January 13th 2013 – TIP2013: Building a Science DMZ
Jason Zurawski – Senior Research Engineer

Performance Measurement & Monitoring via perfSONAR
Outline

- Problem Definition & Motivation
- TCP & Metrics
- perfSONAR overview
- Case studies
- Site deployment recommendations
- perfSONAR host recommendations
- Wrap Up
"In any large system, there is always something broken."

Jon Postel

• Consider the technology:
  – 100G (and larger soon) Networking
  – Changing control landscape (e.g. SDN, be it OSCARS or OpenFlow, or something new)
  – Smarter applications and abstractions

• Consider the realities:
  – Heterogeneity in technologies
  – Mutli-domain operation
  – “old applications on new networks” as well as “new applications on old networks”
Most network design lends itself to the introduction of flaws:
- Heterogeneous equipment
- Cost factors heavily into design – e.g. *Get what you pay for*
- Design heavily favors protection and availability over performance

Communication protocols are not advancing as fast as networks
- *TCP/IP* is the king of the protocol stack
  - Guarantees reliable transfers
  - Adjusts to failures in the network
  - Adjusts speed to be *fair* for all

User Expectations
- “The Network is Slow/Broken” – is this the response to almost any problem? Hardware? Software?
- Empower users to be more informed/more helpful
Local testing will not find all problems

Performance is poor when RTT exceeds 20 ms

Performance is good when RTT is < 20 ms

Switch with small buffers

5 – 2/8/13, © 2012 Internet2 – zurawski@internet2.edu
Soft Network Failures

- Soft failures are where basic connectivity functions, but high performance is not possible.
- TCP was intentionally designed to hide all transmission errors from the user:
  - “As long as the TCPs continue to function properly and the internet system does not become completely partitioned, no transmission errors will affect the users.” (From IEN 129, RFC 716)
- Some soft failures only affect high bandwidth long RTT flows.
- Hard failures are easy to detect & fix
  - soft failures can lie hidden for years!
- One network problem can often mask others
Common Soft Failures

• Packet Loss
  – “Congestive”; the realities of a general purpose network
  – “Non-Congestive”; fixable, if you can find it

• Random Packet Loss
  – Bad/dirty fibers or connectors
  – Low light levels due to amps/interfaces failing
  – Duplex mismatch

• Small Queue Tail Drop
  – Switches not able to handle the long packet trains prevalent in
    long RTT sessions and local cross traffic at the same time

• Un-intentional Rate Limiting
  – Processor-based switching on routers due to faults, acl’s, or mis-
    configuration
  – Security Devices
    • E.g.: 10X improvement by turning off Cisco Reflexive ACL
Sample Results: Finding/Fixing soft failures

Rebooted router with full route table

Gradual failure of optical line card
Say Hello to your Frienemy – The Firewall

• Designed to stop ‘traffic’
  – Read this slowly a couple of times...
  – Performing a read of headers and/or data. Matching signatures

• Contain small buffers
  – Concerned with protecting the network, not impacting your performance

• Will be **a lot** slower than the original wire speed
  – A “**10G Firewall**” may handle 1 flow close to 10G, doubtful that it can handle a couple.

• If **firewall-like** functionality is a must – consider using router filters instead
  – Or per host firewall configurations …
Performance Through the Firewall

• Blue = “Outbound”, e.g. campus to remote location upload
• Green = “Inbound”, e.g. download from remote location
Performance Outside of the Firewall

- Blue = “Outbound”, e.g. campus to remote location upload
- Green = “Inbound”, e.g. download from remote location
- Note – This machine is in the *SAME RACK*, it just bypasses the firewall vs. that of the previous
Firewall Experiment Overview

- 2 Situations to simulate:
  - “Outbound” Bypassing Firewall
    - Firewall will normally not impact traffic leaving the domain. Will pass through device, but should not be inspected
  - “Inbound” Through Firewall
    - Statefull firewall process:
      - Inspect packet header
      - If on cleared list, send to output queue for switch/router processing
      - If not on cleared list, inspect and make decision
      - If cleared, send to switch/router processing.
      - If rejected, drop packet and blacklist interactions as needed.
  - Process slows down all traffic, even those that match a white list
Server & Client (Outbound)

