

perfSONAR Monitoring for LHCONE

Shawn McKee/University of Michigan

LHCONE Meeting

Ann Arbor, Michigan

September 15th, 2014

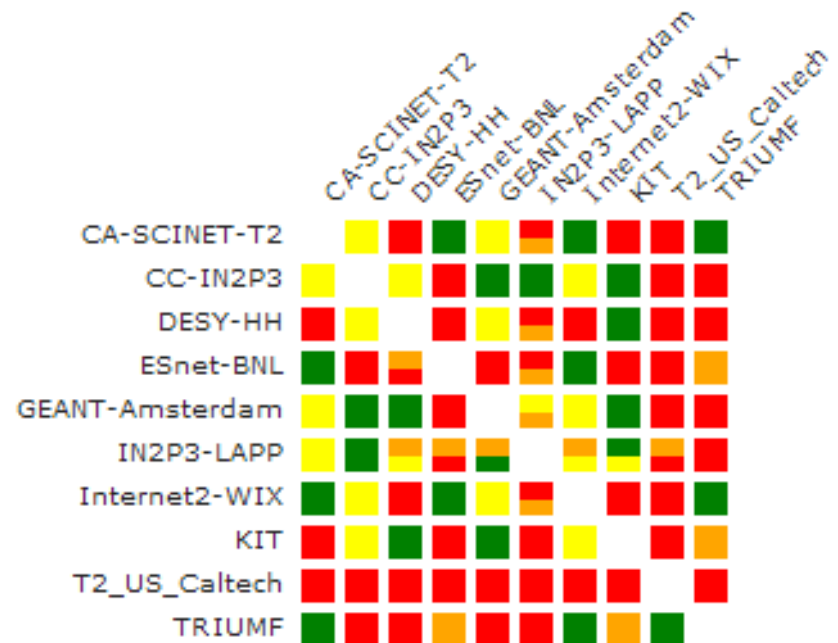
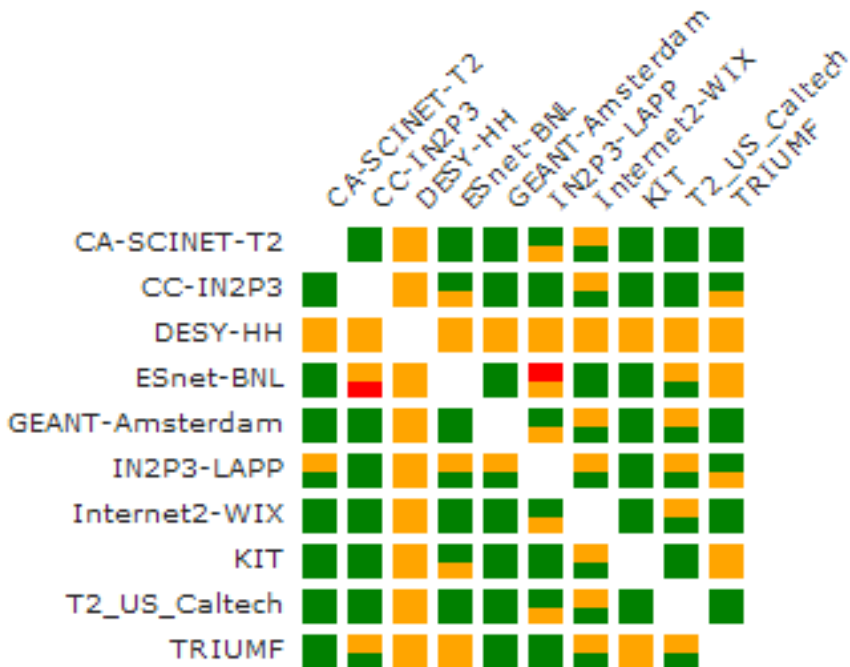
Overview of Talk

- ❄ Status of perfSONAR Monitoring for LHCONE
- ❄ Debugging Process To-date
- ❄ OSG Network Service
 - ❑ Overview of Datastore, OMD and mesh-configuration (Soichi)
 - ❑ OSG Subnet for monitoring on LHCONE??
- ❄ Discussion

LHCONE Network Matrices: 28Apr2014

OWAMP (Latency)

BWCTL (Bandwidth)



No packet loss, packet loss > 0.01

BW > 0.9 Gb, 0.5 < BW < 0.9 Gb, BW < 0.5 Gb

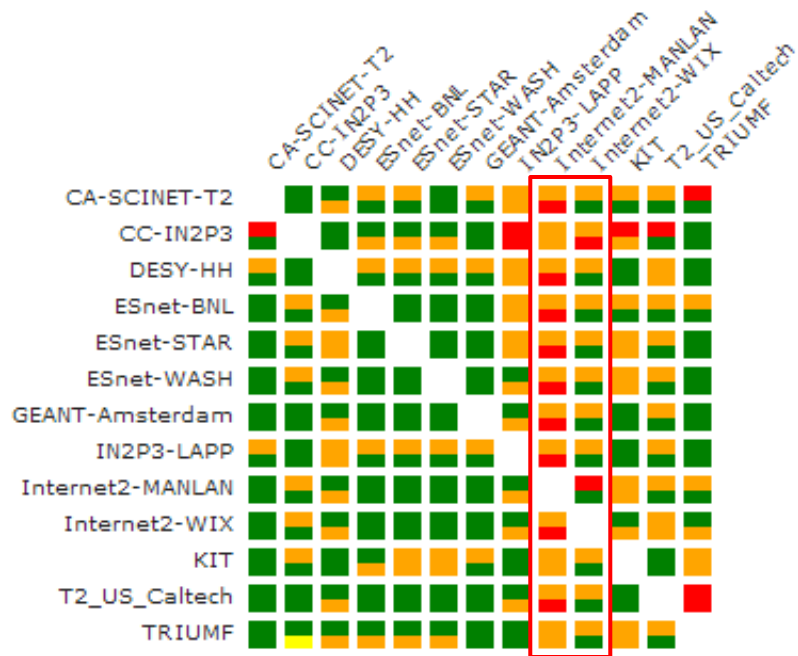
Main issue was too much “orange” indicating missing measurements/data

Sources are “row”, Destination is “column”

