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

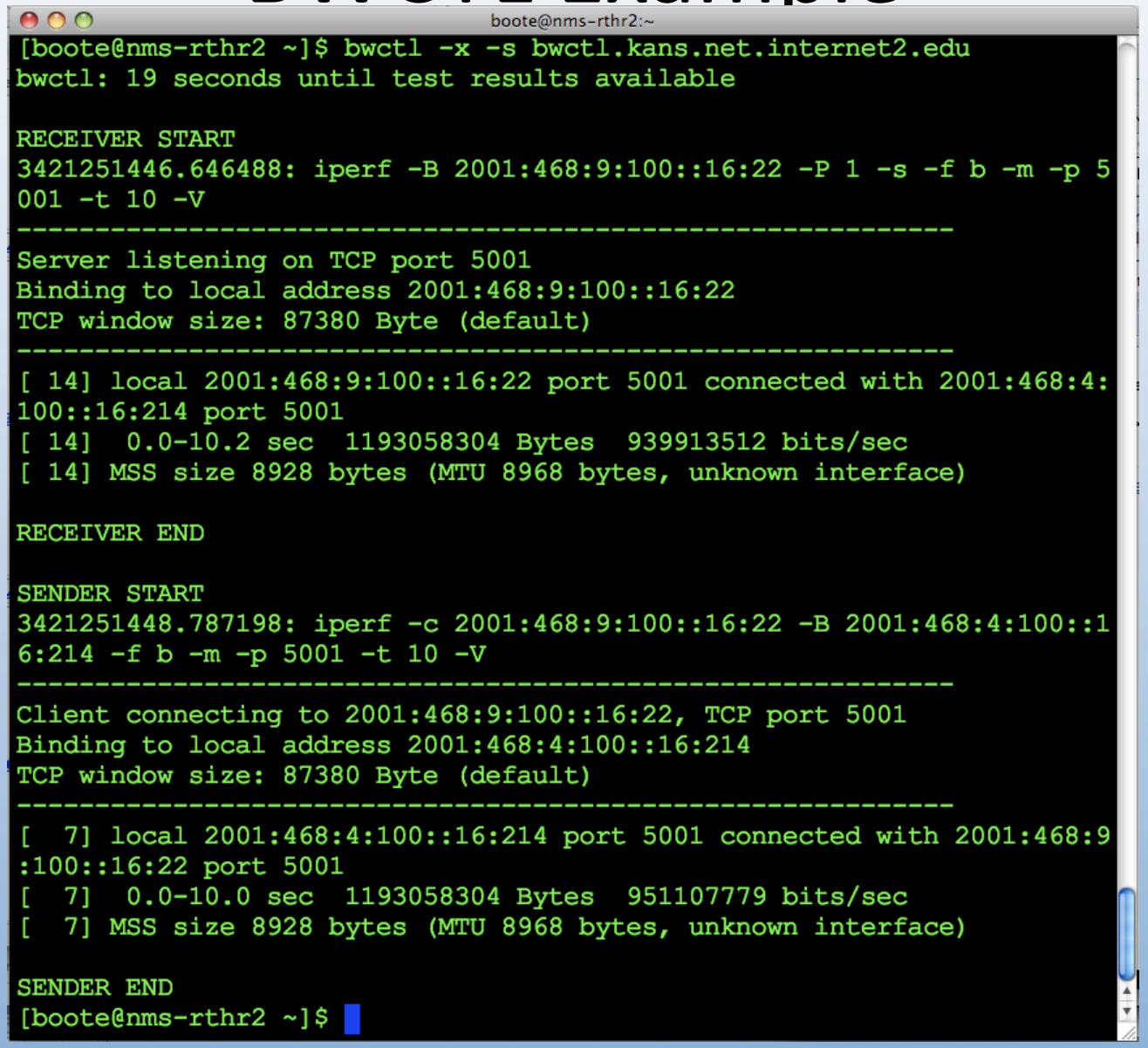
- Communication can be “open”, “authenticated”, or “encrypted” (encrypted reserved for future use)
- Requests include a request for a time slot as well as a full parameterization of the test
- Third party requests
- If no server is available on the localhost, client handles test endpoint
- *Mostly* the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

Functionality (bwctld)