- Run “nuttcp” server:
  - nuttcp -S -p 10200 –nofork

- Run “nuttcp” client:
  - nuttcp -T 10 -i 1 -p 10200 bwctl.newy.net.internet2.edu
  - 92.3750 MB / 1.00 sec = 774.3069 Mbps 0 retrans
  - 111.8750 MB / 1.00 sec = 938.2879 Mbps 0 retrans
  - 111.8750 MB / 1.00 sec = 938.3019 Mbps 0 retrans
  - 111.7500 MB / 1.00 sec = 938.1606 Mbps 0 retrans
  - 111.8750 MB / 1.00 sec = 938.3198 Mbps 0 retrans
  - 111.8750 MB / 1.00 sec = 938.2653 Mbps 0 retrans
  - 111.8750 MB / 1.00 sec = 938.1931 Mbps 0 retrans
  - 111.9375 MB / 1.00 sec = 938.4808 Mbps 0 retrans
  - 111.6875 MB / 1.00 sec = 937.6941 Mbps 0 retrans
  - 111.8750 MB / 1.00 sec = 938.3610 Mbps 0 retrans
  - 1107.9867 MB / 10.13 sec = 917.2914 Mbps 13 %TX 11 %RX 0 retrans 8.38 msRTT
Server & Client (Inbound)

- Run “nuttcp” server:
  - `nuttcp -S -p 10200 -nofork`

- Run “nuttcp” client:
  - `nuttcp -r -T 10 -i 1 -p 10200 bwctl.newy.net.internet2.edu`
  - 4.5625 MB / 1.00 sec = 38.1995 Mbps 13 retrans
  - 4.8750 MB / 1.00 sec = 40.8956 Mbps 4 retrans
  - 4.8750 MB / 1.00 sec = 40.8954 Mbps 6 retrans
  - 6.4375 MB / 1.00 sec = 54.0024 Mbps 9 retrans
  - 5.7500 MB / 1.00 sec = 48.2310 Mbps 8 retrans
  - 5.8750 MB / 1.00 sec = 49.2880 Mbps 5 retrans
  - 6.3125 MB / 1.00 sec = 52.9006 Mbps 3 retrans
  - 5.3125 MB / 1.00 sec = 44.5653 Mbps 7 retrans
  - 4.3125 MB / 1.00 sec = 36.2108 Mbps 7 retrans
  - 5.1875 MB / 1.00 sec = 43.5186 Mbps 8 retrans
  - 53.7519 MB / 10.07 sec = 44.7577 Mbps 0 %TX 1 %RX 70 retrans 8.29 msRTT
• Start “tcpdump” on interface (note – isolate traffic to server’s IP Address/Port as needed):
  - sudo tcpdump -i eth1 -w nuttcp1.dmp net 64.57.17.66
  - tcpdump: listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes
  - 974685 packets captured
  - 978481 packets received by filter
  - 3795 packets dropped by kernel

• Perform “tcptrace” analyses:
  - tcptrace -G nuttcp1.dmp
  - 1 arg remaining, starting with 'nuttcp1.dmp'
  - Ostermann's tcptrace -- version 6.6.7 -- Thu Nov 4, 2004

  - 974685 packets seen, 974685 TCP packets traced
  - elapsed wallclock time: 0:00:33.083618, 29461 pkts/sec analyzed
  - trace file elapsed time: 0:00:10.215806
  - TCP connection info:
    - 1: perfsonar.hep.brown.edu:47617 - nms-rthr2.newy32aoa.net.internet2.edu:5000 (a2b) 18> 17< (complete)
    - 2: perfsonar.hep.brown.edu:60349 - nms-rthr2.newy32aoa.net.internet2.edu:10200 (c2d) 845988> 128662< (complete)
Plotting (Outbound) - Complete
Plotting (Outbound) - Zoom
Plotting (Inbound) - Complete
Outline

