Network Performance Tools

Jeff Boote
Internet2/R&D

June 1, 2008
NANOG 43/ Brooklyn, NY
Overview

• BWCTL
• OWAMP
• NDT/NPAD
BWCTL: What is it?

A resource allocation and scheduling daemon for arbitration of iperf tests
Problem Statement

• Users want to verify available bandwidth from their site to another.

Methodology

– Verify available bandwidth from each endpoint to points in the middle to determine problem area.
Typical Solution

• Run “iperf” or similar tool on two endpoints and hosts on intermediate paths
Typical road blocks

• Need software on all test systems
• Need permissions on all systems involved (usually full shell accounts*)
• Need to coordinate testing with others *
• Need to run software on both sides with specified test parameters *

(* BWCTL was designed to help with these)
Implementation

Applications
- bwctld daemon
- bwctl client

Open source license and development

Built upon protocol abstraction library
- Supports one-off applications
- Allows authentication/policy hooks to be incorporated
**Functionality (bwctl)**

bwctl client application makes requests to both endpoints of a test

- Communication can be “open”, “authenticated”, or “encrypted” (encrypted reserved for future use)
- Requests include a request for a time slot as well as a full parameterization of the test
- Third party requests
- If no server is available on the localhost, client handles test endpoint
- *Mostly* the same command line options as iperf (some options limited or not implemented.)
Functionality (bwctlld)

bwctlld on each test host
  – Accepts requests for “iperf” tests including time slot and parameters for test
  – Responds with a tentative reservation or a denied message
  – Reservations by a client must be confirmed with a “start session” message
  – Resource “Broker”
  – Runs tests
  – Both “sides” of test get results
**BWCTL Example**

```
[bgoode@nms-rthr2 ~]$ bwctl -x -s bwctl.kans.net.internet2.edu
bwctl: 19 seconds until test results available

RECEIVER START
--------------------------------------------------------------------------
Server listening on TCP port 5001
Binding to local address 2001:468:9:100::16:22
TCP window size: 87380 Byte (default)
--------------------------------------------------------------------------
[ 14] 0.0-10.2 sec 1193058304 Bytes 939913512 bits/sec
[ 14] MSS size 8928 bytes (MTU 8968 bytes, unknown interface)

RECEIVER END

SENDER START
--------------------------------------------------------------------------
Client connecting to 2001:468:9:100::16:22, TCP port 5001
Binding to local address 2001:468:4:100::16:214
TCP window size: 87380 Byte (default)
--------------------------------------------------------------------------
[ 7] 0.0-10.0 sec 1193058304 Bytes 951107779 bits/sec
[ 7] MSS size 8928 bytes (MTU 8968 bytes, unknown interface)

SENDER END
[bgoode@nms-rthr2 ~]$  
```
**BWCTL Data (Dash-Board)**

![Internet2 IPv4 TCP Throughput](image)

**Internet2 IPv4 TCP Throughput**

[Throughput(Mbps) / Date&Time]

<table>
<thead>
<tr>
<th>BWTCP4</th>
<th>Senders</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ann_arbor_mi</td>
<td>washington_dc</td>
<td>arlington_va</td>
</tr>
<tr>
<td>ann_arbor_mi</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>washington_dc</td>
<td>48.0 / Sat May 31 18:32:16 UTC 2008</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arlington_va</td>
<td>*/ *</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Receivers</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ann_arbor_mi</td>
<td>40.6 / Sat May 31 18:48:40 UTC 2008</td>
<td></td>
<td></td>
</tr>
<tr>
<td>washington_dc</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arlington_va</td>
<td>*/ *</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
BWCTL Data – Path History
Resource Allocation (bwctld)

• Each connection is “classified” (authentication)

• Each classification is hierarchical and has an associated set of hierarchical limits:
  – Connection policy (allow_open_mode)
  – Bandwidth (allow_tcp, allow_udp, bandwidth)
  – Scheduling (duration, event_horizon, pending)
    • A time slot is simply a time-dependent resource that needs to be allocated just like any other resource. It therefore follows the resource allocation model.
BWCTL: 3-party Interaction

bwctl client

bwctl resource broker (master daemon)

bwctl (request broker)

bwctl (peer agent)

iperf (test process)

bwctl (request broker)

bwctl (peer agent)

iperf (test process)
BWCTL: No Local Server

bwctl client

bwctl resource broker (master daemon)

(request broker)

(request broker)

(peer agent)

(peer agent)

iperf (test process)

iperf (test process)

initial connection

requests/results

requests/results

verify time/results

resource requests

test stream
Tester Applications

• Iperf is primary “tester”
  – Well known – widely used

• Problems integrating exec’d tool
  – Server initialization (port number allocation)
  – error conditions
  – No indication of partial progress (How full was the send buffer when the session was killed?)