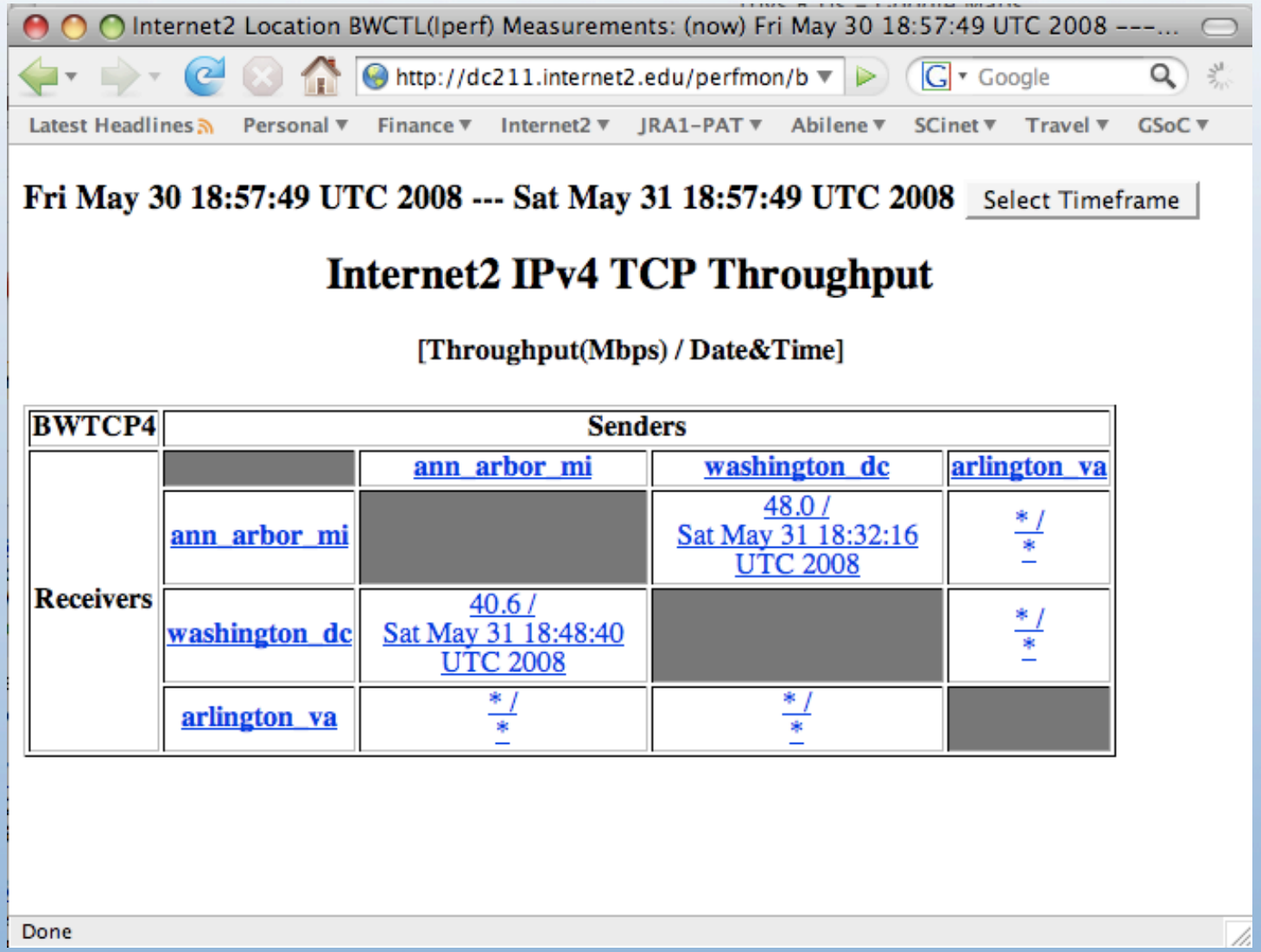
bwctld on each test host

- 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

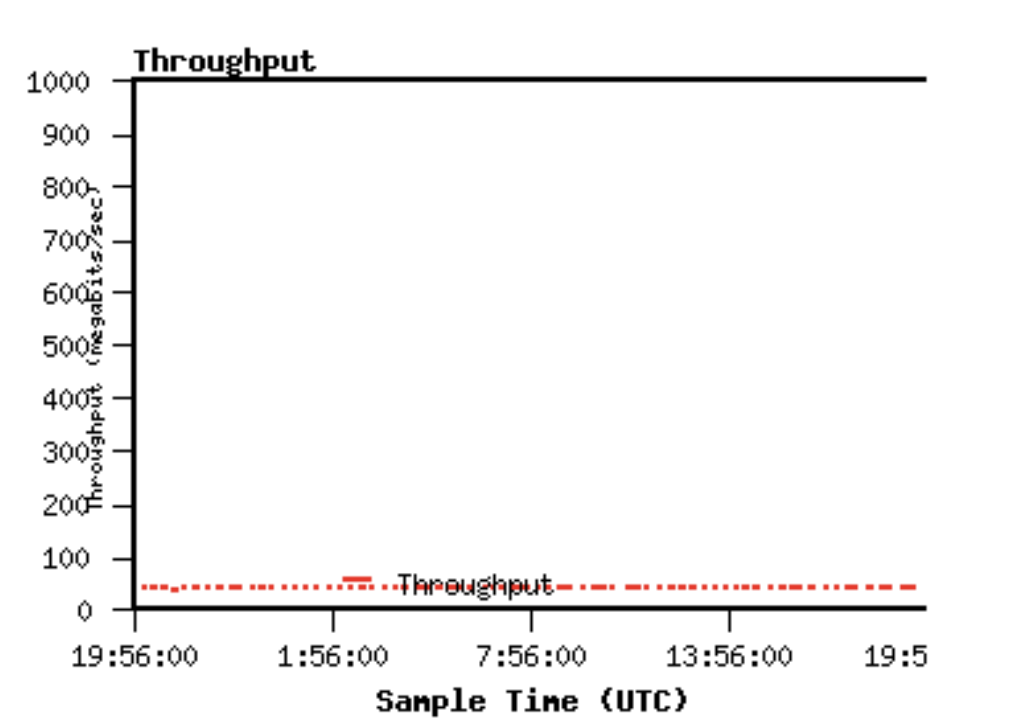
BWCTL Example



BWCTL Data (Dash-Board)



