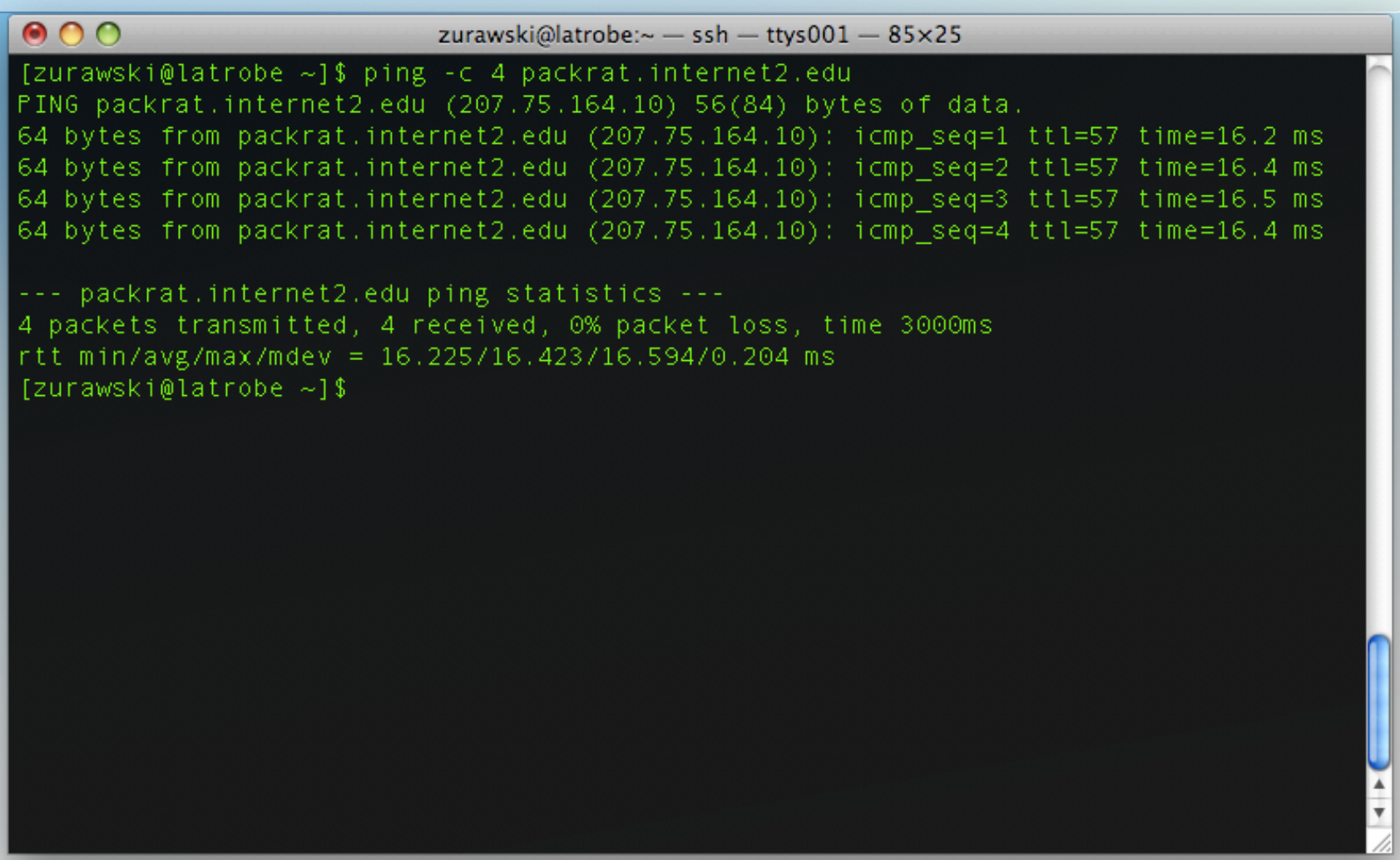
Outline

- Introduction
- Ping
- Traceroute/Tracepath
- NDT
- NPAD
- XSEDE maddash-gui Dashboard
- GridFTP

Basic Diagnostic Tools

- Ping
  - Round Trip (e.g. source to destination, and back)
  - Confirms that remote host is 'up'
  - Some network operators block these packets
    - Play w/ command options to see if that will change anything

Ping Output


zurawski@latrobe:~ — ssh — ttys001 — 85×25
[zurawski@latrobe ~]$ ping -c 4 packrat.internet2.edu
PING packrat.internet2.edu (207.75.164.10) 56(84) bytes of data.
64 bytes from packrat.internet2.edu (207.75.164.10): icmp_seq=1 ttl=57 time=16.2 ms
64 bytes from packrat.internet2.edu (207.75.164.10): icmp_seq=2 ttl=57 time=16.4 ms
64 bytes from packrat.internet2.edu (207.75.164.10): icmp_seq=3 ttl=57 time=16.5 ms
64 bytes from packrat.internet2.edu (207.75.164.10): icmp_seq=4 ttl=57 time=16.4 ms
--- packrat.internet2.edu ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 16.225/16.423/16.594/0.204 ms
[zurawski@latrobe ~]$

Ping Output

- Intro message
  - Identifies remote host name and IP address
  - States size of packets being sent
    - Setting larger sizes may reveal hidden problems
- Output lines
  - Who responded, and the RTT, maybe other details
- Summary lines
  - Number of packets sent/received/lost
  - RTT statistics min/average/max

N.B. 1 msec RTT = ~50 miles of between hosts on the WAN.
Longest RTT on XSEDE is currently ~60 msec, NICS<->SDSC

(We can argue on the differences between Copper and Glass, but its not consequential unless the distances are great)