• Problem Definition & Motivation
• TCP & Metrics
• perfSONAR overview
• Case studies
• Site deployment recommendations
• perfSONAR host recommendations
• Wrap Up
• Transmission Control Protocol
  – One of the core protocols of the Internet Protocol Suite (along with IP [Internet Protocol])
  – TCP doesn’t relay when things are going wrong via the OS Kernel (e.g. a lost packet is re-transmitted without any knowledge to the application).
  – Loss is actually “required” for TCP to work, this is how it is able to enforce fairness (e.g. Loss means congestion, therefore back off).
  – No distinction between congestive and non-congestive losses
  – Not optimized for modern networks (LFN) by default. Latency has a pretty profound effect on performance.
TCP

• TCP Measurements (from some of the tools we use):
  – Always includes the end system
  – Are sometimes called “memory-to-memory” tests since they don’t involve a spinning disk
  – Set expectations for well coded application

• There are limits of what we can measure
  – TCP *hides* details
  – In hiding the details it can obscure what is causing errors
  – Many things can limit TCP throughput
    • Loss
    • Congestion
    • Buffer Starvation
    • Out of order delivery
General Operational Pattern

- Sender buffers up data to send into segments (respect the MSS) and numbers each
- The ‘window’ is established and packets are sent in order from the window
- The flow of data and ACK packets will dictate the overall speed of TCP for the length of the transfer
- TCP starts fast, until it can establish the available resources on the network.
- The idea is to grow the window until a loss is observed
- This is the signal to the algorithm that it must limit the window for the time being, it can slowly build it back up
TCP – Quick Overview (Slow Start)
TCP – Quick Overview

- General Operational Pattern – cont
  - Receiver will acknowledge packets as they arrive
    - ACK Each (old style)
    - Cumulative ACK (“I have seen everything up to this segment”)
    - Selective ACK (sent to combat a complete retransmit of the window)
  - TCP relies on loss to a certain extent – it will adjust it’s behavior after each loss
    - Congestive (e.g. reaching network limitation, or due to traffic)
    - Non-congestive (due to actual problems in the network)
  - Congestion avoidance stage follows slow start, window will remain a certain size and data rates will increase/decrease based on loss in the network
  - Congestion Control algorithms modify the behavior over time
    - Control how large the window may grow
    - Control how fast to recover from any loss
Parallel streams can help in some situations

- TCP attempts to be “fair” and conservative
- Sensitive to loss, but more streams hedge bet
- Circumventing fairness mechanism
  - 1 stream vs. n background: you get 1/(n+1)
  - X streams vs. n background: you get x/(n+x)
  - Example: 2 background, 1 stream: 1/3 = 33% of available resources
  - Example: 2 background, 8 streams: 8/10 = 80% of available resources

- There is a point of diminishing returns

- To get full TCP performance, the TCP window needs to be large enough to accommodate the Bandwidth Delay Product
• **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
    • 25ms = 20Mbps
    • 50ms = 10Mbps
    • 75ms = 6.67Mbps
    • 100ms = 5Mbps
  - **Autotuning** does help by growing the window when needed...
A small about of packet loss makes a huge difference in TCP performance

- A Nagios alert based on our regular throughput testing between one site and ESnet core alerted us to poor performance on high latency paths
- No errors or drops reported by routers on either side of problem link
  - only perfSONAR bwctl tests caught this problem
- Using packet filter counters, we saw 0.0046% loss in one direction
  - 1 packets out of 22000 packets
- Performance impact of this: (outbound/inbound)
  - To/from test host 1 ms RTT: 7.3 Gbps out / 9.8 Gbps in
  - To/from test host 11 ms RTT: 1 Gbps out / 9.5 Gbps in
  - To/from test host 51 ms RTT: 122 Mbps out / 7 Gbps in
  - To/from test host 88 ms RTT: 60 Mbps out / 5 Gbps in
    - More than 80 times slower!
The Metrics

• Use the correct tool for the Job
  – To determine the correct tool, maybe we need to start with what we want to accomplish ...

