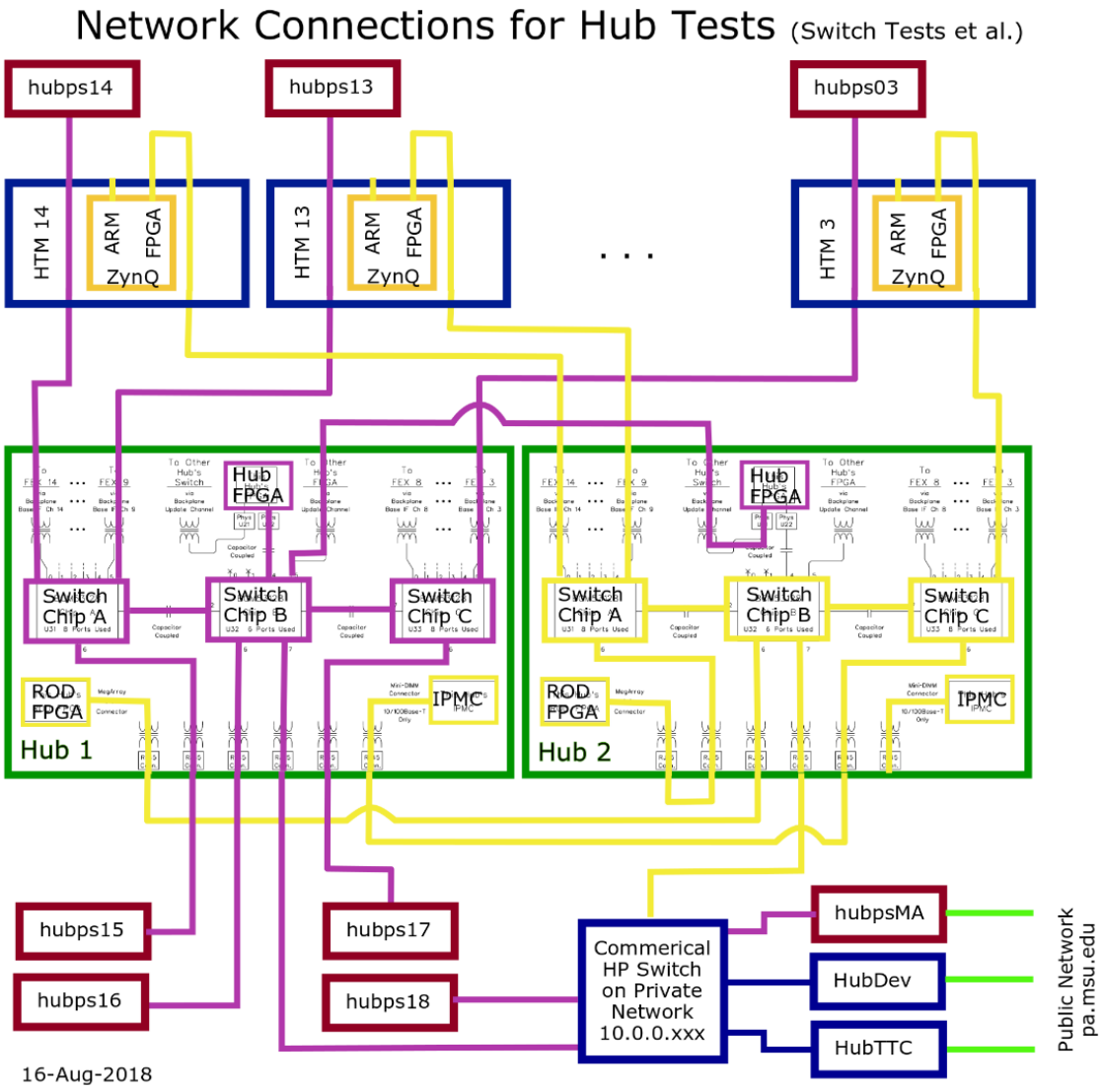
Testing Setup:

The current testing setup for the Hub GbE switch test consists of 17 computers running CentOS7 with perfSONAR-toolkit installed. PerfSONAR is a network measurement toolkit developed by cooperation between several groups including ESNet, Internet2, the University of Michigan, and Indiana University. Most tests ran and measurements made will be done using perfSONAR. More information regarding perfSONAR can be found at https://www.perfsonar.net/