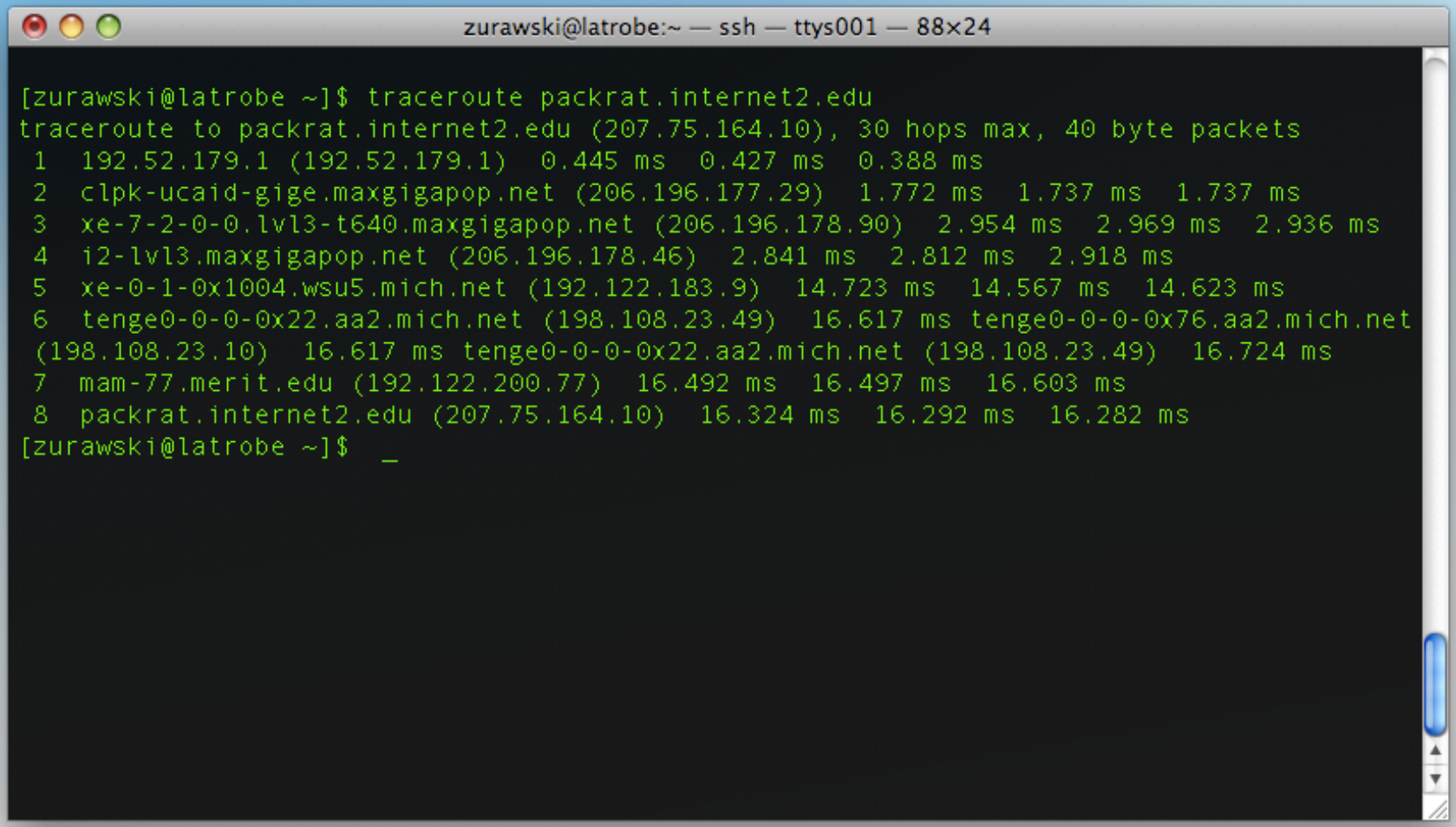
Outline

- Introduction
- Ping
- Traceroute/Tracepath
- NDT
- NPAD
- XSEDE maddash-gui Dashboard
- GridFTP

Basic Diagnostic Tools

- Traceroute
  - Identifies the routers along the path
  - Same blocking problem as above
  - Routers treat ICMP packets with lower priority
    - See presentation from prior JTs:
    - http://www.internet2.edu/presentations/jt2009jul/20090722-litvanyi.pdf

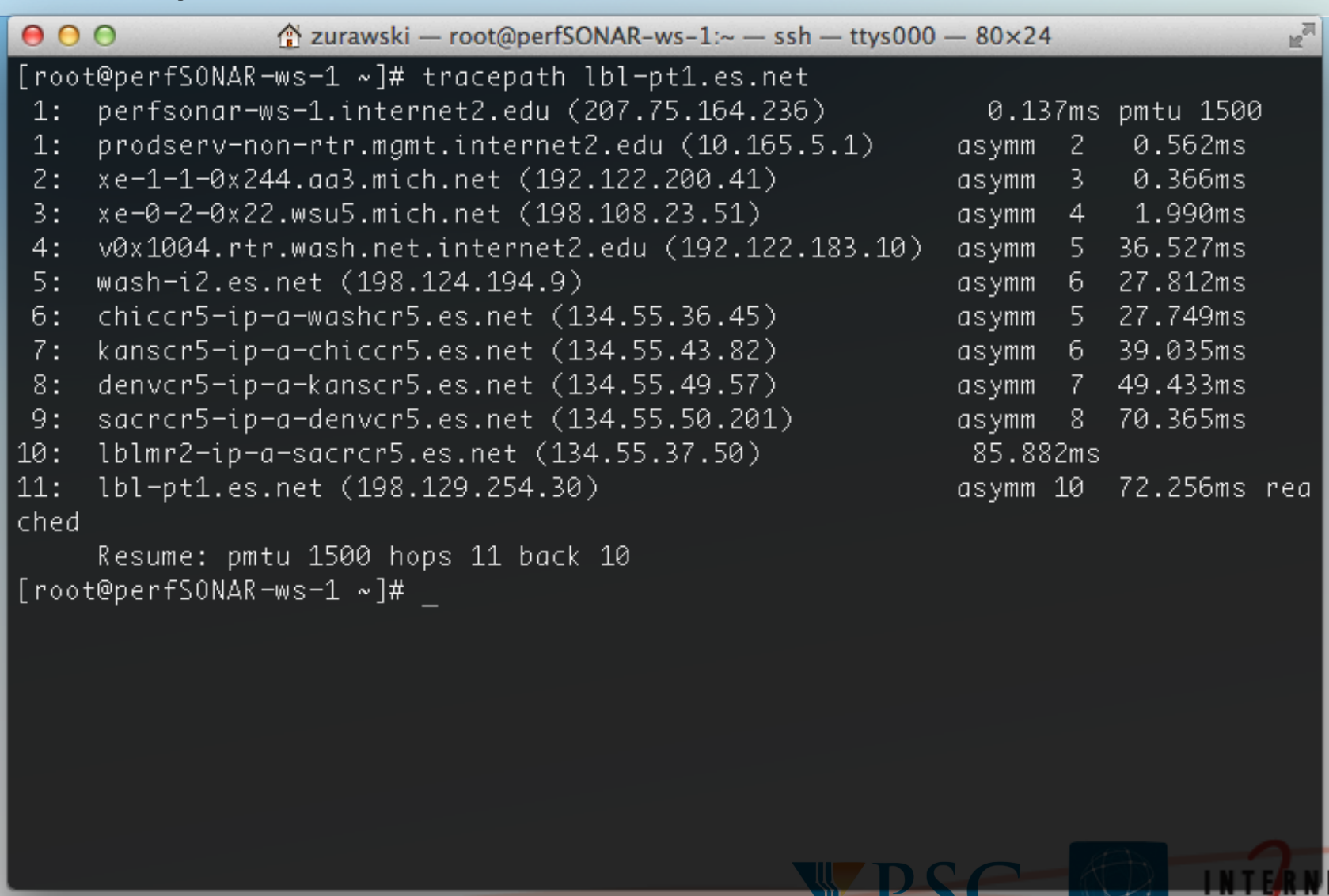
Traceroute Output



Traceroute Output

- Intro messages
  - Name and address of remote host
  - Maximum number of link before giving up
- Status messages
  - One line per router in path
  - ‘*’ indicates router didn’t respond
  - Routers usually rate limit replies
  - No name indicates DNS entry is missing
  - Hops required to reach remote host or max number from above