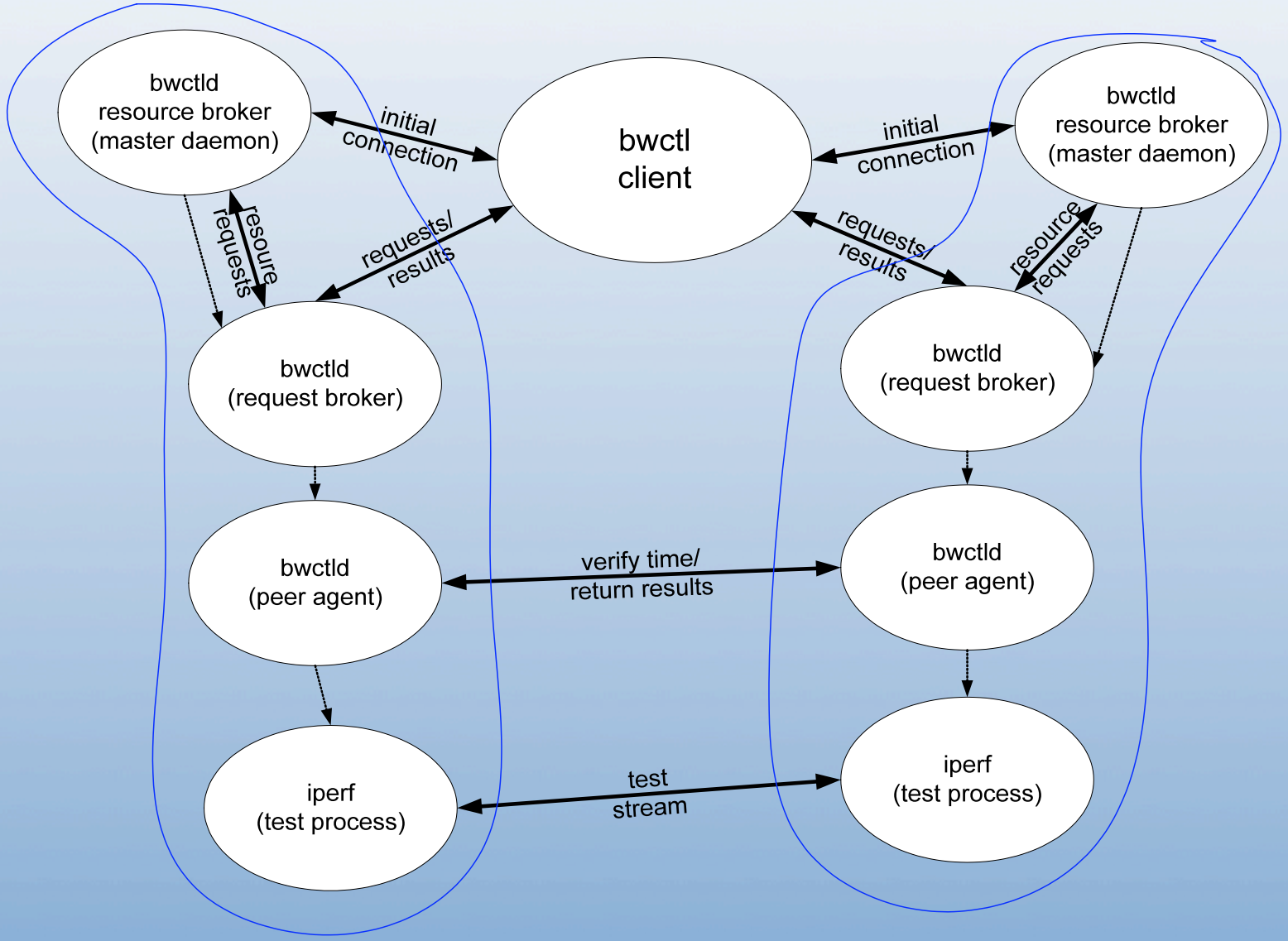
BWCTL Data – Path History



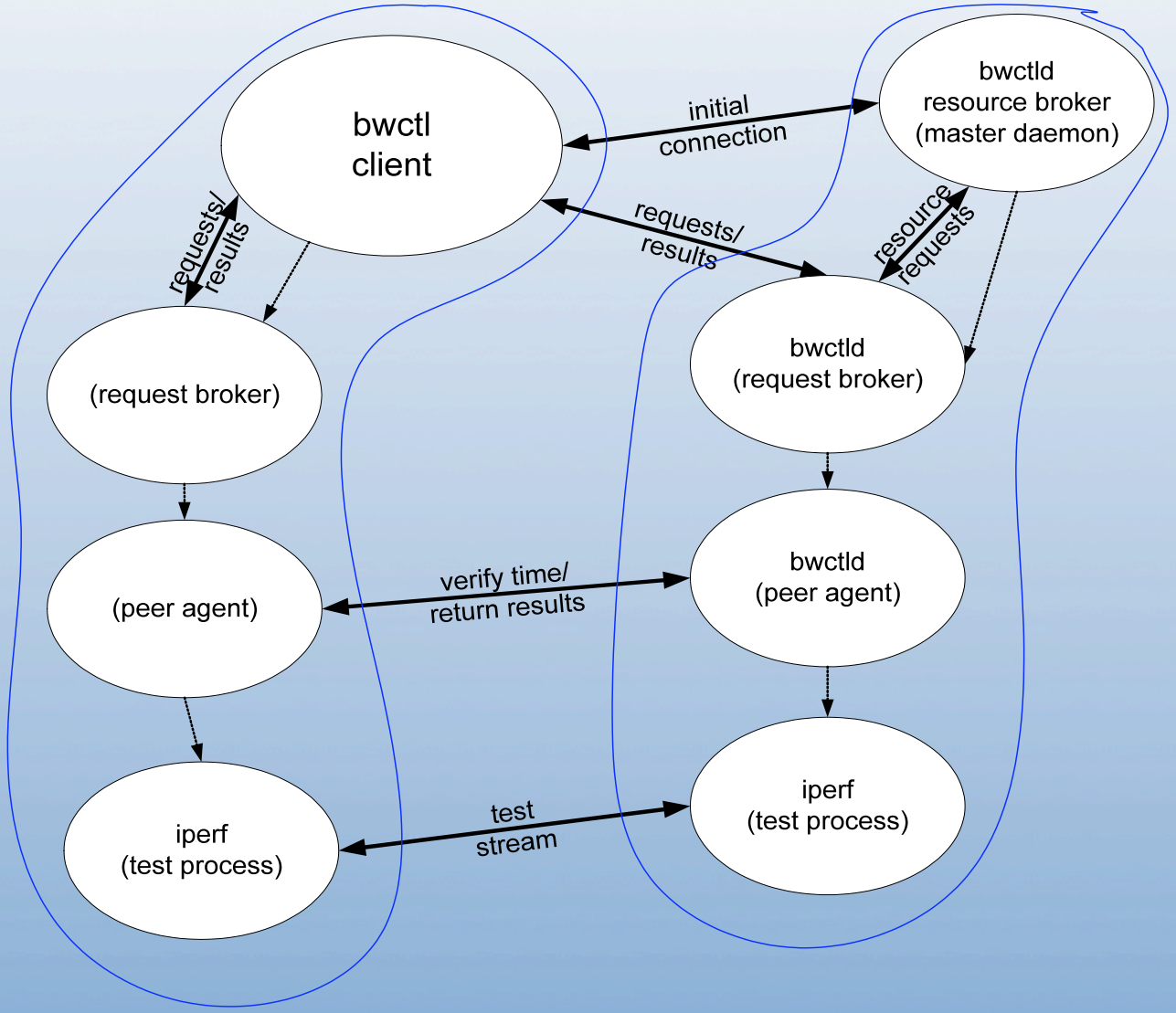
Resource Allocation (bwctld)