16 of the computers, hubps03-18 function as testing nodes, and each one is associated with a particular port of the Hub's GbE switch. Hubps03-14 connect by ethernet to a front panel ethernet port on HTM cards 3-14, which in turn connect to the Hub's backplane ports. Hubps15, hubps16, and hubps17 connect to three of the Hub's front panel ethernet ports. Hubps18 connects to a commercial GbE switch, which in turn is connected to an ethernet port on the Hub's front panel. This additional switch is used so that all testing nodes can be reached by the remaining computer, hubpsMA. HubpsMA acts as both a measurement archive that stores all test results, and as a coordinator that schedules tests between nodes. To accomplish this, hubpsMA has perfSONAR-centralmanagement in addition to perfSONAR-toolkit.



Testing Plan:

There are two main aspects of the GbE switch that we wish to test: that it truly functions at gigabit speeds, and that under stressful conditions there is no/minimal packet loss. To that end, there are two tests we perform: a throughput test, and a multi-directional OWAMP test.

Throughput Test:

The throughput test is meant to test the ability of each port on the switch to send and receive data at gigabit speeds. Testing of a commercial HP GbE switch showed that it sent and received data at a rate of ~935 Mbits/sec, so we use this as our benchmark for the Hub switch. That is, we consider a throughput in the realm of 900 Mbits/sec or higher acceptable.

The throughput measurement is performed using iperf3, a tool for measuring the maximum achievable bandwidth of a network. Iperf3 was developed by ESNet and Lawrence Berkeley National Laboratory. More information on iperf3 can be found at https://iperf.fr/. Using this tool, one can measure the throughput between two nodes over a period of time specified by the user. Using pScheduler, a tool included in perfSONAR-toolkit, one can schedule for a throughput measurement to be performed between two nodes using iperf3, with options for storing the results in a measurement archive and scheduling for the measurement to be performed at regular intervals or at certain times.

We have a script written that performs a throughput measurement between one or several source nodes against every other node, as well as the reverse direction, to find the throughput of each node on the switch. The script then organizes the results of these measurements into a concise and more easily readable log file, which includes a report of whether each node meets the minimum throughput requirements specified by the user. The raw output of measurements is also stored in a seperate and much longer log file. The throughput requirements, as well as various other options, are specified as command line arguments when calling the script. In the near future, I plan to also allow for these specifications to be made in a configuration file that is specified when calling the script.

Early Results:

I ran a throughput test that measured the throughput between hubps15 and all other test nodes for a duration of twenty minutes in each direction. The results showed all ports to send and receive data at a rate of above 910 Mbits/sec. More tests with longer durations will follow in the future.

```
################################################################################
#                Throughput Test for  Dec 5, 2018                              #
#                                                                              #
################################################################################

================================================================================
 Test Results for node hubps03
================================================================================

Test Results for connection from hubps15 to hubps03

The schedules on all hosts are clear


Time initiated: Wed Dec  5 18:52:42 2018
The average throughput from hubps15 to hubps03 was: 912.9 Mbits/s.
The minimum throughput from hubps15 to hubps03 was: 910.2 Mbits/s.
The maximum throughput from hubps15 to hubps03 was: 920.6 Mbits/s.

The minimum throughput is above 900 Mbits/sec and the maximum throughput is below 1000 Mbits/sec.

The path from hubps15 to hubps03 meets the throughput requirements

Summary from raw JSON output:
{
   "throughput-bits": 912881551.0451915,
   "start": 0,
   "end": 1199.9999690055847,
   "throughput-bytes": 136932229120,
   "retransmits": 0
}

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------

Test Results for connection from hubps03 to hubps15

The schedules on all hosts are clear


Time initiated: Wed Dec  5 19:13:24 2018
The average throughput from hubps03 to hubps15 was: 912.5 Mbits/s.
The minimum throughput from hubps03 to hubps15 was: 910.2 Mbits/s.
The maximum throughput from hubps03 to hubps15 was: 916.5 Mbits/s.

The minimum throughput is above 900 Mbits/sec and the maximum throughput is below 1000 Mbits/sec.

The path from hubps03 to hubps15 meets the throughput requirements

Summary from raw JSON output:
{
   "throughput-bits": 912470800.2107097,
   "start": 0,
   "end": 1200.0000460147858,
   "throughput-bytes": 136870625280,
   "retransmits": 0
}

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------

================================================================================
```

Illustration 1: Sample from output log of throughput test.