Tracepath


zurawski — root@perfSONAR-ws-1:~ — ssh — ttys000 — 80×24
[root@perfSONAR-ws-1 ~]# tracepath lbl-pt1.es.net
1: perfsonar-ws-1.internet2.edu (207.75.164.236) 0.137ms pmtu 1500
1: prodserv-non-rtr.mgmt.internet2.edu (10.165.5.1) asymm 2 0.562ms
2: xe-1-1-0x244.aa3.mich.net (192.122.200.41) asymm 3 0.366ms
3: xe-0-2-0x22.wsu5.mich.net (198.108.23.51) asymm 4 1.990ms
4: v0x1004.rtr.wash.net.internet2.edu (192.122.183.10) asymm 5 36.527ms
5: wash-i2.es.net (198.124.194.9) asymm 6 27.812ms
6: chiccr5-ip-a-washcr5.es.net (134.55.36.45) asymm 5 27.749ms
7: kanscr5-ip-a-chiccr5.es.net (134.55.43.82) asymm 6 39.035ms
8: denvcr5-ip-a-kanscr5.es.net (134.55.49.57) asymm 7 49.433ms
9: sacrcr5-ip-a-denvcr5.es.net (134.55.50.201) asymm 8 70.365ms
10: lblmr2-ip-a-sacrcr5.es.net (134.55.37.50) 85.882ms
11: lbl-pt1.es.net (198.129.254.30) asymm 10 72.256ms rea
ched
Resume: pmtu 1500 hops 11 back 10
[root@perfSONAR-ws-1 ~]# _

Tracepath

- Traces path to a network host discovering MTU along this path (UDP Based)
- Output:
  - First column = TTL of the probe
    - TTL is obtained from reply from network (usually)
    - If we don't get this, a guess will be made (number is followed by '?')
  - Second column = network hop, which replied to the probe and other information:
    - value of RTT.
    - Path MTU, if/when it changes.
    - Asymmetric indicator or the probe finishes before it reach prescribed hop & difference between number of hops in forward and backward direction (N.B. most of this is an application guess, Caveat Emptor)
  - Last line = information about all the path
    - Detected Path MTU
    - Amount of hops to the destination
    - Guess about amount of hops from the destination to us (asymmetry could make this different)

Outline

- Introduction
- Ping
- Traceroute/Tracepath
- NDT
- NPAD
- XSEDE maddash-gui Dashboard
- GridFTP

Hands on Testing of NDT

- Open a browser (any will do), you will need Java installed/ enabled (working on that though …)
- MLab (Commodity Networking)
  - http://ndt.iupui.donar.measurement-lab.org:7123/
- Internet2 (R&E Networking)
  - http://ndt.losa.net.internet2.edu:7123/
  - To not overwhelm the server, also try replacing 'losa' with:
    - atla
    - chic
    - hous
    - kans
    - newy
    - salt
    - seat
    - wash

Hands on Testing of NDT

- XSEDE
  - http://ps.psc.xsede.org:7123/
  - To not overwhelm the server, also try replacing 'psc' with:
    - iu
    - ncar
    - ncsa
    - nics
    - purdue
    - sdsc
    - tacc

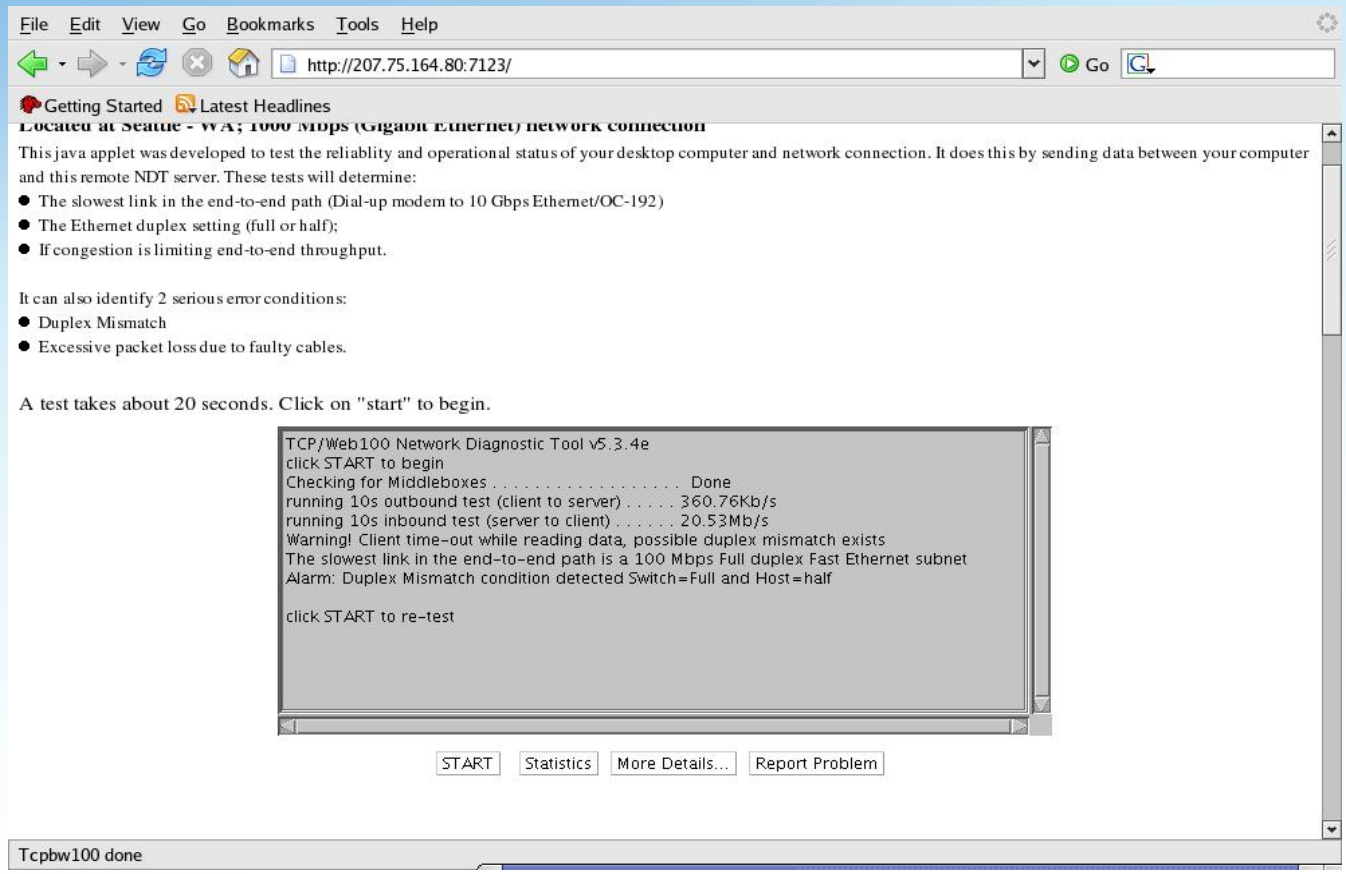
Hands on Testing of NDT

- Command Line Version
- Run: **`web100clt -n ndt.losa.net.internet2.edu`**
  - To not overwhelm the server, also try replacing 'losa' with:
    - atla
    - chic
    - hous
    - kans
    - newy
    - salt
    - seat
    - wash
- Use "**`-l`**" or "**`-ll`**" flags for more more information
- Use the "**`-d`**" or "**`-dd`**" to see debug information (not very interesting)

NDT User Interface

- Web-based JAVA applet allows testing from any browser
  - One Click testing
  - Option to dig deep into available results
  - Send report of results to network administrators
- Unpopularity of JAVA will force a different UI
  - Mlab project has an HTML5 skin they are working on
  - For now there are no other options, volunteers welcome
- Command-line client allows testing from remote login shell
  - Same options available
  - Client software can be built independent of server software