• thrulay/nuttcp are available in latest ‘RC’ version of bwctl
General Requirements

• Iperf version 2.0 and 2.0.2
• NTP (ntpd) synchronized clock on the local system
  – Used for scheduling
  – More important that errors are accurate than the clock itself
• Firewalls:
  – Lots of ports for communication and testing
• End hosts must be tuned!

http://www.psc.edu/networking/perf_tune.html
Supported Systems

• FreeBSD 4.x, 5.x
• Linux 2.4, 2.6
• (Most recent versions of UNIX should work)
Policy/Security Considerations

• DoS source
  – Imagine a large number of compromised BWCTLD servers being used to direct traffic

• DoS target
  – Someone might attempt to affect statistics web pages to see how much impact they can have

• Resource consumption
  – Time slots
  – Network bandwidth
Policy Recommendations

- Restrictive for UDP
- More liberal for TCP tests
- More liberal still for “peers”
- Protect AES keys!
Availability

• Currently available
  
  http://e2epi.internet2.edu/bwctl/

Mail lists:
• bwctl-users@internet2.edu
• bwctl-announce@internet2.edu

https://mail.internet2.edu/wws/lists/engineering
OWAMP: What is it?

OWD or One-Way PING

- A control protocol
- A test protocol
- A sample implementation of both
Why the OWAMP protocol?

• Find problems in the network
  – Congestion usually happens in one direction first...
  – Routing (asymmetric, or just changes)
  – SNMP polling intervals mask high queue levels that active probes can show

• There have been many implementations to do One-Way delay over the years (Surveyor, Ripe...)
  – The problem has been interoperability.
  – http://www.ietf.org/rfc/rfc4656.txt
OWAMP Control protocol

- Supports authentication and authorization
- Used to configure tests
  - Endpoint controlled port numbers
  - Extremely configurable send schedule
  - Configurable packet sizes
- Used to start/stop tests
- Used to retrieve results
  - Provisions for dealing with partial session results
OWAMP Test protocol

• Packets can be “open”, “authenticated”, or “encrypted”
Sample Implementation

Applications
- owampd daemon
- owping client

Open source license and development

Built upon protocol abstraction library
- Supports one-off applications
- Allows authentication/policy hooks to be incorporated
Functionality (owping client)

– owping client requests OWD tests from an OWAMP server
– Client can be sender or receiver
– Communication can be “open”, “authenticated”, or “encrypted”
– Supports the setup of many tests concurrently
– Supports the buffering of results on the server for later retrieval
Functionality (owampd)

owampd

- Accepts requests for OWD tests
- Responds with accepted/denied
- Tests are formally started with a StartSessions message from the client.
- Runs tests
- Sessions with packets received at the server are buffered for later retrieval
OWPING Example

boote@nms-rlat:~[360]$ owping nms-rlat.newy.net.internet2.edu
Approximately 13.0 seconds until results available

--- owping statistics from [64.57.17.34]:45355 to [nms-rlat.newy.net.internet2.edu]:44244 ---
SID:   40391162cbec228e81118c1953a5eef9
100 sent, 0 lost (0.000%), 0 duplicates
one-way delay min/median/max = 11/11/11 ms, (err=0.0442 ms)
one-way jitter = 0 ms (P95-P50)
Hops = 3 (consistently)
no reordering

--- owping statistics from [nms-rlat.newy.net.internet2.edu]:44247 to [64.57.17.34]:45356 ---
SID:   40391122cbec228e6bb1bde827906fe35
last:  2008-05-31T19:16:41.979
100 sent, 0 lost (0.000%), 0 duplicates
one-way delay min/median/max = 10.9/11/11 ms, (err=0.0442 ms)
one-way jitter = 0 ms (P95-P50)
Hops = 3 (consistently)
no reordering

boote@nms-rlat:~[361]$
## OWAMP Data (Dash-Board)