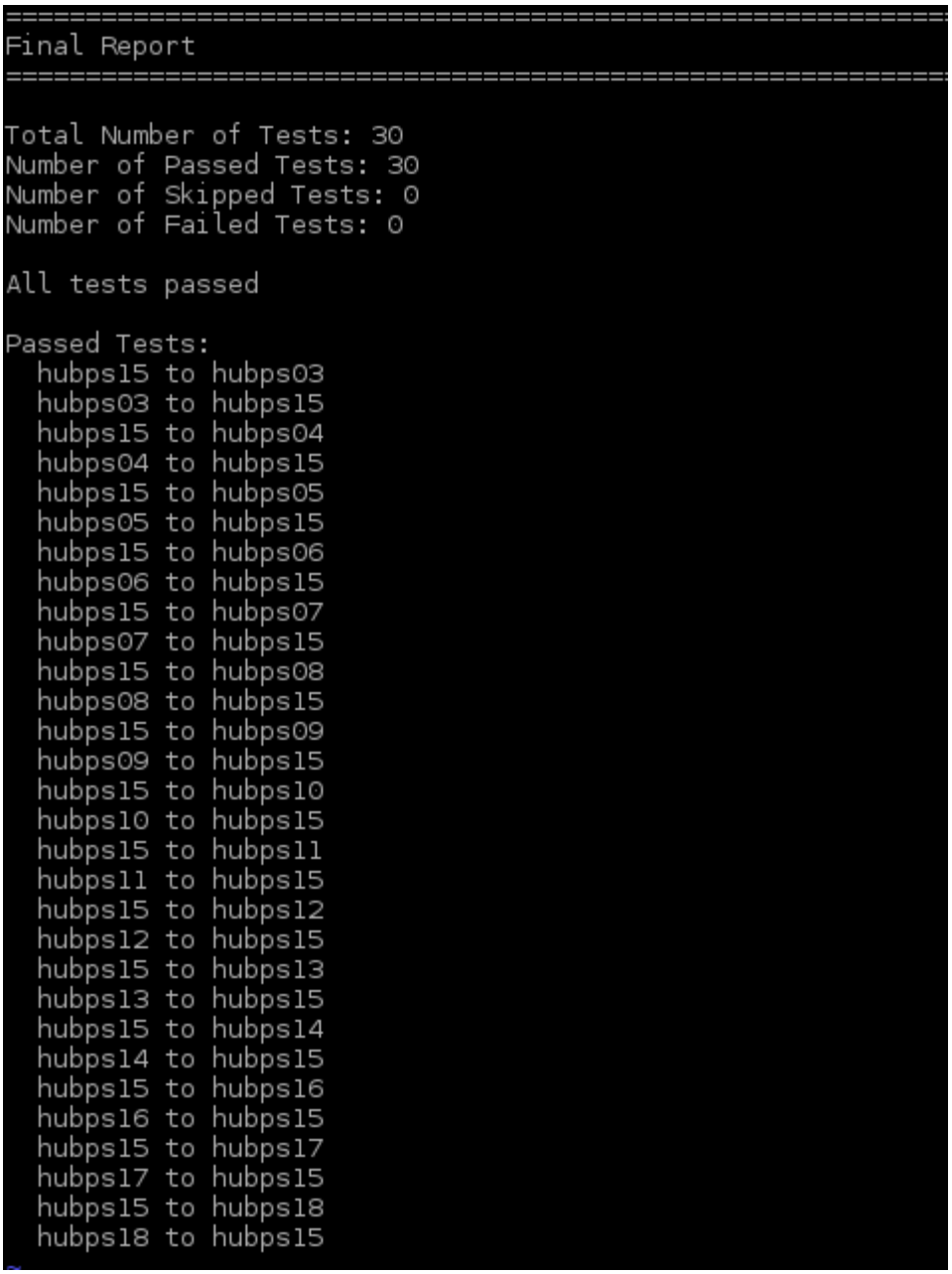


Illustration 2: Summary Report from throughput log.