NDT Results


File Edit View Go Bookmarks Tools Help
http://207.75.164.80:7123/
Go
Getting Started
Latest Headlines
This java applet was developed to test the reliablity and operational status of your desktop computer and network connection. It does this by sending data between your computer and this remote NDT server. These tests will determine:
The slowest link in the end-to-end path (Dial-up modem to 10 Gbps Ethernet/OC-192)
The Ethernet duplex setting (full or half);
If congestion is limiting end-to-end throughput.
It can also identify 2 serious error conditions:
Duplex Mismatch
Excessive packet loss due to faulty cables.
A test takes about 20 seconds. Click on "start" to begin.
TCP/Web100 Network Diagnostic Tool v5.3.4e
click START to begin
Checking for Middleboxes . . . . . Done
running 10s outbound test (client to server) 360.76Kb/s
running 10s inbound test (server to client) 20.53Mb/s
Warning! Client time-out while reading data, possible duplex mismatch exists
The slowest link in the end-to-end path is a 100 Mbps Full duplex Fast Ethernet subnet
Alarm: Duplex Mismatch condition detected Switch=Full and Host=half
click START to re-test
START
Statistics
More Details...
Report Problem
Tcpbw100 done

Motivation for Work

- Measure performance ***to user's machine***
  - Lots of tools to measure performance to a nearby server
  - Also 'pluggable' hardware to measure everything up to the network cable
  - Want something to accurately show ***what the user is seeing***
- Develop "single shot" diagnostic tool that doesn't use historical data
- Combine numerous Web100 variables to analyze connection
- Develop network signatures for 'typical' network problems
  - Based on heuristics and experience
  - Lots of problems have a ***smoking gun*** pattern, e.g. duplex mismatch, bad cable, etc.

How It works

- Simple bi-directional test to gather end to end data
  - Test from client to server, and the reverse
  - Gets the 'upload' and 'download' directions
- Gather multiple data variables from server
  - Via Web100, also some derived metrics (packet inter arrival times)
- Compare measured performance to analytical values
  - How fast ***should*** a connection be, given the observations of the host and network
- Translate network values into plain text messages
- Geared toward campus area network

Web100 Project

- Joint PSC/NCAR project funded by NSF
- Develop a ***system mib***, similar to data that is exposed via SNMP
- 'First step' to gather TCP data
  - Kernel Instrument Set (KIS)
- Requires patched Linux kernel
- Geared toward wide area network performance
- Goal is to automate tuning to improve application performance
- Patches available for ***vanilla*** kernels (e.g. non vendor modified)

- Web10G
  - API and Kernel patches under testing
  - Proof of Concept NDT also in testing
  - Expected to release a version of pSPT sometime in 2013 with both

Web Based Performance Tool

- Operates on Any client with a Java enabled Web browser
  - No additional client software needs to be installed
  - No additional configuration required
- What it can do:
  - State if Sender, Receiver, or Network is operating properly
  - Provide accurate application tuning info
  - Suggest changes to improve performance
- What it can't do
  - Tell you exactly where in the network the problem is
  - Tell you how well or poorly "other" servers perform
  - Tell you how well or poorly "other" clients will perform

Dissecting Results (in Bold)

```
[zurawski@head ~]$ web100clt -n ndt.chic.net.internet2.edu -l
Testing network path for configuration and performance problems  --  Using IPv6 address
Checking for Middleboxes . . . . . . . . . . . . . . . . . . .  Done
checking for firewalls . . . . . . . . . . . . . . . . . . . .  Done
running 10s outbound test (client to server) . . . . .  91.62 Mb/s
running 10s inbound test (server to client) . . . . . . 91.05 Mb/s
The slowest link in the end-to-end path is a 100 Mbps Full duplex Fast Ethernet subnet
Information: Other network traffic is congesting the link
Information [S2C]: Packet queuing detected: 15.19% (local buffers)
Server 'ndt.chic.net.internet2.edu' is not behind a firewall. [Connection to the ephemeral port was
successful]
Client is probably behind a firewall. [Connection to the ephemeral port failed]

      ------  Web100 Detailed Analysis  ------

Web100 reports the Round trip time = 34.20 msec;the Packet size = 1428 Bytes; and
There were 153 packets retransmitted, 731 duplicate acks received, and 883 SACK blocks received
Packets arrived out-of-order 1.70% of the time.
This connection is sender limited 86.70% of the time.
This connection is network limited 12.02% of the time.

    Web100 reports TCP negotiated the optional Performance Settings to:
RFC 2018 Selective Acknowledgment: ON
RFC 896 Nagle Algorithm: ON
RFC 3168 Explicit Congestion Notification: OFF
RFC 1323 Time Stamping: ON
RFC 1323 Window Scaling: ON; Scaling Factors - Server=7, Client=10
The theoretical network limit is 65.51 Mbps
The NDT server has a 32768 KByte buffer which limits the throughput to 7485.38 Mbps
Your PC/Workstation has a 2033 KByte buffer which limits the throughput to 464.47 Mbps
The network based flow control limits the throughput to 180.30 Mbps

Client Data reports link is '  5', Client Acks report link is '  5'
Server Data reports link is '  5', Server Acks report link is '  5'
Information: Network Middlebox is modifying MSS variable (changed to 1440)
Server IP addresses are preserved End-to-End
Client IP addresses are preserved End-to-End
```


J. Zurawski – zurawski@es.net & K. Benninger - benninger@psc.edu

Finding Results of Interest

- Duplex Mismatch
  - This is a serious error and nothing will work right. Reported on *main* page, on *Statistics* page, and **mismatch:** on *More Details* page
- Packet Arrival Order
  - Inferred value based on TCP operation. Reported on *Statistics* page, (with loss statistics) and **order:** value on *More Details* page
- Packet Loss Rates
  - Calculated value based on TCP operation. Reported on *Statistics* page, (with out-of-order statistics) and **loss:** value on *More Details* page
- Path Bottleneck Capacity
  - Measured value based on TCP operation. Reported on *main* page

General Requirements – Support

- Source for client should compile for all modern *NIX
  - *BSD, Linux, OS X
  - configure/make/make install
- Web100 Patched Linux Kernel (2.6 lineage) required for server
  - perfSONAR-PS Project also offers two alternatives:
    - pS Performance Toolkit (bootable ISO)
    - Pre-packaged kernel with Web100 for CentOS 5 & 6 (http://software.internet2.edu)
- Other Software
  - Java SDK
  - libpcap
- RPMs compiled specifically for CentOS 5.x and 6.x
  - May work with other RPM based systems (Fedora, RHEL)

Potential Risks

- Non-standard kernel required
  - Web100 patching may be difficult to apply to new kernels
  - Hard to keep up with vendor patching
  - GUI tools can be used to monitor other ports
  - Consider using pS Performance Toolkit enhancements if this scares you…
- Public servers generate trouble reports from remote users
  - Respond or ignore emails
- Test streams can trigger IDS alarms
  - Configure IDS to ignore NDT server

Availability

- Main Page:
  - http://www.internet2.edu/performance/ndt
  - http://software.internet2.edu
- Mailing lists:
  - ndt-users@internet2.edu
  - ndt-announce@internet2.edu

Outline

- Introduction
- Ping
- Traceroute/Tracepath
- NDT
- NPAD
- XSEDE maddash-gui Dashboard
- GridFTP

Why is the end-to-end problem so difficult?