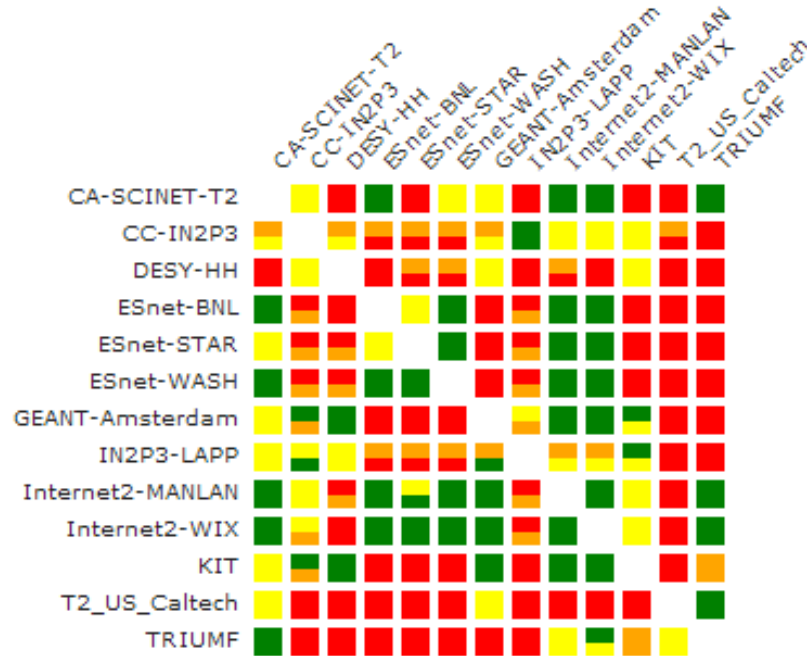
Each box split into two regions indicating where the test is run: top corresponds to “row”, bottom to “column”

LHCONE Network Matrices: 11 Aug 2014

OWAMP (Latency)



BWCTL (Bandwidth)



No packet loss, packet loss > 0.01

BW > 0.9 Gb, 0.5 < BW < 0.9 Gb, BW < 0.5 Gb

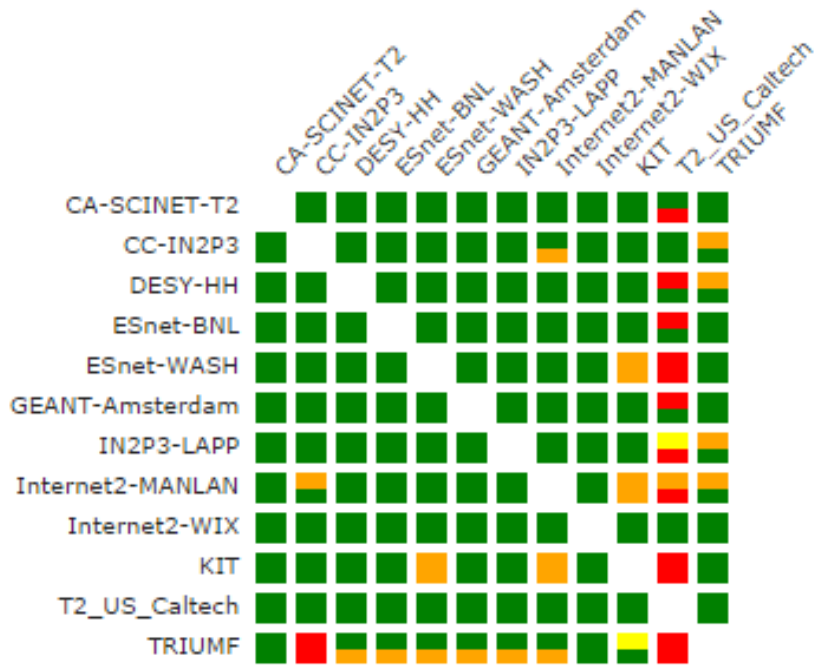
Main issue is STILL too much “orange” indicating missing measurements/data

Sources are “row”, Destination is “column”

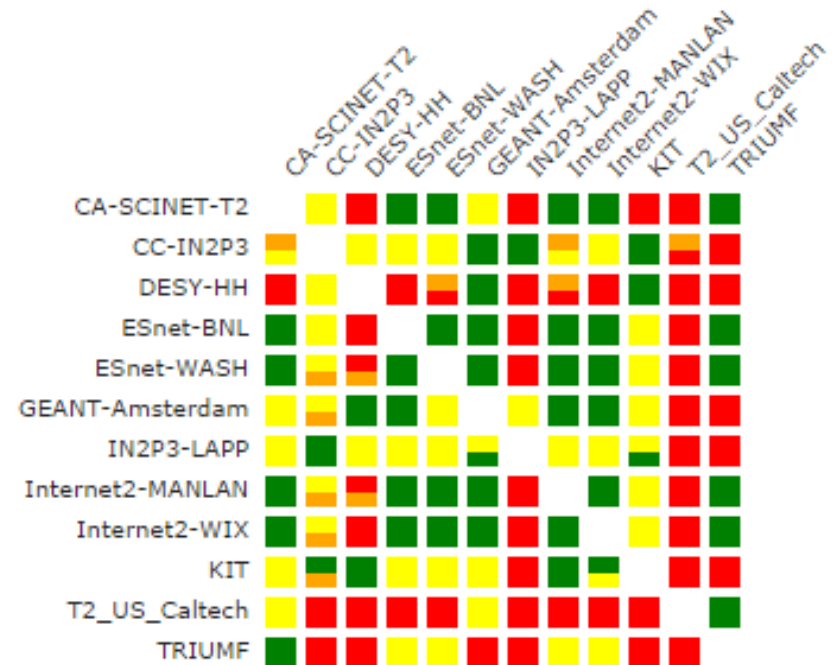
Each box split into two regions indicating where the test is run: top corresponds to “row”, bottom to “column”

LHCONE Network Matrices: 15Sep2014

OWAMP (Latency)



BWCTL (Bandwidth)



No packet loss, packet loss > 0.01

BW > 0.9 Gb, 0.5 < BW < 0.9 Gb, BW < 0.5 Gb

Improvements since the APAN meeting...mostly due to the work of Jason Zurawski (see later slides). Still a little orange remaining...some problems seem to be re-occurring after we have fixed them.

Also we have MOST of the needed people in the room now...can we fix the rest?