- 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: No Local Server



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

  http://www-didc.lbl.gov/TCP-tuning/buffers.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)

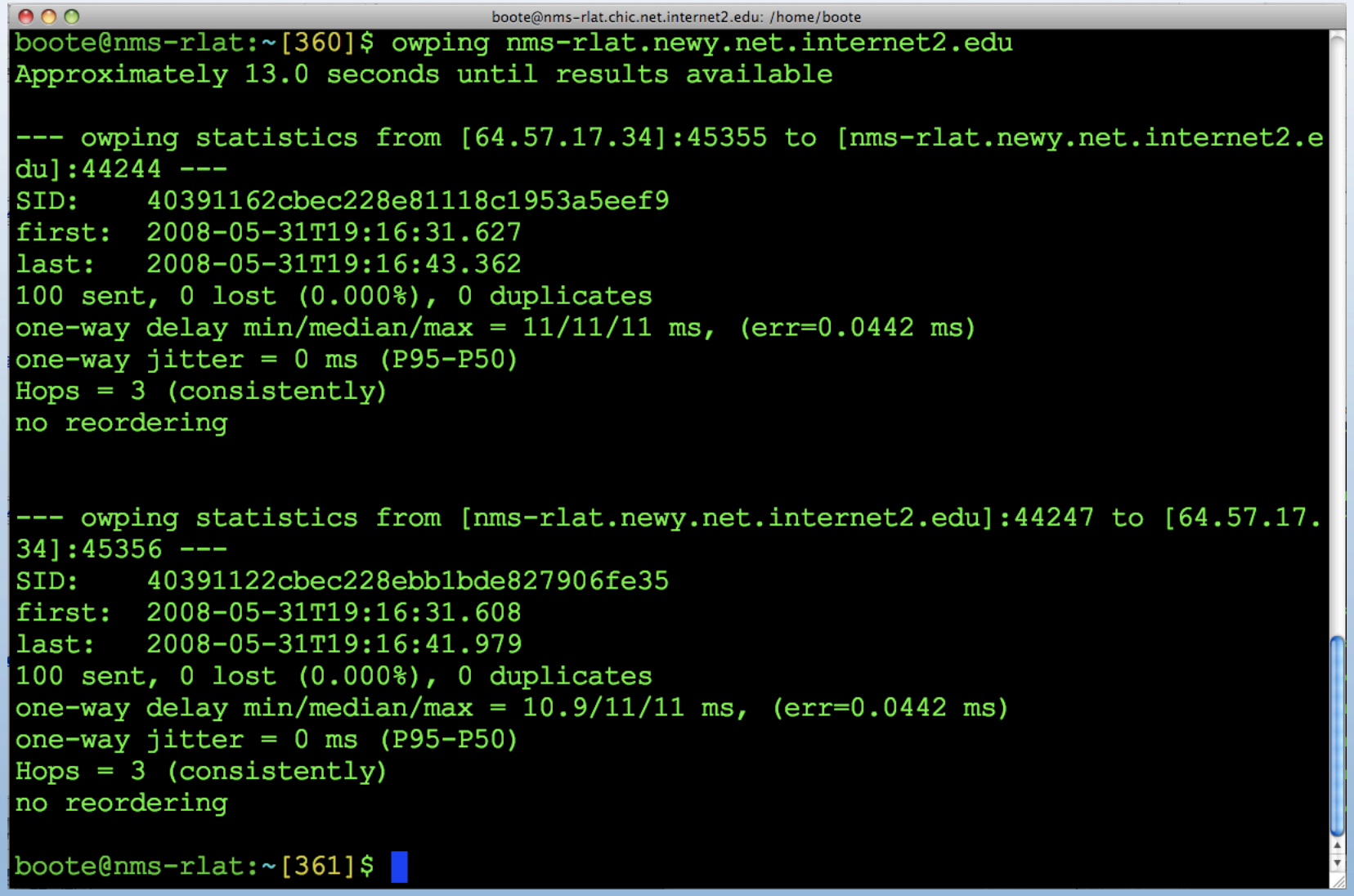
- owping client requests OWD tests from an OWAMP server
- Client can be sender or receiver
- Communication can be “open”, “authenticated”, or “encrypted”
- Supports the setup of many tests concurrently
- Supports the buffering of results on the server for later retrieval