- By design TCP/IP hides the ‘net from upper layers
  - TCP/IP provides basic reliable data delivery
  - The “hour glass” between applications and networks
- This is a good thing, because it allows:
  - Invisible recovery from data loss, etc
  - Old applications to use new networks
  - New application to use old networks
- But then (nearly) all problems have the same symptom
  - Less than expected performance
  - The details are hidden from nearly everyone

TCP Tuning is painful debugging

- All problems reduce performance
  - But the specific symptoms are hidden
- Any one problem can prevent good performance
  - Completely masking all other problems
- Trying to fix the weakest link of an invisible chain
  - General tendency is to guess and “fix” random parts
  - Repairs are sometimes “random walks”
  - Repair one problem at time at best
- The solution is to instrument TCP

The Web100 project

- Use TCP's ideal diagnostic vantage point
  - What is limiting the data rate?
  - RFC 4898 TCP-ESTATS-MIB
    - Standards track
    - Prototypes for Linux (www.Web100.org) and Windows Vista
  - Also TCP Autotuning
    - Automatically adjusts TCP buffers
    - Linux 2.6.17 default maximum window size is 4 M Bytes
    - Included in Windows 7 (may need fix – Microsoft Article ID: 983528)
- But this has led to a new insight:
  - Nearly all symptoms scale with round trip time

Nearly all symptoms scale with RTT

- Examples
  - TCP Buffer Space: $Rate = Window / RTT$
  - Packet loss: $Rate = (MSS / RTT)(1 / \sqrt{Loss})$
- Think: the extra time needed to overcome a flaw is proportional to the RTT

J. Zurawski – zurawski@es.net & K. Benninger - benninger@psc.edu

Symptom scaling breaks diagnostics

- Local Client to Server
  - Flaw has insignificant symptoms
  - All applications work, including all standard diagnostics
  - False pass all diagnostic tests
- Remote Client to Server: all applications **fail**
  - Leading to faulty implication of other components
    - Implies that the flaw is in the wide are network

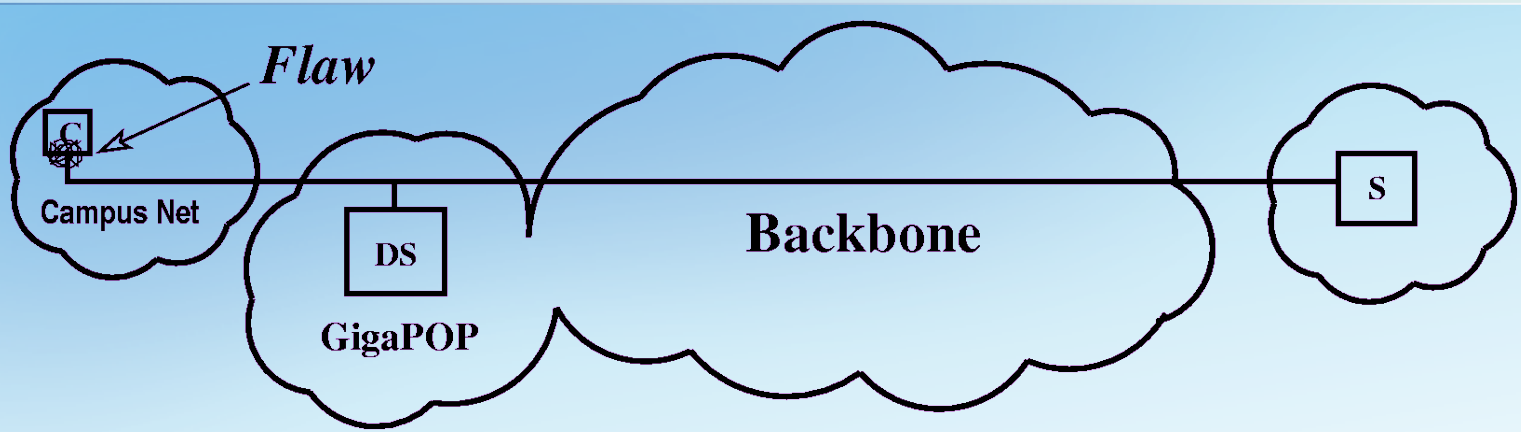
The confounded problems

- For nearly all network flaws
  - The only symptom is reduced performance
  - But the reduction is scaled by RTT
- Therefore, flaws are undetectable on short paths
  - False pass for even the best conventional diagnostics
  - Leads to faulty inductive reasoning about flaw locations
  - Diagnosis often relies on tomography and complicated inference techniques
- ***This is the real end-to-end problem***

The pathdiag solution

- Test a short section of the path
  - Most often first or last mile
- Use Web100 to collect detailed TCP statistics
  - Loss, delay, queuing properties, etc
- Use models to extrapolate results to the full path
  - Assume that the rest of the path is ideal
  - You have to specify the end-to-end performance goal
    - Data rate and RTT
- Pass/Fail on the basis of the extrapolated performance
- Note: Web100 is being replaced by Web10G

Deploy as a Diagnostic Server



- Use pathdiag in a Diagnostic Server (DS)
- Specify End to End target performance
  - From server (S) to client (C) (RTT and data rate)
- Measure the performance from DS to C
  - Use Web100 in the DS to collect detailed statistics
    - On both the path and client
  - Extrapolate performance assuming ideal backbone
- Pass/Fail on the basis of extrapolated performance

Hands on Testing of NPAD

- Internet2 (R&E Networking)
  - http://npad.losa.net.internet2.edu:8000
  - To not overwhelm the server, also try replacing 'losa' with:
    - atla
    - chic
    - kans
    - hous
    - newy
    - salt
    - seat
    - wash

Hands on Testing of NPAD

- XSEDE
  - http://ps.psc.xsede.org:8000/
  - To not overwhelm the server, also try replacing 'psc' with:
    - iu
    - ncar
    - ncsa
    - nics
    - purdue
    - sdsc
    - tacc

Results

Test conditions

Tester: (none) (192.88.115.171) [?]
Target: (none) (209.68.2.209) [?]
Logfile base name: hyaku.pair.com:2007-02-28-23:19:31 [?]
This report is based on a 90 Mb/s target application data rate [?]
This report is based on a 20 ms Round-Trip-Time (RTT) to the target application [?]
The Round Trip Time for this path section is 2.518223 ms.
The Maximum Segment Size for this path section is 1448 Bytes. [?]

Target host TCP configuration test: Fail! [?]

Warning: TCP connection is not using SACK. [?]
Critical Failure: Received window scale is 2, it should be 3. [?]
The maximum receiver window (128k) is too small for this application (and/or some tests). [?]
Diagnosis: The target (client) is not properly configured. [?]
> See TCP tuning instructions at http://www.psc.edu/networking/projects/tcptune/ [?]

Path measurements [?]

Data rate test: Pass! [?]

Pass data rate check: maximum data rate was 93.900110 Mb/s [?]

Loss rate test: Pass! [?]

Pass: measured loss rate 0.000848% (117889 packets between loss events). [?]
FYI: To get 90 Mb/s with a 1448 byte MSS on a 20 ms path the total end-to-end loss budget is 0.002029% (49275 packets between losses). [?]

Results

Path measurements [?]

Data rate test: Pass! [?]

Pass data rate check: maximum data rate was 93.900110 Mb/s [?]

Loss rate test: Pass! [?]

Pass: measured loss rate 0.000848% (117889 packets between loss events). [?]
FYI: To get 90 Mb/s with a 1448 byte MSS on a 20 ms path the total end-to-end loss budget is 0.002029% (49275 packets between losses). [?]