Monitoring the TA Link Outage

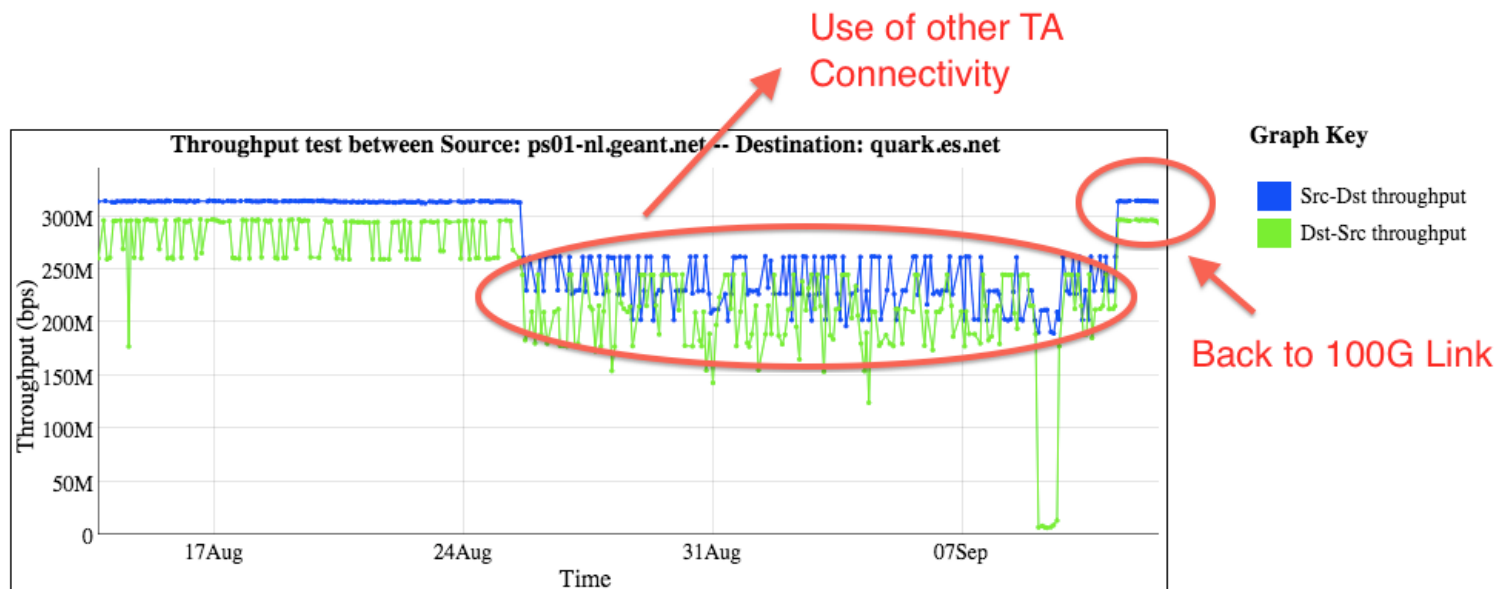
Slides from Jason Zurawski/ESnet



Monitoring the TA Link Outage

perfSONAR BWCTL Graph

perfSONAR



- ESnet (BNL) to GEANT (NL)
- Things to take away:
 - 100G headroom reduces congestion seen on 10G link(s)
 - Establishing a new steady state – for a 30 second, autotuned TCP test every 4 hours – should we ‘expect’ to see the 300 Mbps or should it be higher?
 - **Argument against:** should do better with 10G hosts and a 100G network (maybe?)
 - **Argument for:** 30 seconds isn’t enough for TCP ramp up – and window won’t grow to the needed amount (BDP = 120MB for 10G and 100ms).
 - Also note iperf is a 30 second **average**

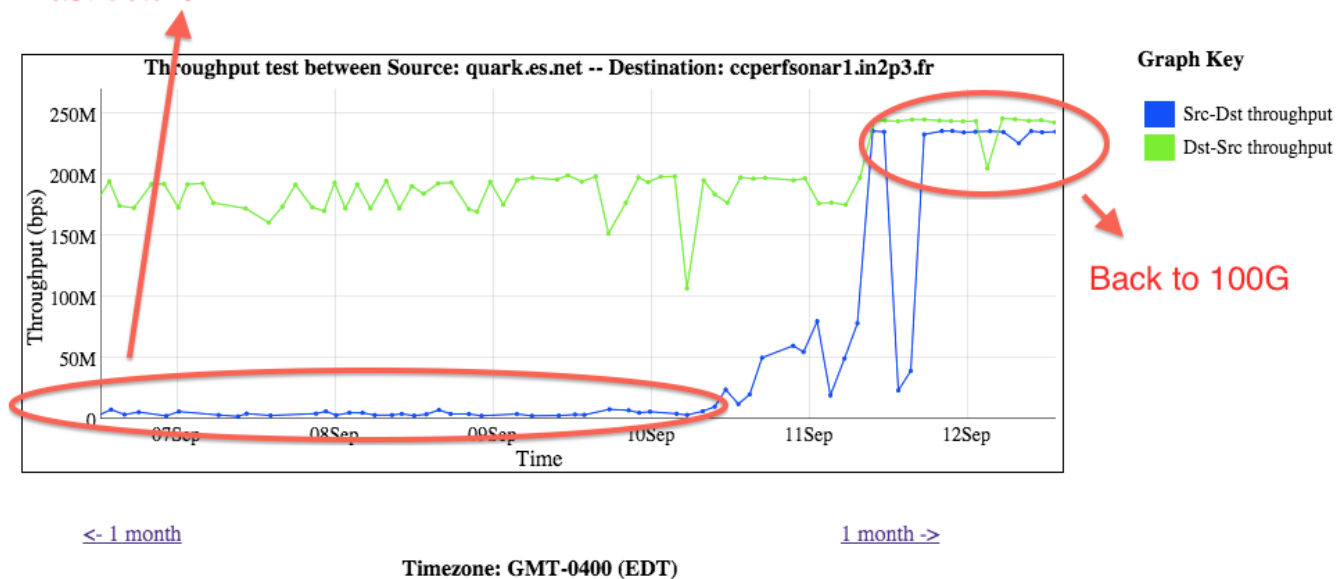


Monitoring the TA Link Outage

perfSONAR BWCTL Graph

perfSONAR

Really poor performance over 10G infrastructure?



- ESnet (BNL) to IN2P3
- Things to take away:
 - The 10G infrastructure is broken in places, has been for a while, and is not being fixed.
 - Another situation where we need to look at what is 'steady state' and figure out if it should be higher, lower, or if we need to set expectations in a different manner.

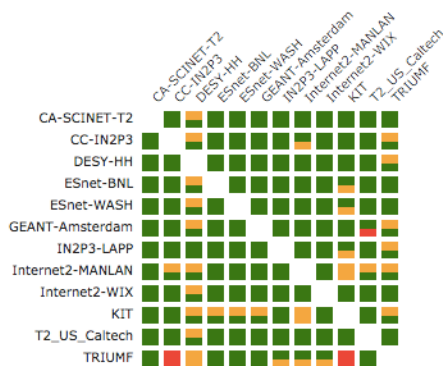
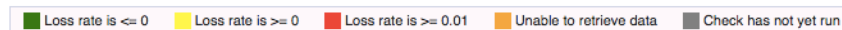