### OWAMP Data

**[Latency (ms) / Packet Loss (%)]**

<table>
<thead>
<tr>
<th>Location</th>
<th>ISP</th>
<th>Atlanta</th>
<th>Chicago</th>
<th>Houston</th>
<th>KansasCity</th>
<th>LosAngeles</th>
<th>NewYorkCity</th>
<th>SaltLakeCity</th>
<th>Seattle</th>
<th>Washington</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Senders</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Annapolis</td>
<td>IPv4</td>
<td>0.010 /</td>
<td>0.010 /</td>
<td>0.010 /</td>
<td>0.010 /</td>
<td>0.010 /</td>
<td>0.010 /</td>
<td>0.010 /</td>
<td>0.010 /</td>
<td>0.010 /</td>
</tr>
<tr>
<td></td>
<td>IPv6</td>
<td>0.010 /</td>
<td>0.010 /</td>
<td>0.010 /</td>
<td>0.010 /</td>
<td>0.010 /</td>
<td>0.010 /</td>
<td>0.010 /</td>
<td>0.010 /</td>
<td>0.010 /</td>
</tr>
<tr>
<td>Chicago</td>
<td>IPv4</td>
<td>12.218 /</td>
<td>12.218 /</td>
<td>12.218 /</td>
<td>12.218 /</td>
<td>12.218 /</td>
<td>12.218 /</td>
<td>12.218 /</td>
<td>12.218 /</td>
<td>12.218 /</td>
</tr>
<tr>
<td></td>
<td>IPv6</td>
<td>12.218 /</td>
<td>12.218 /</td>
<td>12.218 /</td>
<td>12.218 /</td>
<td>12.218 /</td>
<td>12.218 /</td>
<td>12.218 /</td>
<td>12.218 /</td>
<td>12.218 /</td>
</tr>
<tr>
<td>Houston</td>
<td>IPv4</td>
<td>5.507 /</td>
<td>5.507 /</td>
<td>5.507 /</td>
<td>5.507 /</td>
<td>5.507 /</td>
<td>5.507 /</td>
<td>5.507 /</td>
<td>5.507 /</td>
<td>5.507 /</td>
</tr>
<tr>
<td></td>
<td>IPv6</td>
<td>5.507 /</td>
<td>5.507 /</td>
<td>5.507 /</td>
<td>5.507 /</td>
<td>5.507 /</td>
<td>5.507 /</td>
<td>5.507 /</td>
<td>5.507 /</td>
<td>5.507 /</td>
</tr>
<tr>
<td>KansasCity</td>
<td>IPv4</td>
<td>15.294 /</td>
<td>15.294 /</td>
<td>15.294 /</td>
<td>15.294 /</td>
<td>15.294 /</td>
<td>15.294 /</td>
<td>15.294 /</td>
<td>15.294 /</td>
<td>15.294 /</td>
</tr>
<tr>
<td></td>
<td>IPv6</td>
<td>15.294 /</td>
<td>15.294 /</td>
<td>15.294 /</td>
<td>15.294 /</td>
<td>15.294 /</td>
<td>15.294 /</td>
<td>15.294 /</td>
<td>15.294 /</td>
<td>15.294 /</td>
</tr>
</tbody>
</table>

[http://owamp.net.internet2.edu/owamp_grid.cgi][1]

[1]: http://owamp.net.internet2.edu/owamp_grid.cgi
OWAMP Data (Path)

OWAMP (network latency)
Paths to and from Washington_DC: Sat May 31 14:50:48 EDT --- Sat May 31 14:50:48 EDT

IPv4 Latency

Sender: Washington_DC

Receiver: Washington_DC

Ann Arbor MI

Graphs showing latency data with different percentile lines.
Resource Allocation

• Each connection is “classified” (authentication)
• Each classification is associated with a set of hierarchical limits
  – Bandwidth (bandwidth)
  – Session buffer (disk)
  – Data retention (delete_on_fetch)
  – Connection policy (allow_open_mode)

(no time dependent dimension to resource allocation in owampd)
Architecture
General Requirements

• NTP (ntpd) synchronized clock on the local system
  – Specific configuration requirements as specified in NTP talk...
  – Strictly speaking, owamp will work without ntp. However, your results will be meaningless in many cases
• gnumake for build process
Supported Systems

• FreeBSD 4.7+, 5.x, 6.0 (64-bit)
• Linux 2.4, 2.6 (64-bit)
• MacOS X 10.4.X
• Solaris 10.4.5
• (Most recent versions of UNIX should work)
Recommended Hardware

• Stable System Clock
  – Temperature controlled environment
  – No power management of CPU

• No strict requirements for CPU, Memory, Bus speed
  – More tasking schedules will require more capable hardware
Operational concerns