• What do we care about measuring?
  – Latency (Round Trip and One Way)
  – Jitter (Delay variation)
  – Packet Loss, Duplication, out-of-orderliness (transport layer)
  – Interface Utilization/Discards/Errors (network layer)
  – Achievable Bandwidth (e.g. “Throughput”)
  – Traveled Route
  – MTU Feedback
Latency

• Round Trip (e.g. source to destination, and back)
  – Hard to isolate the direction of a problem
  – Congestion and queuing can be masked in the final measurement
  – Can be done with a single ‘beacon’ (e.g. using ICMP responses)

• One Way (e.g. measure one direction of a transfer only)
  – Direction of a problem is implicit
  – Detects asymmetric behavior
  – See congestion or queuing in one direction first (normal behavior)
  – Requires ‘2 Ends’ to measure properly
Jitter

• To Quote Wikipedia: "undesired deviation from true periodicity"

• Computer people usually avoid the classic definition and (term) and use "packet delay variation" (PDV) instead

• In layman's terms:
  – Packet trains should be well spaced to aid in processing
  – Bursts can cause queuing on devices (followed by periods of inactivity)
  – Jitter is a calculation of this variation in distances between packets. High jitter indicates things are consistently not well spaced
• **Processing Delay**: Time to process a packet
• **Queuing Delay**: Time spent in ingress/egress queues to device
• **Transmission Delay**: Time needed to put the packet on the wire
• **Propagation Delay**: Time needed to travel on the wire
KanREN Monitoring – When Links Die

Packet loss on routing change

Backbone outage.
~10:30 AM on 3/9/2012

KSU -> WSU OWAMP

Early indication?
The term “throughput” 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

(Narrow Link)

(Shaded portion shows background traffic)
Outline

• Problem Definition & Motivation
• TCP & Metrics
• perfSONAR overview
• Case studies
• Site deployment recommendations
• perfSONAR host recommendations
• Wrap Up
Addressing the Problem: perfSONAR

• perfSONAR - an open, web-services-based framework for:
  – running network tests
  – collecting and publishing measurement results

• ESnet and Internet2 are:
  – Deploying the framework across the science community
  – Encouraging people to deploy ‘known good’ measurement points near domain boundaries
    • “known good” = hosts that are well configured, enough memory and CPU to drive the network, proper TCP tuning, clean path, etc.
  – Using the framework to find and correct soft network failures.
US Deployment

- Internet2
  - 4 Machines in each PoP on the current network (2 x Throughput Test Machine, 1 User Test Machine, 1 Latency Test Machine)
  - Plans for single server in all PoPs on new network
  - Internal Testing (http://owamp.net.internet2.edu), and 100s of community initiated tests per week
  - Central Netflow/SNMP Monitoring
  - Assistance available – rs@internet2.edu

- ESnet
  - 2 Machines in each PoP (Latency and Bandwidth Testing)
  - Machines at Customer sites (e.g. federal labs and other scientific points of interest)
  - Full mesh of testing (http://stats.es.net)
  - Assistance available – trouble@es.net
“Buzzwords” have a tendency to lose meaning when overused
- What does ‘perfSONAR’ mean?

Basic idea: Network Performance Matters
- Scientist moving data from a telescope to a lab
- Performers showing audio/video across the world

“Inter” Domain
- Solved science – every admin knows what goes on locally

“Intra” Domain
- Demarcation between networks houses a handoff that is may not be directly watched

“Multi” Domain
- The new normal – your closest collaborator is around the world
perfSONAR Overview – How To Use

• Deployments mean:
  – Instrumentation on a network
  – The ability for a user at location A to run tests to Z, and things “in the middle”
  – Toolkit deployment is the most important step for debugging, and enabling science

• Debugging:
  – End to end test
  – Divide and Conquer
  – Isolate good vs bad (e.g. who to ‘blame’)

Innocent cat

is shocked you blame him for mess. Shocked and hurt.
Global Reach of perfSONAR Monitoring
perfSONAR Architecture Overview

Data Services
- Measurement Points
- Measurement Archives
- Transformations

Information Services
- Service Lookup
- Topology
- Service Configuration
- Auth(n/z) Services