Multisource OWAMP Test:

The multisource owamp test is meant to test that the GbE switch does not drop any packets when data is being sent and received from many different ports on the switch. To that end, we simultaneously run one-way ping measurements between all testing nodes. We do one-way ping so that we can tell specifically if data is being lost when being sent or received form a node. This one-way ping measurement is done using the One-Way Active Measurement Protocol, or OWAMP, tool included in perfSONAR-toolkit. To schedule all these measurements, we use perfSONAR's Meshconfig component (replaced by pSConfig as of version 4.1). This allows one to publish a configuration file in a public location that test nodes can read and use to schedule measurements between each other.

The tests are currently configured so that each test node initiates a forward and reverse test between each other node, for a total of 16*2*15 = 480 tests running simultaneously. In each test, 10 1450-byte packets (not including packet header) are sent every second, for a total of 4800 packets sent each second and 6.96 Mbytes/sec = 55.68 Mbits/sec. There is also some additional traffic when each test node sends test results back to the measurement archive.

Since there are many tests to keep track of, we specify in our mesh configuration file for all test nodes to send their test results to a measurement archive on hubpsMA. We then use the MaDDash tool included in perfSONAR-centralmanagement to create webpage containing a grid that shows the status of each test. We specify three thresholds for OWAMP test results, no packet loss, some packet loss but less than 0.01%, and 0.01% or greater packet loss. A cell on the grid corresponding to an OWAMP test will be a different color based on the results of the test. In this way, we can easily see whether packet loss has occurred. And if some packets have been dropped, clicking on a cell on the grid brings up a graph of packet loss (and other statistics) over time, allowing one to investigate whether the packet loss was significant.

We have tested MaDDash by manually dropping packets. When reading measurements from 7 days of OWAMP testing, MaDDash was able to account for and remember a single manually dropped packet. Scripts have been written to initiate OWAMP testing, and to stop OWAMP testing and clear the tasks scheduled for all test nodes involved, so that throughput tests can be ran without interfering with the results of the OWAMP testing and vice versa.

We can change how far in the past to check the measurement archive for results, allowing us to populate MaDDash with results from a time period of our choosing. In this way we can get rid of undesirable measurements, such as ones that occurred when there is additional traffic from some other source, or if the Hub is powered off when a measurement is made.

Early Results:

So far we have ran the multisource OWAMP test for a period of five days and observed only three packets lost across the entire setup, showing the GbE switch to be capable of handling traffic between many sources at once.