Time:
- NTP issues predominate the problems
- Determining an accurate timestamp “error” is in many ways more difficult than getting a “very good” timestamp
- Working as an “open” server requires UTC time source (For predefined test peers, other options available)

Firewalls:
- Port filter trade-off
  - Administrators like pre-defined port numbers
  - Vendor manufactures would probably like to “prioritize” test traffic
  - Owampd allows a range of ports to be specified for the receiver
Policy/Security Considerations

• Third-Party DoS source
• DoS target
• Resource consumption
  – Memory (primary and secondary)
  – Network bandwidth
Policy Recommendations

• Restrict overall bandwidth to something relatively small
  – Most OWAMP sessions do not require much

• Limit “open” tests to ensure they do not interfere with precision of other tests
Availability

• Currently available

http://e2epi.internet2.edu/owamp/

Mail lists:

• owamp-users@internet2.edu
• owamp-announce@internet2.edu

https://mail.internet2.edu/wws/lists/engineering
Advanced user tools

• NDT
  – Allows users to test network path for a limited number of common problems

• NPAD
  – Allows users to test local network infrastructure while simulating a long path
Underlying User Assumption

• When problems exist, it’s the network’s fault!
Simple Network Picture

Bob’s Host

Network Infrastructure

Carol’s Host
Network Infrastructure
NDT: What is it?

Web browser invoked advanced user based diagnostics

- Allows users to test a network path for a limited number of common problems – from their desktop
- NDT allows user to give the network administrator a detailed view of exactly what the users host is doing
- Allows the user to be an active participant in the debugging process – allows them to more directly see how host configuration effects performance

Attempts to answer the questions:
- What performance should a user expect?
- What is the limiting factor?
NDT Goals

• Identify real problems for real users
  – Network infrastructure is the problem
  – Host tuning issues are the problem
• Make tool simple to use and understand
• Make tool useful for users and network administrators
NDT user interface

- Web-based JAVA applet allows testing from any browser
- Command-line client allows testing from remote login shell
NDT sample Results

This Java applet was developed to test the reliability 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/WB100 Network Diagnostic Tool v5.3.4e
Click START to begin.
Checking for Mobileboxes.................. Done

Running 100 outbound test (client to server)......... 360.76Kbps
Running 100 inbound test (server to client)......... 20.53Mbps

Warning: Client time-out while receiving data, possible duplex mismatch exists.
The slowest link in the end-to-end path is a 100 Mbps Fast Ethernet subnet.
Alarm: Duplex Mismatch condition detected Switch = full and Host = half

Click START to re-test.
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
Finding Results of Interest

• 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
## Finding NDT Servers

### NDT Servers

<table>
<thead>
<tr>
<th>Location</th>
<th>Host</th>
<th>Interface</th>
<th>Online Stats</th>
</tr>
</thead>
</table>

### PNP Equipment Used for Abilene:
- Intel SC80 motherboard,
- 2 x 533 MHz 383, 512 KB L2 cache, 333MHz FSB,
- 2 x 1GB DDR ECC registered RAM (one slot to enable interleaving),
- 2 x 3GB 160GB SSD10 (721640-0000)
- Primary NIC: Systrom Gigabit Ethernet SK-1863 SX
- Secondary NIC: Intel Ethernet Pro 1000 S5 (845885)

### Other Available Servers:

<table>
<thead>
<tr>
<th>Location</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANL</td>
<td><a href="http://miranda.anl.gov:7113">http://miranda.anl.gov:7113</a></td>
</tr>
<tr>
<td>Swiss Education and Research Network</td>
<td>(Switzerland) <a href="http://www.mlcc.web.psi.ch/">http://www.mlcc.web.psi.ch/</a></td>
</tr>
<tr>
<td>University of Michigan (Ann Arbor)</td>
<td><a href="http://www.eecs.umich.edu/">http://www.eecs.umich.edu/</a></td>
</tr>
<tr>
<td>UC at Santa Cruz</td>
<td><a href="http://ntrc.ucsc.edu/">http://ntrc.ucsc.edu/</a></td>
</tr>
<tr>
<td>Thomas Jefferson National Accelerator</td>
<td><a href="http://vjm">http://vjm</a> fron.ucr.edu:7123</td>
</tr>
<tr>
<td>Stanford University</td>
<td><a href="http://retro.scc.stanford.edu/">http://retro.scc.stanford.edu/</a></td>
</tr>
<tr>
<td>University of Hawaii (Honolulu)</td>
<td><a href="http://fermat.ucr.edu:7123">http://fermat.ucr.edu:7123</a></td>
</tr>
<tr>
<td>St. Mary’s College of Maryland (Maryland)</td>
<td><a href="http://ndt.stmarys.edu:7123">http://ndt.stmarys.edu:7123</a></td>
</tr>
</tbody>
</table>

This original idea and implementation of the web-based testing server was designed and implemented by Tom Cunningham from Oak Ridge National Laboratory. It has been extensively modified by Rob Carlson and changed to perform the current functions. This material is based on work supported (in part) by the Office of Science, U.S. Department of Energy under Contract W-7405-ENG-36 and Argonne National Laboratory.
NPAD/pathdiag: What is it?