LHCONE Dashboard Slides

Review of OWAMP

- Got into a more 'green' state over the last 2 weeks

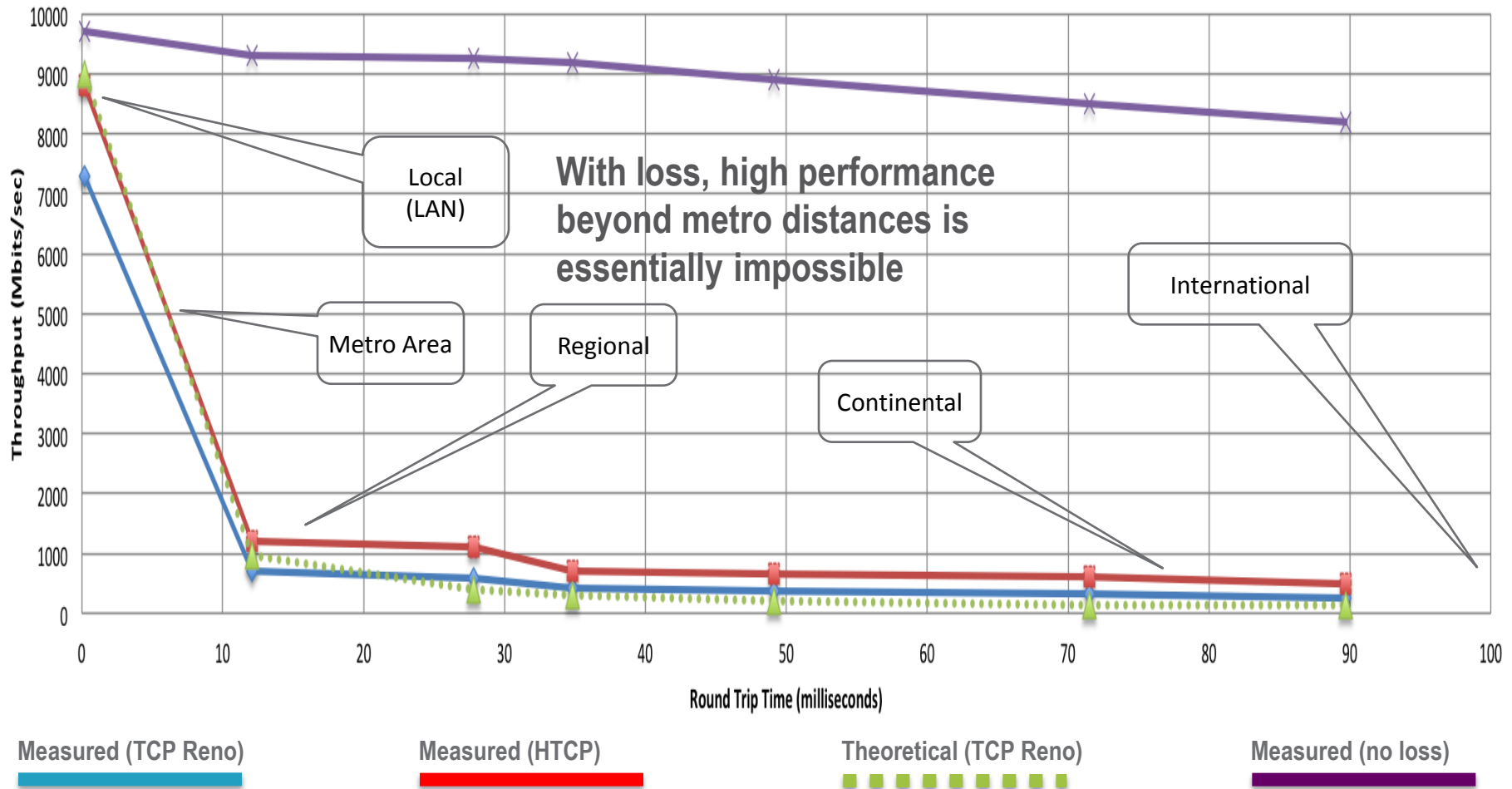
LHCONE testing sites - OWAMP Test Between LHCONE Latency Hosts



- Common problems:
 - Software not updated
 - Limits files were not allowing enough disk space/bandwidth for tests
 - Some sites didn't have the right meshes configured
 - Firewalls and port choices
- General statement – please visit the nodes at least once a week to check on them.
- 3.4 release will have things like yum auto-updates, and some other usability and maintenance features
- We see 'some' packet loss, not widespread or constant

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

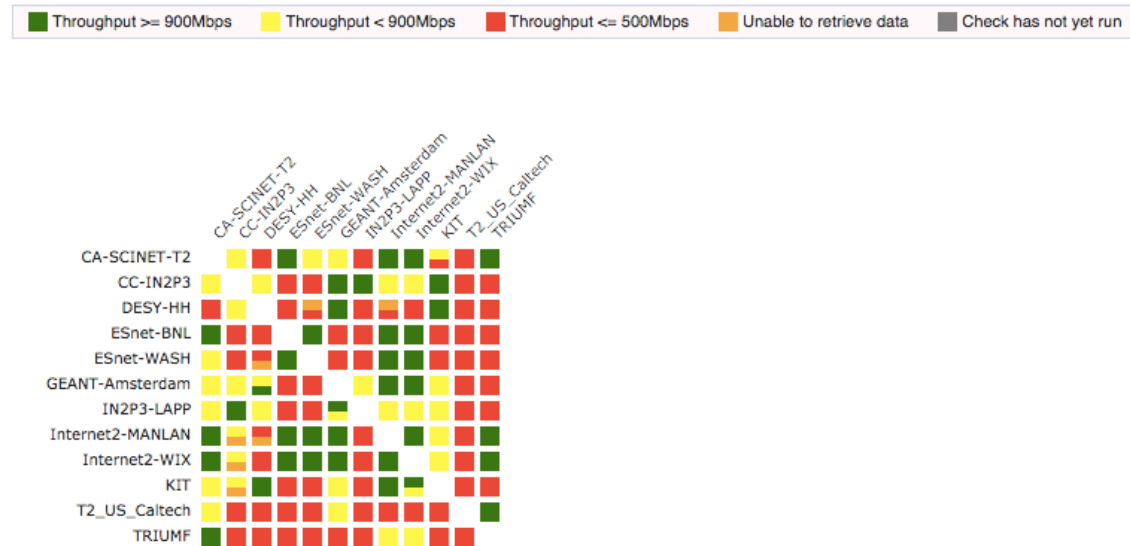
Throughput vs. Increasing Latency with .0046% Packet Loss



But ... is LHCONE Data dealing with Loss?

- A reminder, here is the BWCTL dashboard – with lots of red and yellow.
 - Red = $n < 500\text{Mbps}$
 - Yellow = $900\text{Mbps} > n \geq 500\text{Mbps}$

LHCONE testing sites - TCP BWCTL Test Between LHCONE Bandwidth Hosts



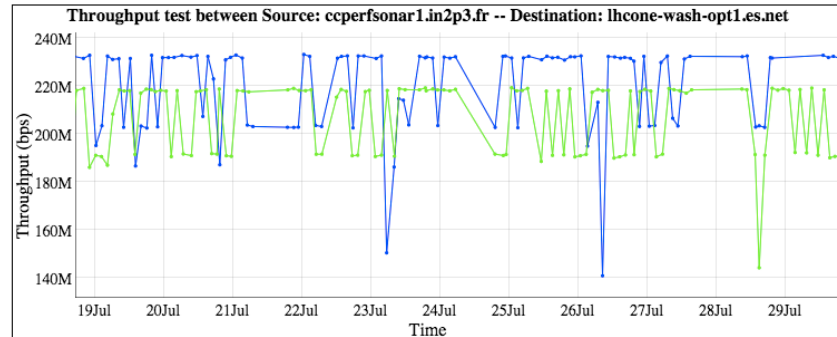
- Are we seeing massive performance problems **everywhere**, or do we need to adjust expectations?
- Recall the environment
 - Lots of latency between hosts
 - 10Gbps expectations – that implies lots of memory is needed per test
 - 100G headroom on the TA piece, mixture of 100G and 10G on the continents



Defining 'Steady' State

perfSONAR

perfSONAR BWCTL Graph



[<- 1 month](#)

[1 month ->](#)

Timezone: GMT-0400 (EDT)

- Should we call this 'normal'?
 - Pro:
 - Stable reading – its within a band of performance and rarely drops (occasional congestion events that are most likely local to one of the hosts).
 - This is not routine packet loss – that would be abysmal
 - Long (100ms) path
 - Single stream of TCP – e.g. very fragile.
 - Con:
 - Its not even 5% of the 'available' capacity between these hosts (assuming 10G bottleneck)
 - We 'could' do better if we manipulated other variables – should we be?