Suggestions for alternate tests

FYI: This path may even pass with a more strenuous application: [?]
Try rate=90 Mb/s, rtt=30 ms
Try rate=93 Mb/s, rtt=29 ms
Or if you can raise the MTU: [?]
Try rate=90 Mb/s, rtt=192 ms, mtu=9000 bytes
Try rate=93 Mb/s, rtt=184 ms, mtu=9000 bytes

Network buffering test: Warning! [?]

This test did not complete due to other problems with the path, target or tester.
> Correct other problems first, and then rerun this test. [?]
Estimated queue size is at least: Pkts: 64 Bytes: 92672
This is probably an underestimate of the actual queue size. [?]
This corresponds to a 7.737751 ms drain time. [?]
To get 90 Mb/s with on a 20 ms path, you need 225000 bytes of buffer space. [?]

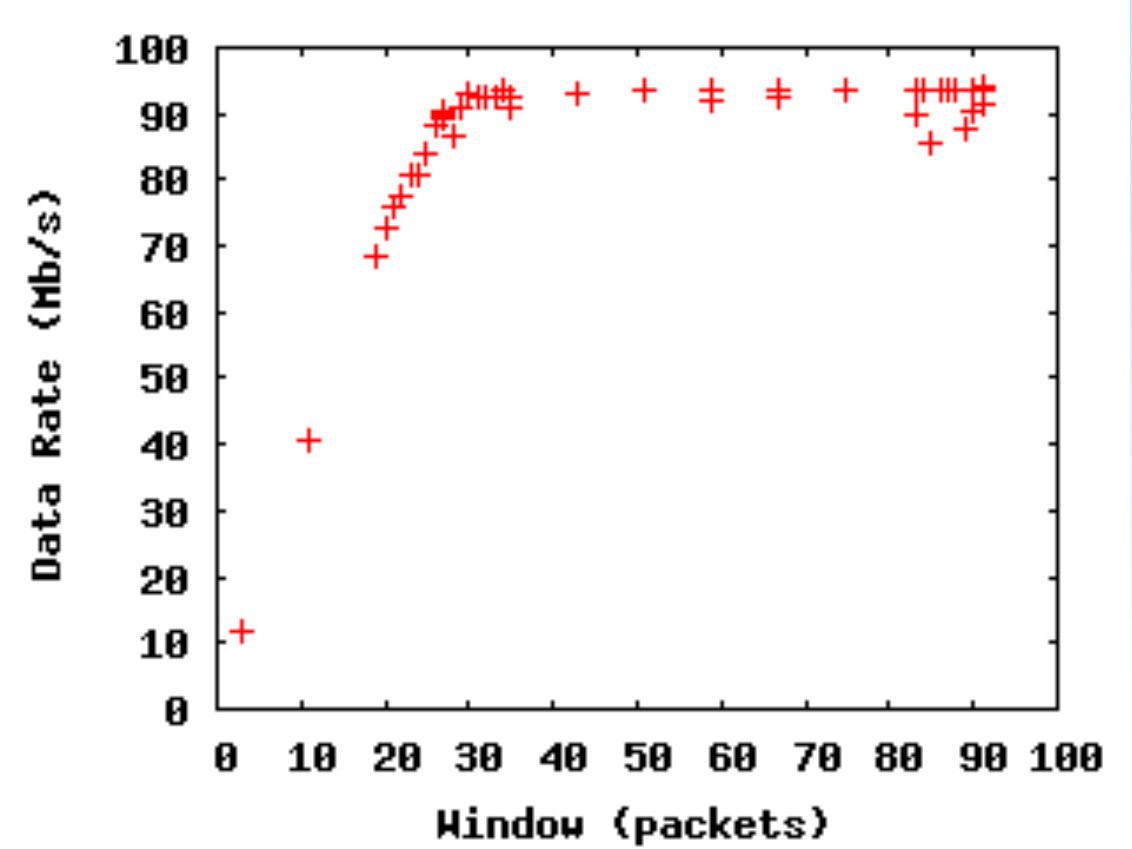
Tester validation: Pass! [?]

No internal tester problems were detected.
Tester version: 4.1.1 (c) 2013 ESnet, Internet2, PSC

Pathdiag

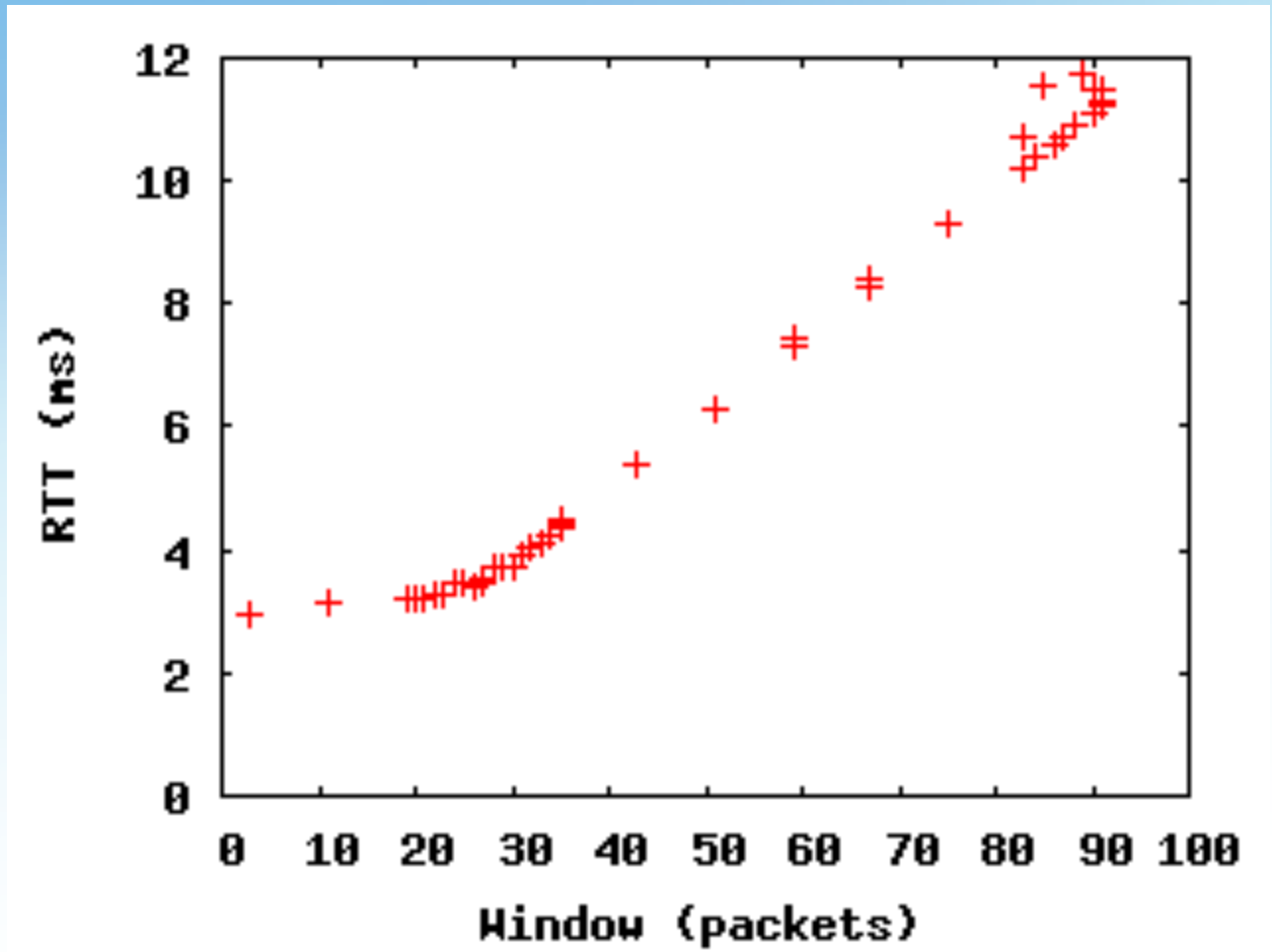
- One click automatic performance diagnosis
  - Designed for use by non-expert end users
  - Accurate end-system and last mile diagnosis
    - Eliminate most false pass results
    - Accurate distinction between host and path flaws
    - Accurate and specific identification of most flaws
  - Basic networking tutorial info
    - Help the end user understand the problem
    - Help train 1st tier support (sysadmin or netadmin)
    - Backup documentation for support escalation
- Empower the user to get the problem fixed
  - The same reports for users and admins