Functionality (owampd)

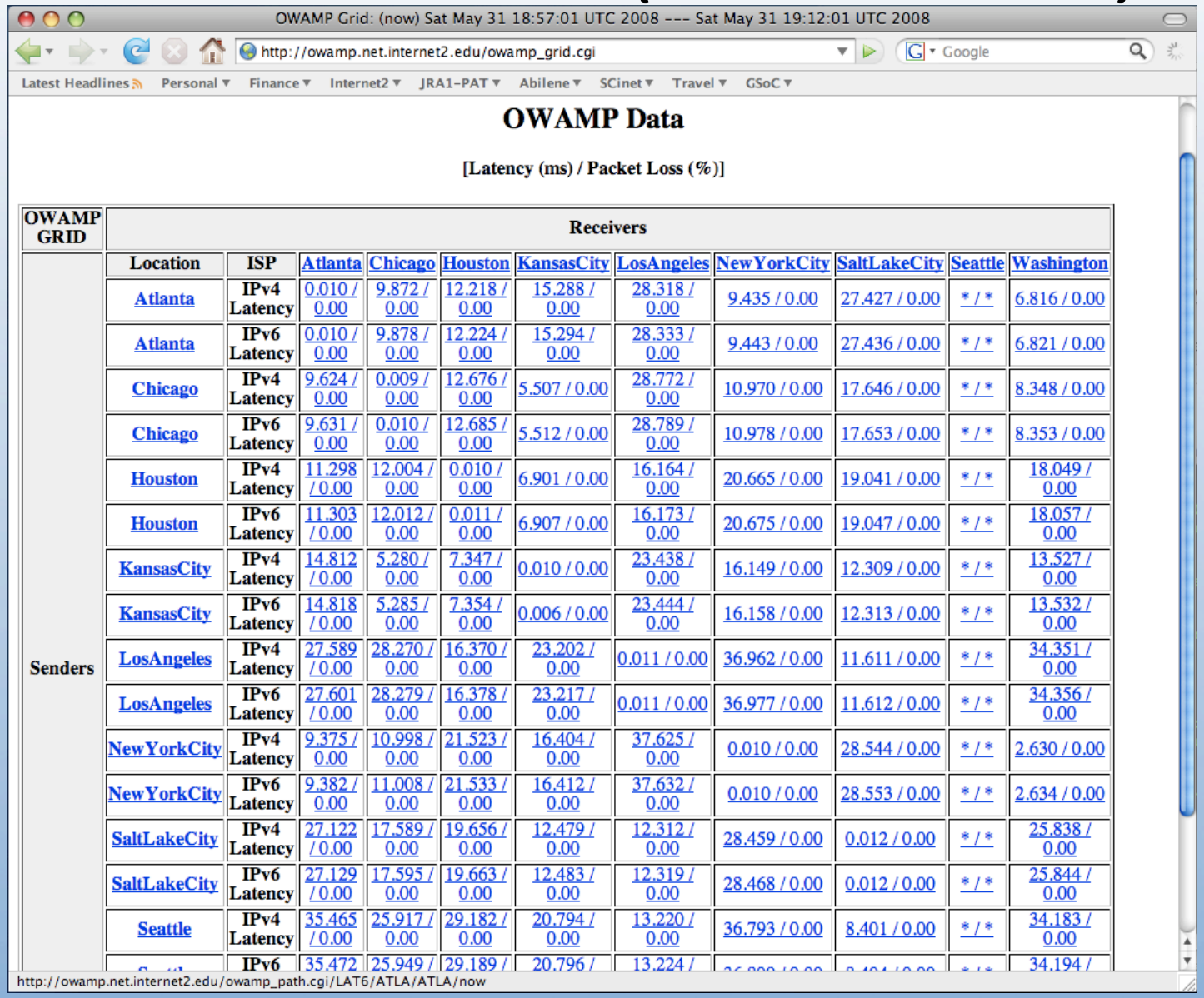
owampd

- 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



OWAMP Data (Dash-Board)