Reminder: Its All About the Buffers

- A prequel – The Bandwidth Delay Product
 - The amount of “in flight” data allowed for a TCP connection ($BDP = \text{bandwidth} * \text{round trip time}$)
 - Example: 10Gb/s TA Connection, ~100ms
 - $10,000,000,000 \text{ b/s} * .1 \text{ s} = 1,000,000,000 \text{ bits}$
 - $1,000,000,000 / 8 = 120,500,000 \text{ bytes}$
 - $120,500,000 \text{ bytes} / (1024 * 1024) \sim 120\text{MB}$
 - 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/>



perfSONAR Toolkit Defaults

- What is installed:

```
# increase Linux TCP buffer limits
net.core.rmem_max = 33554432
net.core.wmem_max = 33554432
# increase Linux autotuning TCP buffer limits
net.ipv4.tcp_rmem = 4096 87380 16777216
net.ipv4.tcp_wmem = 4096 87380 16777216
net.core.netdev_max_backlog = 30000
net.ipv4.tcp_no_metrics_save = 1
net.ipv4.tcp_congestion_control = htcp
```

- Why not 32/64/128/256 M?

- Purpose of pS node is to set expectations, not do the same job as a well tuned GridFTP server
- Single stream TCP testing is the common use case. Think about what would happen if we jacked the memory up:
 - Some LHC hosts are testing against 100s of other nodes. Consuming 128M or more ***PER TCP CONNECTION*** would cause the machines to run out of memory, quickly.
 - All tests – even ones that are ‘close’ and don’t need the BDP oomph, would use that max memory. Waste of resources on occasion



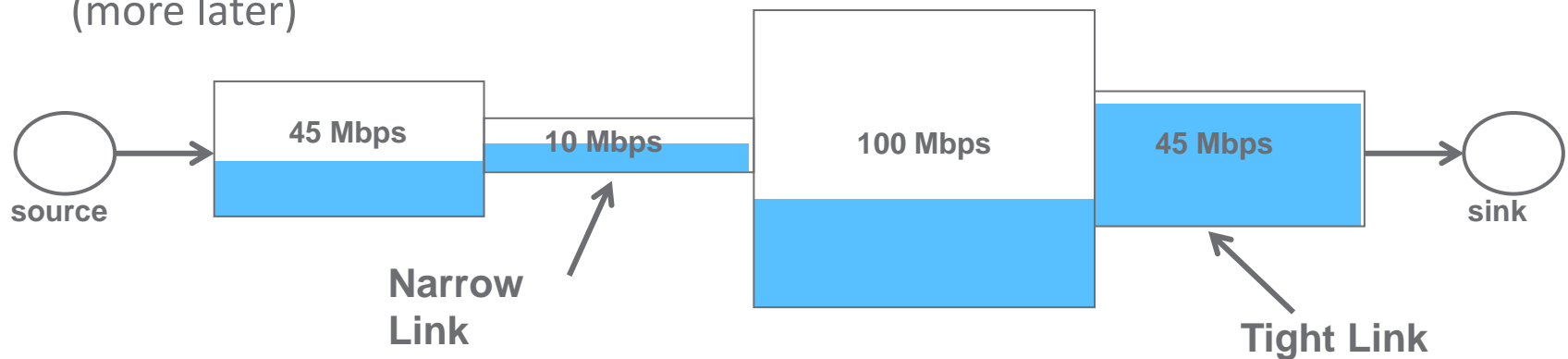
What BWCTL May Not be Telling Us

- 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 500Mbps 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). You need to learn what is expected
- 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.



Reminder: 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 Command Line 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 Happens When BWCTL Says “Crappy”

- Packet loss does this on a 60ms path ... it will hurt worse on a 100ms one:

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

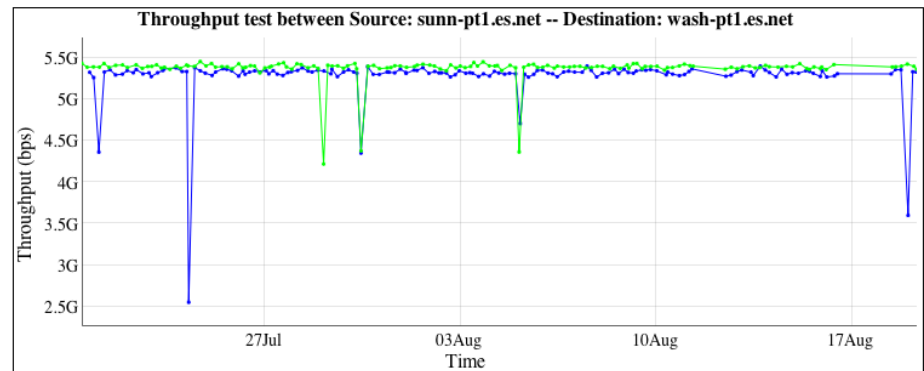
SENDER START
bwctl: exec_line: /usr/bin/nuttcp -vv -p 5004 -i 2.000000 -T 10 -t 198.129.254.58
bwctl: run_tool: tester: nuttcp
bwctl: run_tool: receiver: 198.129.254.58
bwctl: run_tool: sender: 198.124.238.34
bwctl: start_tool: 3598658394.807831
nuttcp-t: v7.1.6: socket
nuttcp-t: buflen=65536, nstream=1, 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.*

ESnet example – “should it should be higher?”

- 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, note that we ‘own’ all of it though
- If I ran a by-hand test, and cranked the parallel streams and window sizes – I could rock it at 9.9Gbps. Is that really the goal though?



<- 1 month

1 month ->

Timezone: GMT-0400 (EDT)



2nd example – “should it should be higher?”