Window Size vs Data Rate



- Try this w/ BWCTL …

Window Size vs RTT



J. Zurawski – zurawski@es.net & K. Benninger - benninger@psc.edu

Key NPAD/pathdiag features

- Test results:
  - Provide a list of specific items to be corrected
    - ***Failed tests are show stoppers for fast apps***
  - Include explanations and tutorial information
  - Clear differentiation between client and path problems
  - Accurate escalation to network or system admins
  - The reports are public and can be viewed by either
- Coverage for a majority of OS and last-mile network flaws
  - Coverage is one way – need to reverse client and server
  - Does not test the application – need application tools
  - Does not check routing – need traceroute
  - ***Eliminates nearly all(?) false pass results***

More features

- Tests becomes more sensitive as the path gets shorter
  - Conventional diagnostics become less sensitive
  - Depending on models, perhaps too sensitive
    - New problem is false fail (e.g. queue space tests)
- Flaws no longer completely mask other flaws
  - A single test often detects several flaws
    - E.g. Can find both OS and network flaws in the same test
  - They can be repaired concurrently
- Archived DS results include raw web100 data
  - Can reprocess with updated reporting SW
    - New reports from old data

Download and install

- User documentation:

http://www.psc.edu/index.php/npad

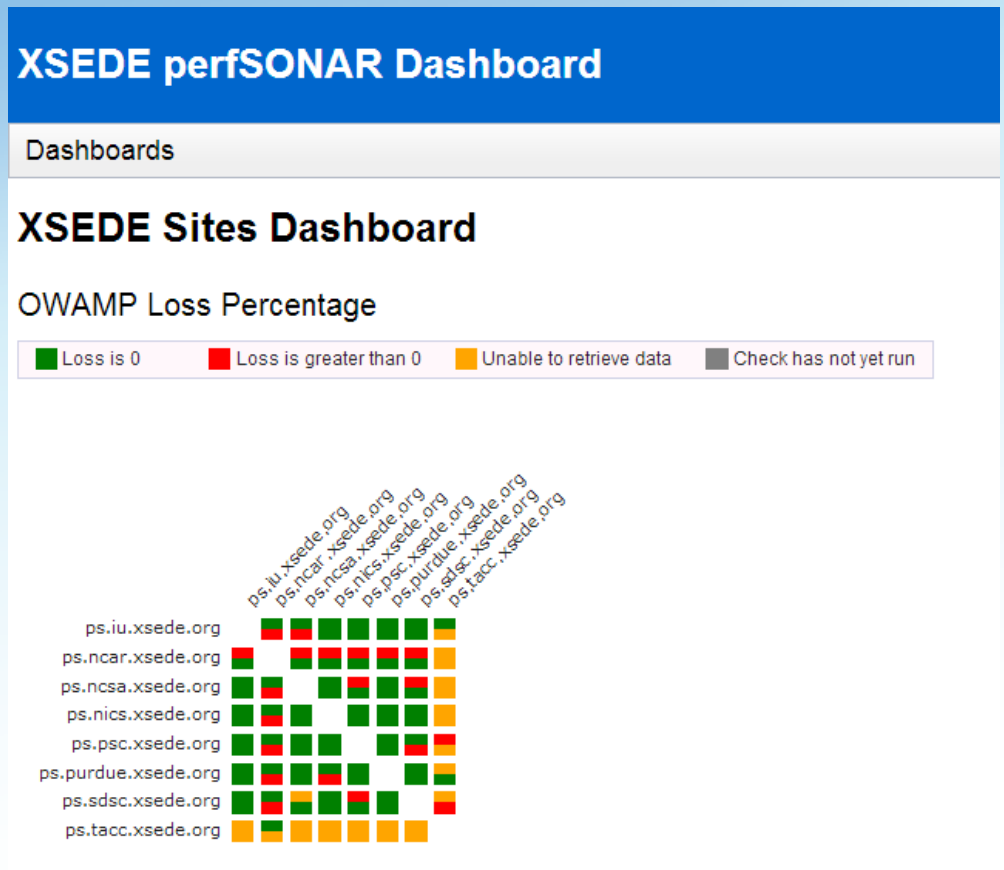
- Follow the link to “Installing a Server”
  - Easily customized with a site specific skin
  - Designed to be easily upgraded with new releases
    - Roughly every 2 months
    - Improving reports through ongoing field experience
  - Drops into existing NDT servers
    - Plans for future integration
- Web100 Kernels and NPAD RPM (CentOS 5.x, 6.x x86 and 64 bit) available
  - http://software.internet2.edu

Outline

- Introduction
- Ping
- Traceroute/Tracepath
- NDT
- NPAD
- XSEDE maddash-gui Dashboard
- GridFTP