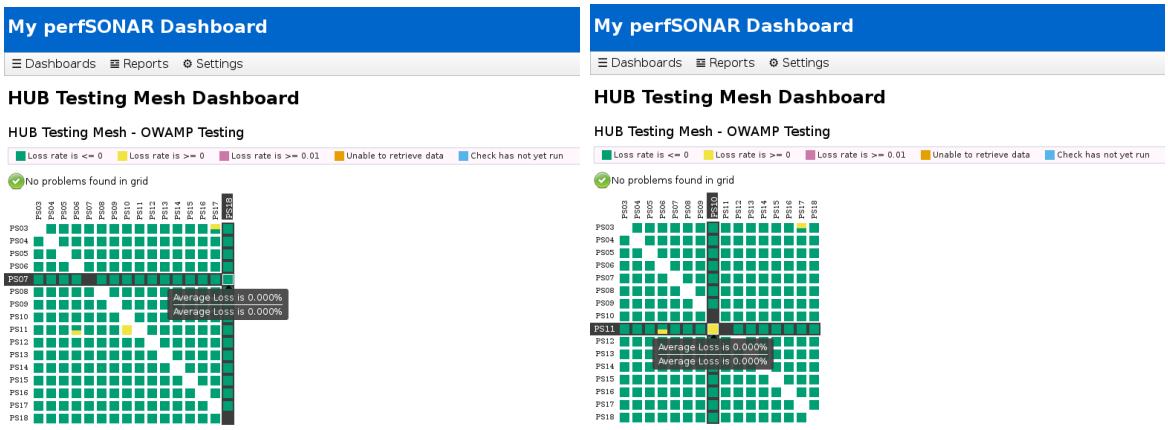


Illustration 3: MaDDash Dashboard Interface

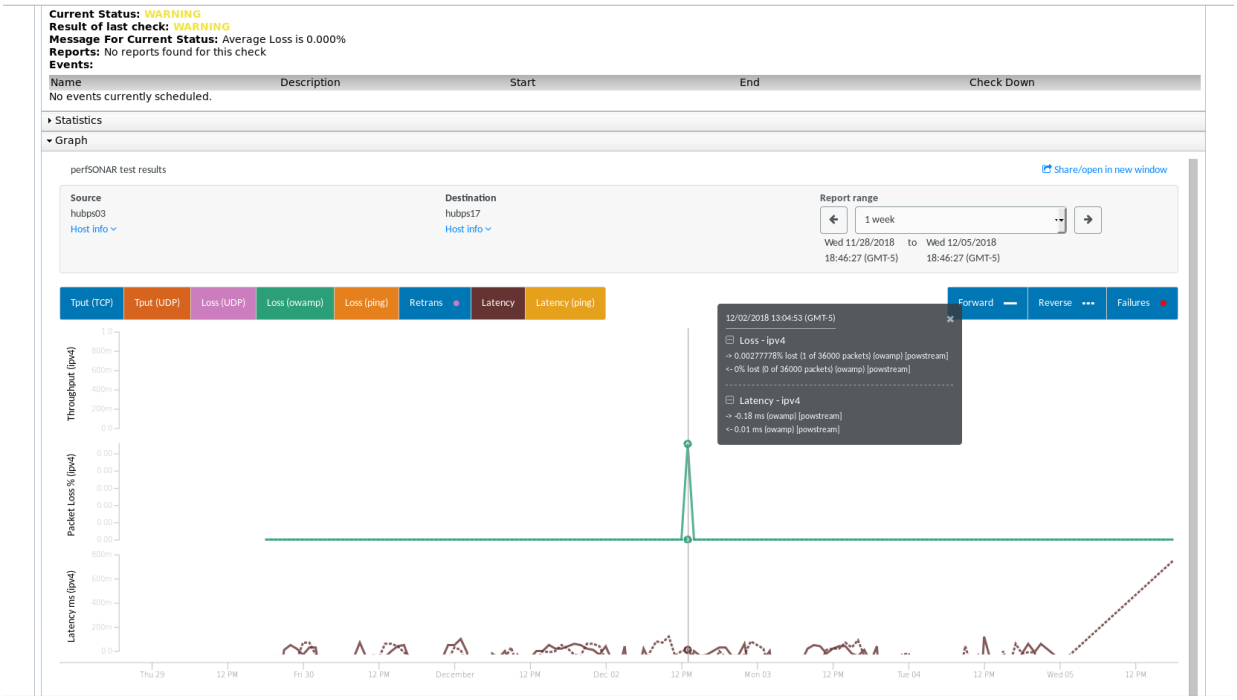


Illustration 4: Graph of packet loss over time. Can see that only one packet was dropped from hubps03 to hubps17 over a five day period.

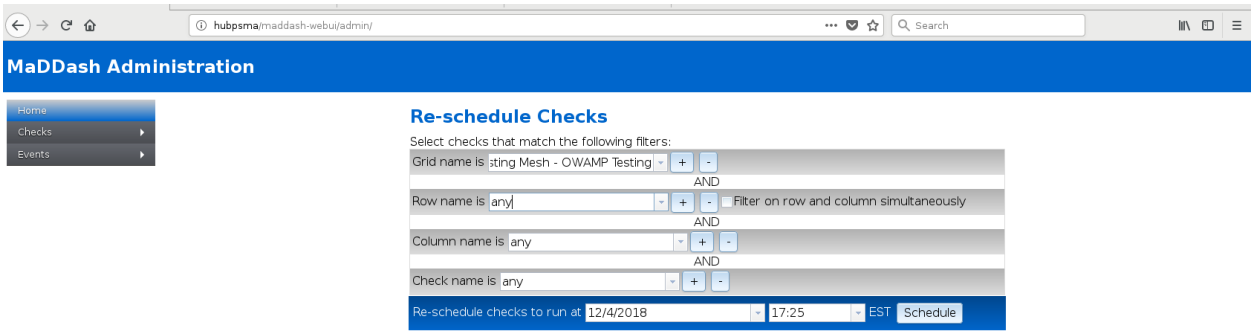


Illustration 5: The MaDDash Dashboard is automatically populated with measurement results at regular intervals over time, but these checks can be manually updated to occur immediately