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



[<- 1 month](#)

[1 month ->](#)

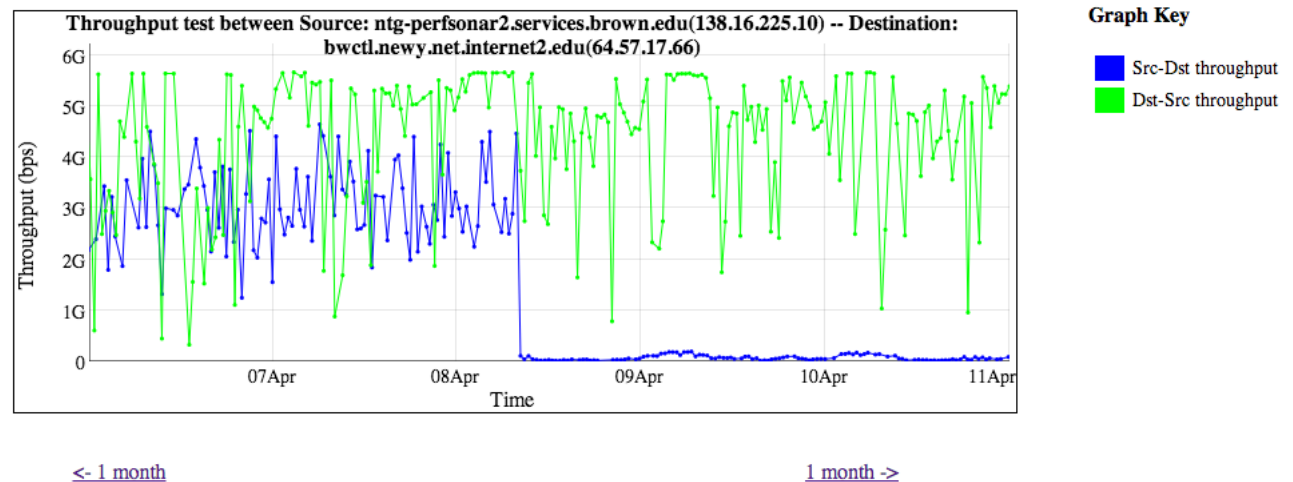
Timezone: GMT-0400 (EDT)

- Is it 5Gbps? No. Should it be? No! Could it be higher? Sure, run a different diagnostic test.
 - Long 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.
 - It’s a little more than 10% of the available capacity – but the data all adds up.



“Crappy” Has a Picture – And it Looks Like This

- 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)



Timezone: GMT-0400 (EDT)

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



Re-Setting Expectations

- Will we see 10Gbps on a regular test?
 - Sure, if we jack up the number of parallel streams, increase the sysctl setting for buffers, manually supply a window size, and run a 60 second test so we can get enough high values to improve the average
- Is that worth it?
 - No. Lets say that again - **NO**
- Alternatives:
 1. Slightly increase the window size of hosts at the sysctl level?
 2. Use set window sizes for tests in the mesh (this would be preferred to prior – can limit this to be used only in this mesh, and not others nodes participate in)
 3. Re-set the expectations in the dashboard (lower thresholds to define a better green vs. yellow vs. red)



OSG Network Service

- ❄ Open Science Grid (OSG) is deploying a network service for WLCG (and LHCONE). It consists of:
 - ❑ A datastore based upon Esmond (new MA in perfSONAR v3.4)
 - ❑ A GUI using MaDDash
 - ❑ A service monitoring component built on OMD
 - ❑ A “mesh-creation-configuration” utility built on registered information in OIM and GOCDB
- ❄ Demo on how the mesh-creation works (have to use slides for this since we need X509 credentials)

OIM / Mesh Config / Hostgroups

The screenshot shows a web browser window with the URL `https://oim-itb.grid.iu.edu/oim/meshconfig`. The page title is "Mesh Config Administrator". The navigation menu includes "Home", "Certificate", "Topology", "Downtimes", "Virtual Organizations", "Support Centers", "Campus Grids", "Projects", and "Soichi Hayashi".

The main content area has tabs for "Host Groups", "Parameter Sets", "Configs", and "Tests". The "Host Groups" tab is active, showing a form to "Add New Group".

The form includes the following fields:

- Service Type:** A dropdown menu with "net.perfSONAR.Bandwidth" selected.
- Group Name:** A text input field containing "USATLAS T1 BWCTL Hosts".
- Required:** A checkbox that is checked.
- OIM Resources:** A text input field containing "lhcmom-bnl" with a red 'X' icon.
- WLCG Resources:** A text input field containing "perf".

A dropdown menu is open over the "WLCG Resources" field, listing the following options:

- GRIF <perfsonar02.datagrid.cea.fr>
- CERN-PROD <perfsonar-ps.cern.ch>
- USC-LCG2 <perfsonar-ps-bandwidth.igfae.usc.es>
- IN2P3-LPSC <lpsc-perfsonar.in2p3.fr>
- DESY-ZN <perfon2.ifh.de>
- FZK-LCG2 <perfsonar-de-kit.gridka.de>
- IN2P3-LPC <drperf-bwctl.in2p3.fr>
- LRZ-LMU <lcz-lrz-perfs2.grid.lrz.de>
- CIEMAT-LCG2 <perfbw.ciemat.es>
- DESY-HH <perfsonar-ps-02.desy.de>
- IN2P3-IRES <sbgperfps2.in2p3.fr>
- UAM-LCG2 <perfsonar02.ft.uam.es>
- IN2P3-CPPM <marperf01.in2p3.fr>

Below the "WLCG Resources" field, there is another form section for "ATLAS T1 OWAMP Hosts" with a "Group Name" field and a "Required" checkbox.

OIM / Mesh Config / Parameters

The screenshot displays the Mesh Config Administrator web interface. The browser address bar shows the URL <https://oim-itb.grid.iu.edu/oim/meshconfig>. The navigation menu includes: OIM (ITB), Home, Certificate, Topology, Downtimes, Virtual Organizations, Support Centers, Campus Grids, Projects, and Soichi Hayashi.

The main content area is titled "Mesh Config Administrator" and has four tabs: Host Groups, Parameter Sets, Configs, and Tests. The "Parameter Sets" tab is active.

There are two parameter set configuration forms visible:

- Top Form:**
 - Service Type: `net.perfSONAR.Latency` (Required)
 - Parameter Set Name: `OWAMP Parameters` (Required)
 - Required Parameters (JSON):

```
{
  "packet_padding": "0",
  "loss_threshold": "10",
  "sample_count": "300",
  "session_count": "18000",
  "bucket_width": "0.001",
  "packet_interval": "0.1",
  "force_bidirectional": "1",
  "type": "perfsonarbuoyVowamp"}
```

(Required)
 - Buttons: "Add New Parameter Set" and a red "X" icon.
- Bottom Form:**
 - Service Type: `net.perfSONAR.Bandwidth` (Required)
 - Parameter Set Name: `BWCTL Parameters` (Required)
 - Buttons: A red "X" icon.