• Web browser invoked advanced user based diagnostics

  – Allows users to test a limited portion of the network path looking for problems that would adversely effect longer paths

  – Attempts to answer the questions:

    • What performance should a user expect?
    • What is the limiting factor?
NPAD/pathdiag

- A new tool from researchers at Pittsburgh Supercomputer Center
- Finds problems that affect long network paths
- Uses Web100-enhanced Linux based server
- Web based Java client
Long Path Problem

1 msec H1 – H2

70 msec H1 – H3

Switch 1

Switch 2

Switch 3

Switch 4

H1

H2

H3

X

1 msec H1 – H2

70 msec H1 – H3
Long Path Problem

• E2E application performance is dependent on distance between hosts
• Full size frame time at 100 Mbps
  – Frame = 1500 Bytes
  – Time = 0.12 msec
  – In flight for 1 msec RTT = 8 packets
  – In flight for 70 msec RTT = 583 packets
TCP Congestion Avoidance

- Cut number of packets by $\frac{1}{2}$
- Increase by 1 per RTT
  - LAN (RTT=1msec)
    - In flight changes to 4 packets
    - Time to increase back to 8 is 4msec
  - WAN (RTT = 70 msec)
    - In flight changes to 292 packets
    - Time to increase back to 583 is 20.4 seconds
NPAD Server main page

NPAD (Network Path and Application Diagnostics) is designed to diagnose network performance problems in your end-system (the machine you have in your home) or in the network between and your remote NPAD server. For each diagnosed problem, the server provides corrective actions with authorizations suitable for non-experts.

Brief instructions:
- The test results are most accurate over a short network path. If the NPAD server is located at Internet2 in Ann Arbor, MI is not near you, look for a closer server from the list of Current NPAD Diagnostic Servers.
- There are two test types: application-performance, goal 1 (high round-trip time and latency) and goal 2 (low round-trip time and latency). Enter the parameters on the form below and click Start Test. Messages will appear in the log window as the test runs. Followed by a diagnostic report.
- Failure to meet the goals will indicate problems that will prevent the application from meeting the end-to-end performance goal. For each message, a question-mark link (?) or red text leads to additional detailed information about the results.
- Every test in NPAD logged and test results are public. We use the logs and results to further refine the software.

For more information, see the NPAD Documentation, especially the sections:
- NPAD Diagnostic Procedures - for test instructions.
- Testing Methodology - why the tests work.
- Outcomes - what to do next in the broader debugging context.

Test from server web100.internet2.edu to this machine

Round Trip Time (ms): 22
Target Rate (Mbps): 7

Please send comments and suggestions about the server to webmaster@internet2.edu
NPAD Sample results
Finding NPAD servers

Current NPAD Diagnostic Servers

Select the NPAD diagnostic server that is the closest to you in terms of network round trip time. This will generally be the geographically closest server connected to the same national backbone as you are. The servers below are organized by connected backbone and sorted east to west.

- Sites connected to Abilene/Internet 2
  - Durham NC via Duke/NCREN/12
  - Pittsburgh PA via PCC/3ROX/12
  - Ann Arbor MI via 12/MERIT/12
  - Oklahoma City OK via ONEnet/12
  - Denver CO via FCRP/12
  - Boulder CO via NCAR/UCAR/FCRP/12

- Sites connected via EUnet
  - Berkeley CA via LBNL/EUnet

- If need a closer server, consider asking your ISP, Ogg/Pop or campus network administrators to install an NPAD diagnostic server.

Interpreting the Results

When you go the nearest NPAD server and run a diagnostic test as suggested above, paching returns a web page which reports all of the test results. The messages indicate...
Try these tools

Network Performance Toolkit


Knoppix disk (OS on a CD) that has:
  Iperf, thrulay, bwctl, owamp, NDT, NPAD, reverse-traceroute/ping...
Questions?