Infrastructure

Analysis/Visualization
- User GUIs
- Web Pages
- NOC Alarms
• PS-Toolkit includes these measurement tools:
  – BWCTL: network throughput
  – OWAMP: network loss, delay, and jitter
  – PINGER: network loss and delay
• Measurement Archives (data publication)
  – SNMP MA – Interface Data
  – pSB MA – Scheduled bandwidth and latency data
• Lookup Service
  – gLS – Global lookup service used to find services
  – hLS – Home lookup service for registering local perfSONAR metadata
• PS-Toolkit includes these Troubleshooting Tools
  – NDT (TCP analysis, duplex mismatch, etc.)
  – NPAD (TCP analysis, router queuing analysis, etc)
perfSONAR-PS Utility - Diagnostics

- The pS Performance Toolkit was designed for diagnostic use and regular monitoring
  - All tools preconfigured
  - Minimal installation requirements
  - Can deploy multiple instances for short periods of time in a domain
perfSONAR-PS Utility - Monitoring

- Regular monitoring is an important design consideration for perfSONAR-PS tools
  - perfSONAR-BUOY and PingER provide scheduling infrastructure to create regular latency and bandwidth tests
  - The SNMP MA integrates with COTS SNMP monitoring solutions
- The pSPT is capable of organizing and visualizing regularly scheduled tests
- NAGIOS can be integrated with perfSONAR-PS tools to facilitate alerting to potential network performance degradation
Outline

- Problem Definition & Motivation
- TCP & Metrics
- perfSONAR overview
- Case studies
- Site deployment recommendations
- perfSONAR host recommendations
- Wrap Up
Common Use Case

• Trouble ticket comes in:
  • “I’m getting terrible performance from site A to site B”
• If there is a perfSONAR node at each site border:
  – Run tests between perfSONAR nodes
    • performance is often clean
  – Run tests from end hosts to perfSONAR host at site border
    • Often find packet loss (using owamp tool)
    • If not, problem is often the host tuning or the disk

• If there is not a perfSONAR node at each site border
  • Try to get one deployed
  • Run tests to other nearby perfSONAR nodes
The following highlights a use of perfSONAR on Internet2 on 10/4/2012

- Latency Monitoring picked up application layer loss and increased jitter on a series of links
- Throughput Monitoring simulated a drop in available bandwidth on the same links
- Netflow Monitoring found an increase in discarded packets
- SNMP Monitoring picked up high utilization

Translation:

- High Use = Potential drops in service availability
- Required intervention to increase capacity and balance traffic
- Measurements picked up the underlying “reason” due to several metrics
perfSONAR Overview – Why To Use
perfSONAR Overview – Why To Use

![Bandwidth Graph](image)

Source: Seattle → Destination: Washington

<table>
<thead>
<tr>
<th></th>
<th>Source → Destination in Mbps</th>
<th>Destination → Source in Mbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max</td>
<td>974.23 Mbps</td>
<td>954.00 Mbps</td>
</tr>
<tr>
<td>Avg</td>
<td>972.79 Mbps</td>
<td>922.59 Mbps</td>
</tr>
<tr>
<td>Last</td>
<td>961.45 Mbps</td>
<td>44.70 Mbps</td>
</tr>
</tbody>
</table>
perfSONAR Overview – Why To Use

![Traffic Sentinel](https://inmon.internet2.edu/inmon/Figures/03-11-12/NTP-TrafficSentinel.png)

- **nms-rpsv-eth1.wash.net.internet2.edu > ge-2/3/0**
- 4 Oct, 02:12 – 4 Oct, 08:12, interval=2 min.

Copyright © 1999-2012 InMon Corp. ALL RIGHTS RESERVED

50 – 2/8/13, © 2012 Internet2 – zurawski@internet2.edu
perfSONAR Overview – Why To Use
Sample Results: Throughput tests

Heavily used path: probe traffic is “scavenger service”

Asymmetric Results: different TCP stacks?
REDnet Use Case – Host Tuning

• Host Configuration – spot when the TCP settings were tweaked...