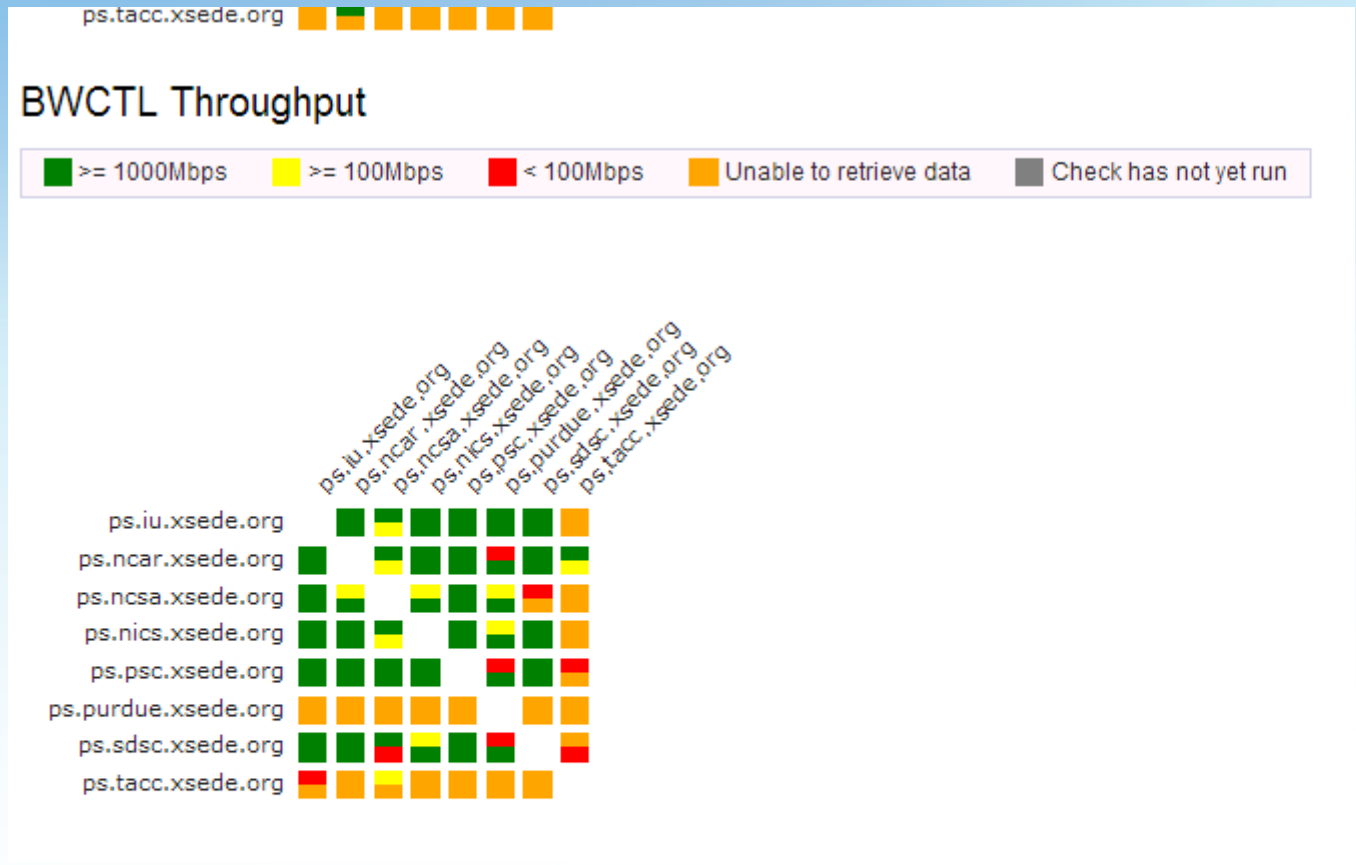
XSEDE maddash-gui perfSONAR Dashboard

- http://psarch.psc.xsede.org/maddash-webui



XSEDE maddash-gui perfSONAR Dashboard

- http://psarch.psc.xsede.org/maddash-webui



Outline

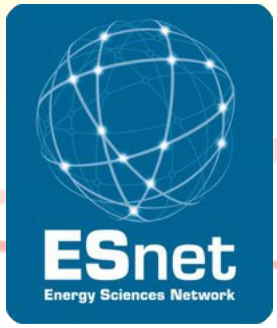
- Introduction
- Ping
- Traceroute/Tracepath
- NDT
- NPAD
- XSEDE maddash-gui Dashboard
- GridFTP

GridFTP

- As used by http://speedpage.psc.edu
- Do-it-yourself "network throughput" test using GridFTP
- Copy /dev/zero to /dev/null
- globus-url-copy -vb -stripe -len 2000000000 gsiftp://gridftp.blacklight.psc.xsede.org/dev/zero gsiftp://gridftp.ranger.tacc.xsede.org/dev/null
  - **-vb** 'verbose' invokes reporting of average and instantaneous MBytes/sec throughput at 5 sec intervals throughout transfer
  - **-stripe** will send a single stream to each one of multiple servers if available
  - **-len** number of bytes to transfer
  - **-p <number>** will send parallel streams to each server
  - **-tcp-bs <bs>** TCP buffer size. Can try various values, but with modern systems it's usually better to let auto-tuning do its thing

From host: NICS Kraken Cray XT5																					
	05/09/2013			05/08/2013			05/07/2013			05/06/2013			05/05/2013			05/04/2013			05/03/2013		
To host:	hi	lo	avg	hi	lo	avg	hi	lo	avg	hi	lo	avg	hi	lo	avg	hi	lo	avg	hi	lo	avg
/dev/zero -> /dev/null	793.6	710.3	752.0	732.5	711.5	722.0	748.6	497.5	623.0	811.0	790.7	800.9	690.7	685.4	688.0	759.4	712.0	735.7	821.0	289.8	555.4
/dev/zero -> file	522.8	328.4	425.6	485.3	476.8	481.0	466.7	228.2	347.5	430.1	345.5	387.8	474.2	460.7	467.5	460.2	434.5	447.3	500.9	308.9	404.9
NICS Keeneland	login	login	N/A	login	login	N/A	login	login	N/A	login	login	N/A	login	login	N/A	login	login	N/A	login	login	N/A
NICS Kraken Cray XT5	272.0	225.2	248.6	254.6	227.5	241.1	236.0	113.1	174.5	240.7	165.6	203.2	276.7	224.0	250.4	233.3	189.1	211.2	213.7	92.7	153.2
NICS Nautilus	53.7	25.5	39.6	244.3	130.5	187.4	154.3	100.0	127.1	178.6	97.5	138.1	312.3	106.1	209.2	270.3	102.9	186.6	179.8	147.2	163.5
PSC Blacklight	46.1	37.9	42.0	55.1	47.1	51.1	34.9	26.2	30.6	36.9	32.6	34.7	login	login	N/A	53.7	36.3	45.0	chk	login	N/A
PSC Data Supercell	57.0	45.8	51.4	25.2	chk	25.2	42.6	time	42.6	4.1	time	4.1	login	login	N/A	chk	time	N/A	time	login	N/A
Purdue Steele	69.1	68.9	69.0	65.8	64.7	65.3	66.4	57.7	62.0	68.8	67.9	68.3	69.5	67.9	68.7	68.9	68.9	68.9	chk	chk	N/A
SDSC Oasis	137.0	136.2	136.6	117.6	chk	117.6	135.9	login	135.9	167.9	114.3	141.1	160.3	142.4	151.3	165.3	130.8	148.0	110.7	30.7	70.7
SDSC Trestles	chk	login	N/A	chk	login	N/A	chk	login	N/A	chk	login	N/A	22.8	login	22.8	98.3	28.9	63.6	61.3	21.2	41.3
TACC Lonestar	65.6	57.2	61.4	58.2	login	58.2	146.6	142.7	144.7	117.5	110.2	113.8	113.3	95.0	104.2	114.5	108.9	111.7	110.7	104.6	107.6
TACC Ranch	92.1	75.5	83.8	105.3	67.9	86.6	login	login	N/A	81.0	8.6	44.8	15.5	10.5	13.0	36.4	26.6	31.5	53.3	49.6	51.4
TACC Stampede	100.3	60.1	80.2	73.6	34.9	54.3	146.3	login	146.3	191.5	94.6	143.0	171.9	91.2	131.6	202.1	86.0	144.1	154.2	111.9	133.1
file -> /dev/null	622.2	154.7	388.5	207.2	200.5	203.8	179.3	109.5	144.4	231.0	220.2	225.6	269.3	267.8	268.6	498.0	212.3	355.2	423.9	225.0	324.4

From host: NICS Nautilus																					
	05/09/2013			05/08/2013			05/07/2013			05/06/2013			05/05/2013			05/04/2013			05/03/2013		
To host:	hi	lo	avg	hi	lo	avg	hi	lo	avg	hi	lo	avg	hi	lo	avg	hi	lo	avg	hi	lo	avg



Using XSEDE & General Measurement Tools

July 22nd 2013, XSEDE Network Performance Tutorial

Jason Zurawski – Internet2/ESnet
Kathy Benninger - Pittsburgh Supercomputing Center

For more information, visit http://www.internet2.edu/workshops/npw

perfSONAR powered