OWAMP Data

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

OWAMP GRID			Receivers								
	Location	ISP	Atlanta	Chicago	Houston	KansasCity	LosAngeles	NewYorkCity	SaltLakeCity	Seattle	Washington
Senders	Atlanta	IPv4 Latency	0.010 / 0.00	9.872 / 0.00	12.218 / 0.00	15.288 / 0.00	28.318 / 0.00	9.435 / 0.00	27.427 / 0.00	* / *	6.816 / 0.00
	Atlanta	IPv6 Latency	0.010 / 0.00	9.878 / 0.00	12.224 / 0.00	15.294 / 0.00	28.333 / 0.00	9.443 / 0.00	27.436 / 0.00	* / *	6.821 / 0.00
	Chicago	IPv4 Latency	9.624 / 0.00	0.009 / 0.00	12.676 / 0.00	5.507 / 0.00	28.772 / 0.00	10.970 / 0.00	17.646 / 0.00	* / *	8.348 / 0.00
	Chicago	IPv6 Latency	9.631 / 0.00	0.010 / 0.00	12.685 / 0.00	5.512 / 0.00	28.789 / 0.00	10.978 / 0.00	17.653 / 0.00	* / *	8.353 / 0.00
	Houston	IPv4 Latency	11.298 / 0.00	12.004 / 0.00	0.010 / 0.00	6.901 / 0.00	16.164 / 0.00	20.665 / 0.00	19.041 / 0.00	* / *	18.049 / 0.00
	Houston	IPv6 Latency	11.303 / 0.00	12.012 / 0.00	0.011 / 0.00	6.907 / 0.00	16.173 / 0.00	20.675 / 0.00	19.047 / 0.00	* / *	18.057 / 0.00
	KansasCity	IPv4 Latency	14.812 / 0.00	5.280 / 0.00	7.347 / 0.00	0.010 / 0.00	23.438 / 0.00	16.149 / 0.00	12.309 / 0.00	* / *	13.527 / 0.00
	KansasCity	IPv6 Latency	14.818 / 0.00	5.285 / 0.00	7.354 / 0.00	0.006 / 0.00	23.444 / 0.00	16.158 / 0.00	12.313 / 0.00	* / *	13.532 / 0.00
	LosAngeles	IPv4 Latency	27.589 / 0.00	28.270 / 0.00	16.370 / 0.00	23.202 / 0.00	0.011 / 0.00	36.962 / 0.00	11.611 / 0.00	* / *	34.351 / 0.00
	LosAngeles	IPv6 Latency	27.601 / 0.00	28.279 / 0.00	16.378 / 0.00	23.217 / 0.00	0.011 / 0.00	36.977 / 0.00	11.612 / 0.00	* / *	34.356 / 0.00
	NewYorkCity	IPv4 Latency	9.375 / 0.00	10.998 / 0.00	21.523 / 0.00	16.404 / 0.00	37.625 / 0.00	0.010 / 0.00	28.544 / 0.00	* / *	2.630 / 0.00
	NewYorkCity	IPv6 Latency	9.382 / 0.00	11.008 / 0.00	21.533 / 0.00	16.412 / 0.00	37.632 / 0.00	0.010 / 0.00	28.553 / 0.00	* / *	2.634 / 0.00
	SaltLakeCity	IPv4 Latency	27.122 / 0.00	17.589 / 0.00	19.656 / 0.00	12.479 / 0.00	12.312 / 0.00	28.459 / 0.00	0.012 / 0.00	* / *	25.838 / 0.00
	SaltLakeCity	IPv6 Latency	27.129 / 0.00	17.595 / 0.00	19.663 / 0.00	12.483 / 0.00	12.319 / 0.00	28.468 / 0.00	0.012 / 0.00	* / *	25.844 / 0.00
	Seattle	IPv4 Latency	35.465 / 0.00	25.917 / 0.00	29.182 / 0.00	20.794 / 0.00	13.220 / 0.00	36.793 / 0.00	8.401 / 0.00	* / *	34.183 / 0.00
		IPv6	35.472	25.949 /	29.189 /	20.796 /	13.224 /				34.194 /

OWAMP Data (Path)

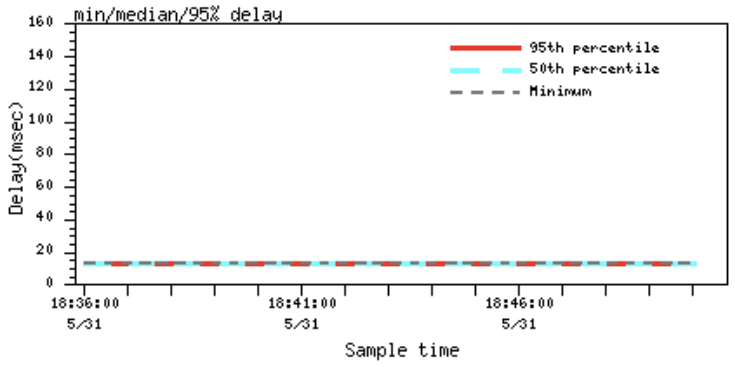
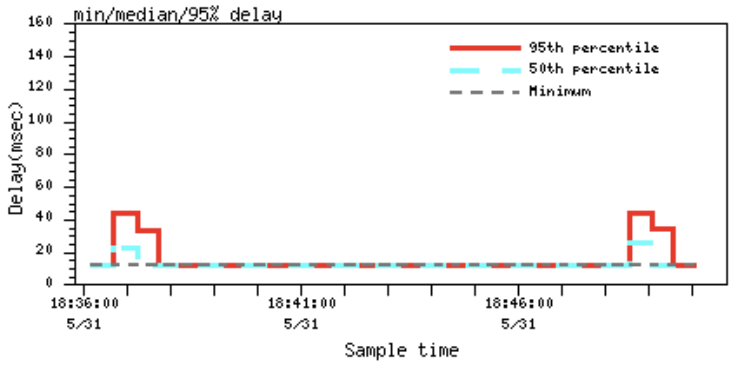
Paths to and from DC: Sat May 31 14:50:48 EDT --- Sat May 31 14:50:48 EDT

http://dc211.internet2.edu/perfmon/owamp_path.cgi/LAT4/DC/AA/now

Latest Headlines | Personal | Finance | Internet2 | JRA1-PAT | Abilene | SCinet | Travel | GSoC

OWAMP (network latency)

Paths to and from Washington_DC: Sat May 31 14:50:48 EDT --- Sat May 31 14:50:48 EDT