• N.B. Example Taken from REDDnet (UMich to TACC, using BWCTL measurement)

• Host Tuning: http://fasterdata.es.net/fasterdata/host-tuning/linux/
Troubleshooting Example: Bulk Data Transfer between DOE SC Centers

• Users were having problems moving data between supercomputer centers, NERSC and ORNL
  – One user was: “waiting more than an entire workday for a 33 GB input file” (this should have taken < 15 min)
• perfSONAR-PS measurement tools were installed
  – Regularly scheduled measurements were started
• Numerous choke points were identified & corrected
  – Router tuning, host tuning, cluster file system tuning
• Dedicated wide-area transfer nodes were setup
  – Now moving 40 TB in less than 3 days
Outline

- Problem Definition & Motivation
- TCP & Metrics
- perfSONAR overview
- Case studies
- Site deployment recommendations
- perfSONAR host recommendations
- Wrap Up
perfSONAR-PS Software

- perfSONAR-PS is an open source implementation of the perfSONAR measurement infrastructure and protocols
  - written in the perl programming language
- [http://software.internet2.edu/pS-Performance_Toolkit/](http://software.internet2.edu/pS-Performance_Toolkit/)
- All products are available as RPMs.
- The perfSONAR-PS consortium supports CentOS (version 5 and 6).
- RPMs are compiled for i386 and x86 64
- Functionality on other platforms and architectures is possible, but not supported.
  - Should work: Red Hat Enterprise Linux and Scientific Linux (v5)
  - Harder, but possible:
    - Fedora Linux, SuSE, Debian Variants
There are two easy ways to deploy a perfSONAR-PS host

“Level 1” perfSONAR-PS install:
- Build a Linux machine as you normally would (configure TCP properly! See: http://fasterdata.es.net/TCP-tuning/)
- Go through the Level 1 HOWTO
  - http://fasterdata.es.net/ps_level1_howto.html
    - Includes bwctl.limits file to restrict to R&E networks only
  - Simple, fewer features, runs on a standard Linux build

Use the perfSONAR-PS Performance Toolkit netinstall CD
- Most of the configuration via Web GUI
  - http://psps.perfsonar.net/toolkit/
- Includes more features (perfSONAR level 3)
Placement of a tester should depend on two things:
- Where a tester will have the most positive of impacts for finding/preventing problems
- Where space/resources are available

We want to find certain sets of problems:
- Edge of your network to edge of your upstream provider
  - E.g. University to Regional
  - Regional to Backbone
- Core of your network to Edge of your network and upstream providers
  - Campus core facility to demarcation point
  - Campus core to ISP
- Location of important devices to remote facilities and points in between
  - Data centers to consumers of said data (e.g. campus to campus)
  - Data centers to ISP
Constructing Zones

• Networks are large and complex, but can be broken into a couple of common components:
  – Main Distribution Frame (MDF) where the WAN connectivity will land.
  – Intermediate Distribution Frames (IDF) in other buildings (major components on a LAN)
  – The Network “core” which may be data center that houses key components (Mail, DNS, HTTP, Telephony)
  – Population centers (Dorms, Offices, Labs, Data Centers)
Sample Site Deployment
Importance of Regular Testing

- You can’t wait for users to report problems and then fix them (soft failures can go unreported for years!)
- Things just break sometimes
  - Failing optics
  - Somebody messed around in a patch panel and kinked a fiber
  - Hardware goes bad
- Problems that get fixed have a way of coming back
  - System defaults come back after hardware/software upgrades
  - New employees may not know why the previous employee set things up a certain way and back out fixes
- Important to continually collect, archive, and alert on active throughput test results
Developing a Measurement Plan

• What are you going to measure?
  – Achievable bandwidth
    • 2-3 regional destinations
    • 4-8 important collaborators
    • 4-10 times per day to each destination
    • 20 second tests within a region, longer across the Atlantic or Pacific
  – Loss/Availability/Latency
    • OWAMP: ~10 collaborators over diverse paths
    • PingER: use to monitor paths to collaborators who don’t support owamp
  – Interface Utilization & Errors

• What are you going to do with the results?
  – NAGIOS Alerts
  – Reports to user community
  – Post to Website
### Status of perfSONAR Throughput Matrix