OIM / Mesh Config / Configs

The screenshot shows a web browser window with two tabs: 'Perfsonar Mesh Configura' and 'OIM (ITB) (Debug)'. The address bar shows the URL <https://oim-itb.grid.iu.edu/oim/meshconfig>. The page title is 'Mesh Config Administrator'. The navigation menu includes 'Home', 'Certificate', 'Topology', 'Downtimes', 'Virtual Organizations', 'Support Centers', 'Campus Grids', 'Projects', and 'Soichi Hayashi'. The main content area has tabs for 'Host Groups', 'Parameter Sets', 'Configs', and 'Tests'. The 'Configs' tab is active, showing a form for adding a new configuration file. The form includes fields for 'Name' (us-atlas), 'Description' (USATLAS Mesh Config), 'Submitter Contact' (Soichi Hayashi <hayashis@iu.edu>), and 'Administrative Contact' (Shawn McKee <smckee@umich.edu>). There is also a 'Disable' checkbox. A second configuration entry for 'us-cms' is partially visible at the bottom.

Mesh Config Administrator

Host Groups Parameter Sets **Configs** Tests

Add New Configuration File

Name * Required

Description * Required

Contact Information

Submitter Contact

Primary

Administrative Contact

Primary

Secondary

Disable

Name * Required

Description * Required

Mesh Config Adding Tests

Perfsonar Mesh Configura x OIM (ITB) (Debug) x

https://oim-itb.grid.iu.edu/oim/meshconfig

OIM (ITB) Home Certificate Topology Downtimes Virtual Organizations Support Centers Campus Grids Projects Soichi Hayashi

Mesh Config Administrator

Host Groups Parameter Sets Configs Tests

Configuration to be part of ✖
us-atlas * Required

Service Type ✖ Name ✖
net.perfSONAR.Latency USATLAS OWAMP Mesh Test * Required

Required Parameters ✖
OWAMP Parameters * Required

Required Mesh Type ✖ Host Group A ✖
MESH USATLAS T1 OWAMP Hosts ✖ USATLAS T2 OWAMP Hosts ✖

Disable

Configuration to be part of ✖
us-atlas * Required

Add New Test

Mesh Types

DISJOINT
TODO - add network diagram depicting disjoint mesh

MESH
TODO - add network diagram depicting mesh

STAR
TODO - add network diagram depicting star

MyOSG / Mesh Config

The screenshot shows a web browser window with two tabs: "Perfsonar Mesh Configura" and "OIM (ITB) (Debug)". The address bar shows the URL: `myosg-itb.grid.iu.edu/miscpfmesh?count_sg_1&count_active=on&count_enabled=on`. The page header includes a navigation menu with items like "MyOSG ITB Instance", "Home", "Resource Group", "Virtual Organization", "Status Map", and "Misc", along with a "Search OSG" field and a "Login" link.

A light blue warning box at the top states: "This is a ITB instance. This URL could disappear anytime or maybe unstable. For production use, please use <http://myosg.grid.iu.edu>".

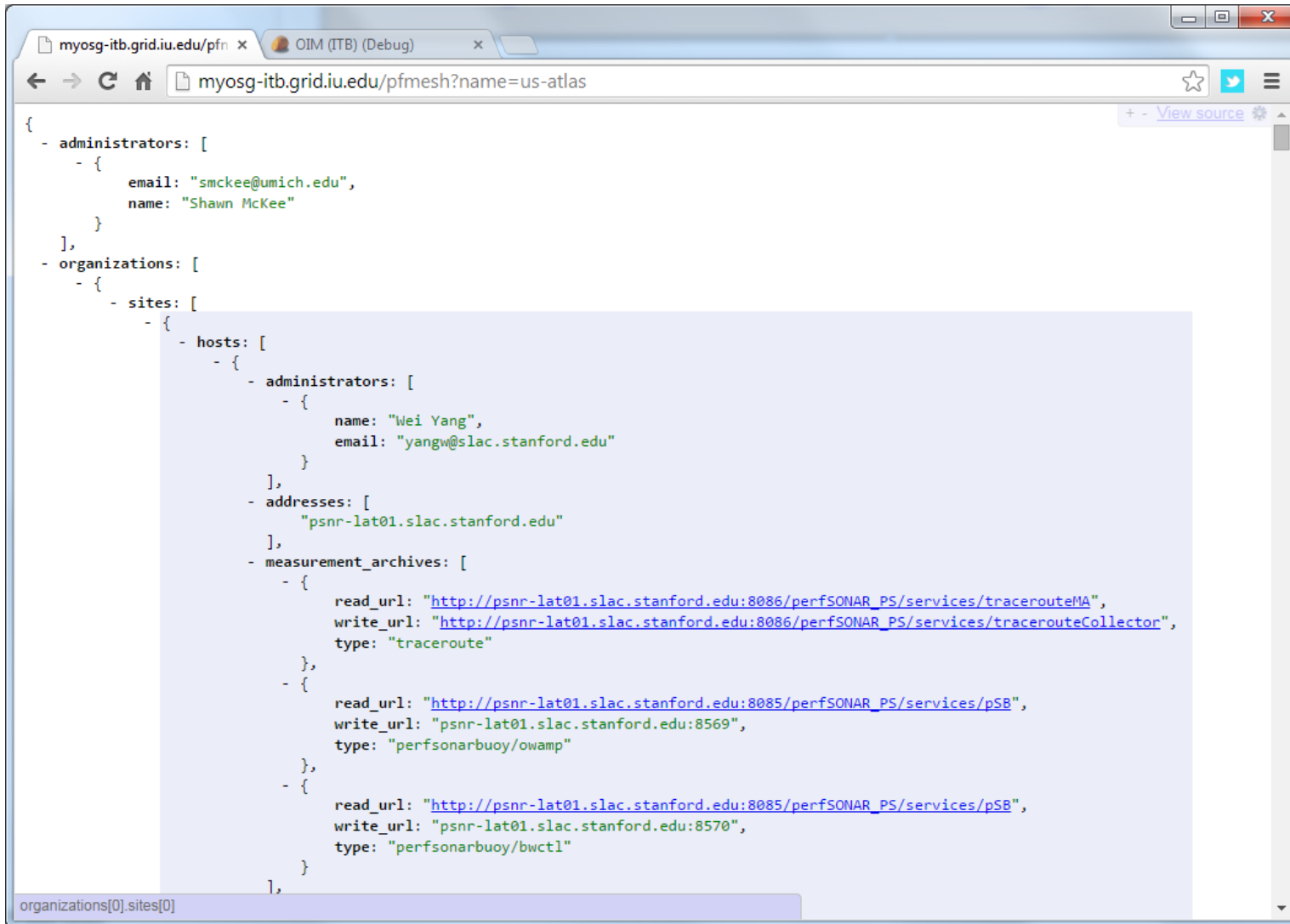
The main content area is titled "Perfsonar Mesh Configurations" and contains a table with the following data:

USATLAS Mesh Config	http://myosg-itb.grid.iu.edu/pfmesh?name=us-atlas
USCMS Mesh Config	http://myosg-itb.grid.iu.edu/pfmesh?name=us-cms
ALL WLCG Endpoints	http://myosg-itb.grid.iu.edu/pfmesh?name=wlcg-all

On the right side, there is an "Update Page" button and a section titled "INFORMATION TO DISPLAY" with a dropdown menu set to "Perfsonar Mesh Configuratio". Below this, there is a "Subscribe to the content of this page." section with links for "Netvibes", "Mobile Content", "XML", and "CSV". A "Create TinyURL" button is also present. At the bottom of the right sidebar, it says "Open [GOC Ticket](#) for assistance."