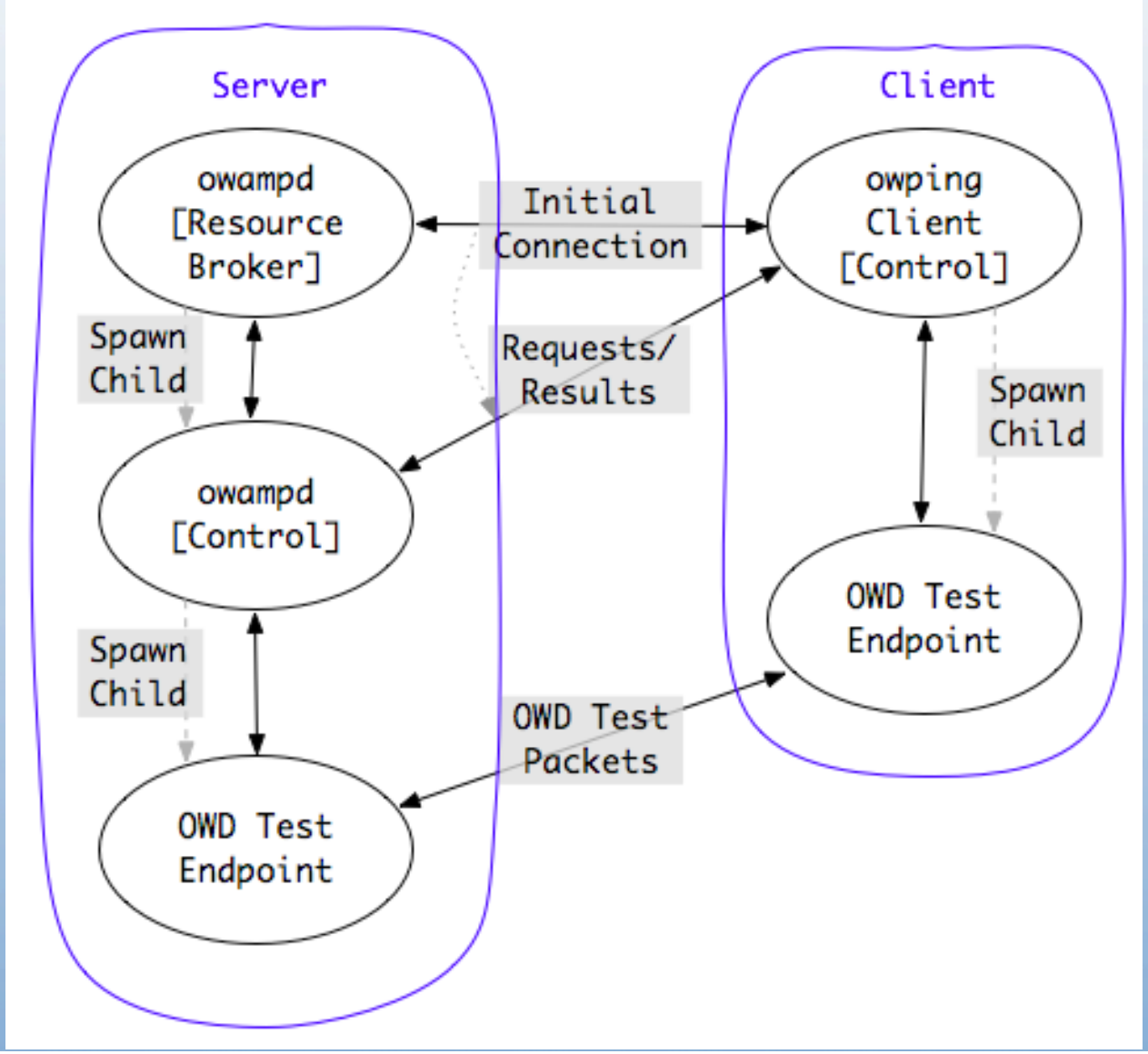
IPv4 Latency	Sender: Washington_DC	Receiver: Washington_DC
Ann_Arbor_MI		