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0: atlas-npt2.bu.edu</td>
<td>-</td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>UNKNOWN</strong></td>
<td><strong>OK</strong></td>
</tr>
<tr>
<td>1: lhcb.fnal.gov</td>
<td><strong>OK</strong></td>
<td>-</td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>UNKNOWN</strong></td>
<td><strong>OK</strong></td>
</tr>
<tr>
<td>2: ps2.ochep.ou.edu</td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td>-</td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>UNKNOWN</strong></td>
<td><strong>OK</strong></td>
</tr>
<tr>
<td>3: psmsu02.aglt2.org</td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td>-</td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>UNKNOWN</strong></td>
<td><strong>OK</strong></td>
</tr>
<tr>
<td>4: netmon2.atlas-swt2.org</td>
<td><strong>OK</strong></td>
<td><strong>UNKNOWN</strong></td>
<td><strong>UNKNOWN</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td>-</td>
<td><strong>UNKNOWN</strong></td>
<td><strong>OK</strong></td>
</tr>
<tr>
<td>5: iut2-net2.iu.edu</td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td>-</td>
<td><strong>UNKNOWN</strong></td>
<td><strong>OK</strong></td>
</tr>
<tr>
<td>6: psnr-bw01.slac.stanford.edu</td>
<td><strong>OK</strong></td>
<td><strong>UNKNOWN</strong></td>
<td><strong>UNKNOWN</strong></td>
<td><strong>UNKNOWN</strong></td>
<td><strong>UNKNOWN</strong></td>
<td><strong>UNKNOWN</strong></td>
<td><strong>UNKNOWN</strong></td>
<td><strong>OK</strong></td>
<td>-</td>
</tr>
<tr>
<td>7: uct2-net2.uchicago.edu</td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td>-</td>
<td><strong>OK</strong></td>
</tr>
<tr>
<td>8: psum02.aglt2.org</td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>OK</strong></td>
<td><strong>UNKNOWN</strong></td>
<td><strong>OK</strong></td>
</tr>
</tbody>
</table>
Outline

• Problem Definition & Motivation
• TCP & Metrics
• perfSONAR overview
• Case studies
• Site deployment recommendations
• perfSONAR host recommendations
• Wrap Up
Host Considerations

- [http://psps.perfsonar.net/toolkit/hardware.html](http://psps.perfsonar.net/toolkit/hardware.html)

- Dedicated perfSONAR hardware is best
- Other applications will perturb results
- Separate hosts for throughput tests and latency/loss tests is preferred
  - Throughput tests can cause increased latency and loss
  - Latency tests on a throughput host are still useful however
- 1Gbps vs 10Gbps testers
  - There are a number of problem that only show up at speeds above 1Gbps
- Virtual Machines do not work well for perfSONAR hosts
  - clock sync issues
  - throughput is reduced significantly for 10G hosts
  - caveat: this has not been tested recently, and VM technology and motherboard technology has come a long way
Choosing hardware for a measurement node is not a complicated process

Some basic guidelines:

- Bare Metal (more on this later)
- x86 Architecture (64Bit is not natively supported in the software, but it can be emulated)
- “Modern” limits for RAM, CPU Speed, Main Storage
  - E.g. it doesn’t need to be brand new, but it should be no older than 8 years (e.g. we have evidence of old Pentium II desktop machines working, but not working well 😊)
  - Recycling is fine, unless you have money to burn on a new device (and who doesn’t!)
Use Cases - Latency

- A 10G card isn’t really need, 1G is recommended (100M would be ok as well, just be sure the driver is recent)
  - Be careful with TCP offload on some NICs, it can introduce OOP
- CPU load is minimal, single core single CPU is fine. Doesn’t need to be a whole lot of MHz/GHz
  - Multi-core/processor systems can sometimes introduce jitter on their own if interpret processing is not handled efficiently
- RAM is also minimal, enough to support a modern Linux distro (1G should be sufficient)
- Main Memory is where you do need some power. OWAMP Regular testing data can build up over time. Several G a month depending on who you are testing against.
  - This can be cleaned out if you are space constrained
  - We recommend 200G to be safe.
Use Cases - Bandwidth