At the bottom of the page, there is a footer with the text: "MyOSG ITB Instance Version 2.27 | [Report Bugs](#) | [Privacy Policy](#)" and "Copyright 2013 - The Trustees of Indiana University - Developed for Open Science Grid".

MyOSG / Mesh Config (us-atlas)



The screenshot shows a web browser window with the address bar displaying `mysosg-itb.grid.iu.edu/pfmesh?name=us-atlas`. The page content is a JSON configuration file. The visible portion of the JSON is as follows:

```
{
- administrators: [
  - {
    email: "smckee@umich.edu",
    name: "Shawn McKee"
  }
],
- organizations: [
  - {
    - sites: [
      - {
        - hosts: [
          - {
            - administrators: [
              - {
                name: "Wei Yang",
                email: "yangw@slac.stanford.edu"
              }
            ],
            - addresses: [
              "psnr-lat01.slac.stanford.edu"
            ],
            - measurement_archives: [
              - {
                read_url: "http://psnr-lat01.slac.stanford.edu:8086/perfSONAR_PS/services/tracerouteMA",
                write_url: "http://psnr-lat01.slac.stanford.edu:8086/perfSONAR_PS/services/tracerouteCollector",
                type: "traceroute"
              },
              - {
                read_url: "http://psnr-lat01.slac.stanford.edu:8085/perfSONAR_PS/services/pSB",
                write_url: "psnr-lat01.slac.stanford.edu:8569",
                type: "perfsonarbuoy/owamp"
              },
              - {
                read_url: "http://psnr-lat01.slac.stanford.edu:8085/perfSONAR_PS/services/pSB",
                write_url: "psnr-lat01.slac.stanford.edu:8570",
                type: "perfsonarbuoy/bwctl"
              }
            ]
          }
        ]
      }
    ]
  }
]
```

The browser's status bar at the bottom left shows the path `organizations[0].sites[0]`.

Issue for LHCONE Monitoring

- ❄ OSG has assigned a subnet for LHC related monitoring and the network service components:
 - ❑ 129.79.53.0/24
- ❄ Right now OSG is not participating in LHCONE...but if we want it to host the “production” monitoring services for WLCG and LHCONE it needs to.
 - ❑ What is the best way forward? Can we get this subnet “joined” (peering) with LHCONE? Is this within the AUP?
- ❄ Discussion after next slide...

Discussion/Questions/Comments?

There is a lot to consider.

I hope we have time for questions, discussion and comments.

Useful URLs

- ❄ LHCOPN instructions for perfSONAR-PS (out-of-date):
 - ❑ <https://twiki.cern.ch/twiki/bin/view/LHCOPN/PerfsonarPS>
- ❄ LHCONE “initial” monitoring setup page
 - ❑ <https://twiki.cern.ch/twiki/bin/view/LCG/PerfsonarDeployment>
- ❄ Open Science Grid Networking URL
 - ❑ <https://www.opensciencegrid.org/bin/view/Documentation/NetworkingInOSG>
- ❄ perfSONAR tools, tips and best practices
 - ❑ <http://www.usatlas.bnl.gov/twiki/bin/view/Projects/LHCperfSONAR>
- ❄ MaDDash Monitoring
 - ❑ <http://maddash.aglt2.org/maddash-webui/index.cgi?dashboard=LHCONE%20testing%20sites>
- ❄ OMD Monitoring
 - ❑ https://maddash.aglt2.org/WLCGperfSONAR/check_mk/index.py?start_url=%2FWLCGperfSONAR%2Fcheck_mk%2Fview.py%3Fview_name%3Dhostgroups

OMD for LHCONE perfSONAR-PS

http://maddash.aglt2.org/WLCGperfSONAR/check_mk

OMD (Open Monitoring Distribution) wraps a set of Nagios packages into a single pre-configured RPM

User WLCGs

Pw at meeting ☺

The screenshot displays the Check_MK web interface. On the left, there is a sidebar with navigation options under 'Views', including Dashboards, Hosts, Hostgroups, Services, and Servicegroups. The main content area shows a 'Tactical Overview' with 207 hosts and 8 unhandled problems, and 2404 services with 538 unhandled problems. Below this is a 'Quicksearch' field and a 'Views' menu. The main table displays the 'Hostgroup LHCONE perfSONAR-PS Toolkit nodes' with columns for state, host, icons, alias, and performance metrics (OK, Wa, Un, Cr, Pd).

state	Host	Icons	Alias	OK	Wa	Un	Cr	Pd
UP	ccperfsonar1.in2p3.fr		ccperfsonar1.in2p3.fr	7	0	0	2	0
UP	ccperfsonar2.in2p3.fr		ccperfsonar2.in2p3.fr	8	0	0	0	0
UP	lapp-ps01.in2p3.fr		lapp-ps01.in2p3.fr	7	0	0	2	0
UP	lapp-ps02.in2p3.fr		lapp-ps02.in2p3.fr	8	0	0	0	0
UP	lhcone-star-opt1.es.net		lhcone-star-opt1.es.net	6	0	0	0	0
UP	lhcone-wash-opt1.es.net		lhcone-wash-opt1.es.net	6	0	0	0	0
UP	lhcone.test.manlan.internet2.edu		lhcone.test.manlan.internet2.edu	10	0	0	0	0
UP	lhcone.test.wix.internet2.edu		lhcone.test.wix.internet2.edu	10	0	0	0	0
UP	perfsonar-de-kit.gridka.de		perfsonar-de-kit.gridka.de	9	0	0	0	0
UP	perfsonar-ps-01.desy.de		perfsonar-ps-01.desy.de	8	0	0	0	0
UP	perfsonar-ps-02.desy.de		perfsonar-ps-02.desy.de	9	0	0	0	0
UP	perfsonar.ultralight.org		perfsonar.ultralight.org	9	0	0	0	0
UP	perfsonar2-de-kit.gridka.de		perfsonar2-de-kit.gridka.de	8	0	0	0	0
UP	perfsonar2.ultralight.org		perfsonar2.ultralight.org	8	0	0	0	0
UP	ps-bandwidth.lhcmon.triumf.ca		ps-bandwidth.lhcmon.triumf.ca	9	0	0	0	0
UP	ps-bandwidth.scinet.utoronto.ca		ps-bandwidth.scinet.utoronto.ca	9	0	0	0	0
UP	ps-latency.lhcmon.triumf.ca		ps-latency.lhcmon.triumf.ca	8	0	0	0	0
UP	ps-latency.scinet.utoronto.ca		ps-latency.scinet.utoronto.ca	8	0	0	0	0
UP	ps01-nl.geant.net		ps01-nl.geant.net	11	0	0	0	0
UP	quark.es.net		quark.es.net	6	0	0	0	0