- 1G is a common use case, but if you can do 10G aim for this
  - Same caveat about drivers – there are some nasty kernel/driver interactions stories out there ...
- CPU should be beefy, you do want a pretty good pentium/xeon on your side. Multi-cores/processors are not a requirement
- RAM should be consistent with the CPU, 2G+ is good
- The main memory requirements are not as great as the latency machine, 100G is more than enough.
Modern Server Class Hardware

- Internet2 uses Dell Power Edge 1950s (from 2005!) and these are still kicking
- I have been testing some Dell R310s lately. Pretty cost effective (EDU pricing of around $1.5k if you add on a 10G card and some LR optics)
- Supermicro makes a nice 1U/Half Size machine with an Atom processor. These are excellent for Latency testing (don’t push it with the bandwidth though)
Good Choices

- Desktop Towers
  - I don’t test these often, most are probably ok for temporary use cases.
  - “Energy Saving” models are a little suspect, these could reduce CPU power and effect the clock

- Laptops
  - I wouldn’t recommend this for longer term use, but for diagnostics they are mobile and effective
• Virtual Machines
  – Our largest concern is the clock
    • A VM gets its time updates from the Hypervisor
    • The HV gets updates via the system (hopefully it is running NTP)
    • If the VM is also running NTP, it will attempt to keep the clock stable, but the ‘backdoor’ updates to the VM clock from the HV will skip time forward/backward – confusing NTP
    • Think about what happens if the VM is swapped out ...
  – Situations where a VM is ok:
    • NDT/NPAD Beacon
    • 1G bandwidth testing
    • SNMP Collection, NAGIOS Operation
  – Situations where it is not:
    • OWAMP measurements
    • 10G Throughput
Poor Choices

- 1G host plugged into 100M Switch ... Pick out where we moved to a 1G Switch ...

![Graph showing network traffic distribution](image-url)
Poor Choices

• Mac Mini and similar micro-machines
  – Largest concern here is that the 1G NIC is on the motherboard, and competes for BUS resources.
    • This introduces jitter in latency measurements
    • Reduces throughput tests
  – Power management can be funky too
• Desktops/Laptops (for permanent placement)
  – Power management is a concern for aforementioned reasons
  – Onboard NICs are common here as well
Outline

• Problem Definition & Motivation
• TCP & Metrics
• perfSONAR overview
• Case studies
• Site deployment recommendations
• perfSONAR host recommendations
• Wrap Up
• Soft failures are everywhere
• We all need to look for them, and not wait for users to complain
• perfSONAR is MUCH more useful when its on every segment of the end-to-end path
• Ideally all networks and high BW end sites to deploy at least a “level 1” host
• 10G test hosts are needed to troubleshoot 10G problems

• perfSONAR is MUCH more useful when its open
• Locking it down behind firewalls/ACLs defeats the purpose
perfSONAR-PS Community

• perfSONAR-PS is working to build a strong user community to support the use and development of the software.

• perfSONAR-PS Mailing Lists
  – Announcement List:
    https://mail.internet2.edu/wws/subrequest/perfsonar-ps-announce
  – Users List:
    https://mail.internet2.edu/wws/subrequest/performance-node-users
  – Announcement List:
    https://mail.internet2.edu/wws/subrequest/performance-node-announce
The Way Forward - Training

- Network Performance Workshop
  - [http://www.internet2.edu/workshops/npw/](http://www.internet2.edu/workshops/npw/)
  - 15 over the last 2 years
  - 7 Affiliated with Internet2 events, 8 privately sponsored

- Structure
  - 1 or 2 Day training
  - Learn about the tools (perfSONAR), but more importantly how to use them in a campus/regional setting to solve real problems

- Contact Jason ([zurawski@internet2.edu](mailto:zurawski@internet2.edu)) if this sounds like something you want to host at your campus/regional

---

There has been a failure on the INTERNETS
Performance Measurement & Monitoring via perfSONAR

January 13\textsuperscript{th} 2013 – TIP2013: Building a Science DMZ
Jason Zurawski – Senior Research Engineer

For more information, visit \url{http://psps.perfsonar.net}