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

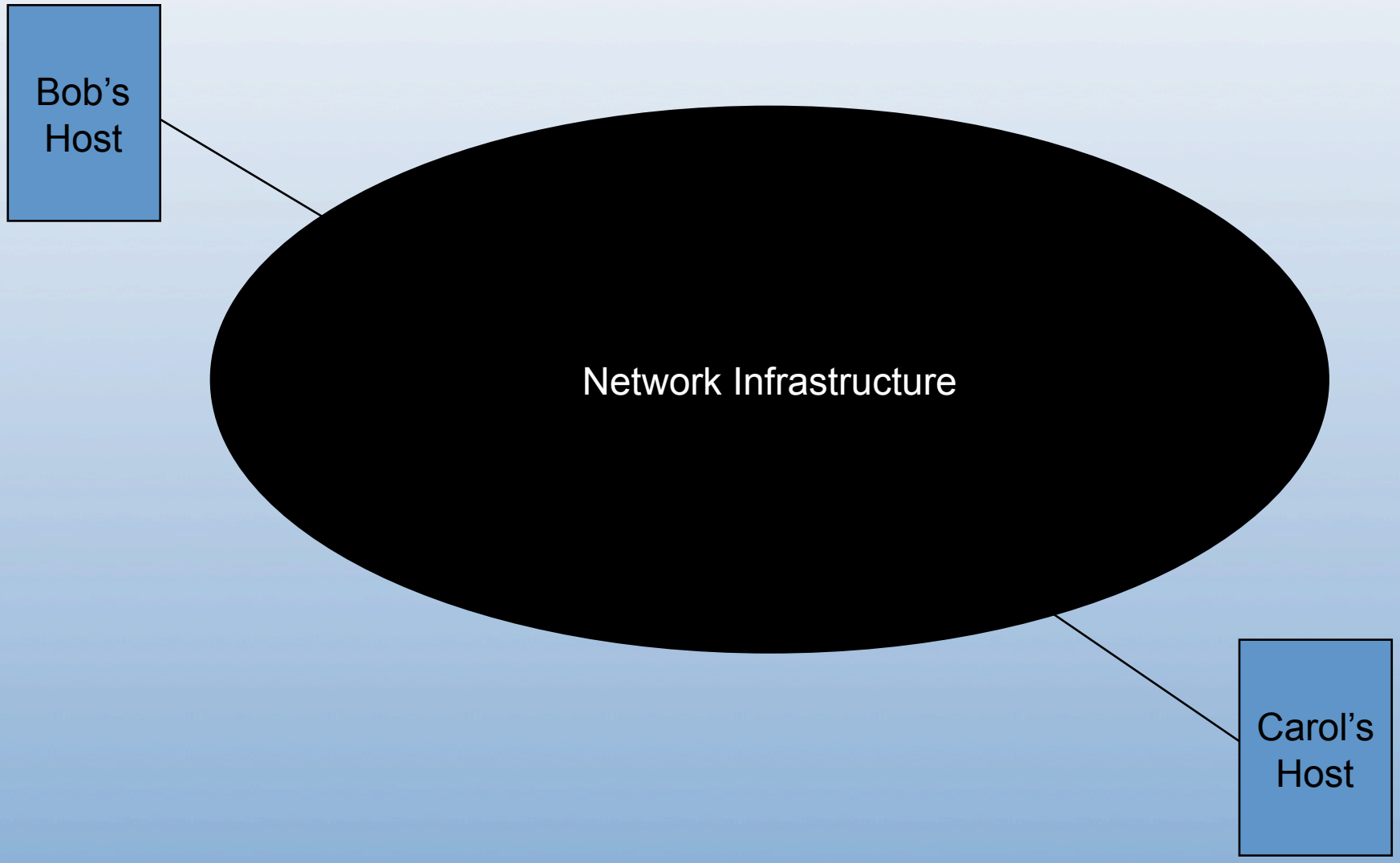
Advanced user tools

- 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

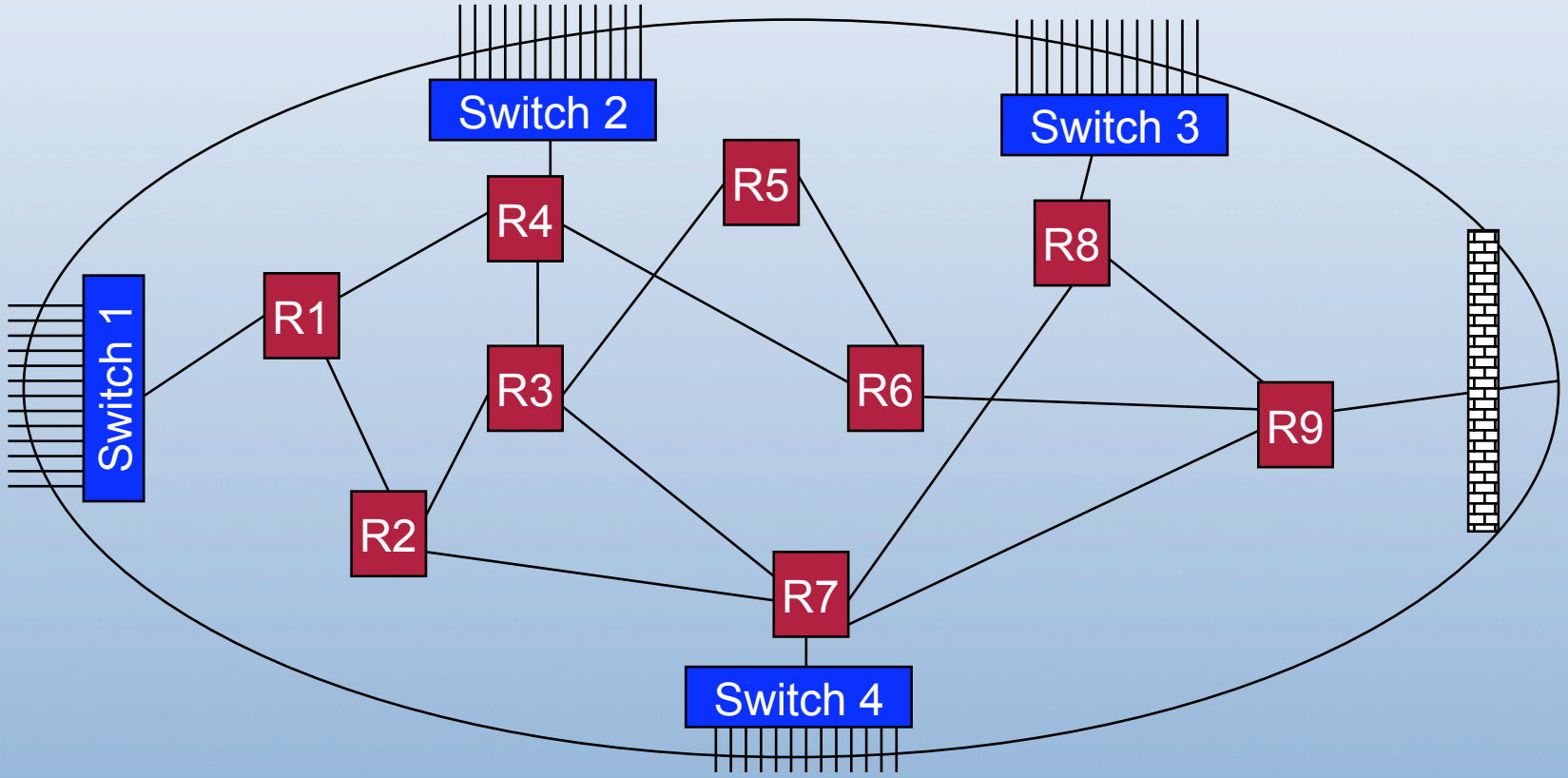
Underlying User Assumption

- When problems exist, it's the networks fault!

Simple Network Picture



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?

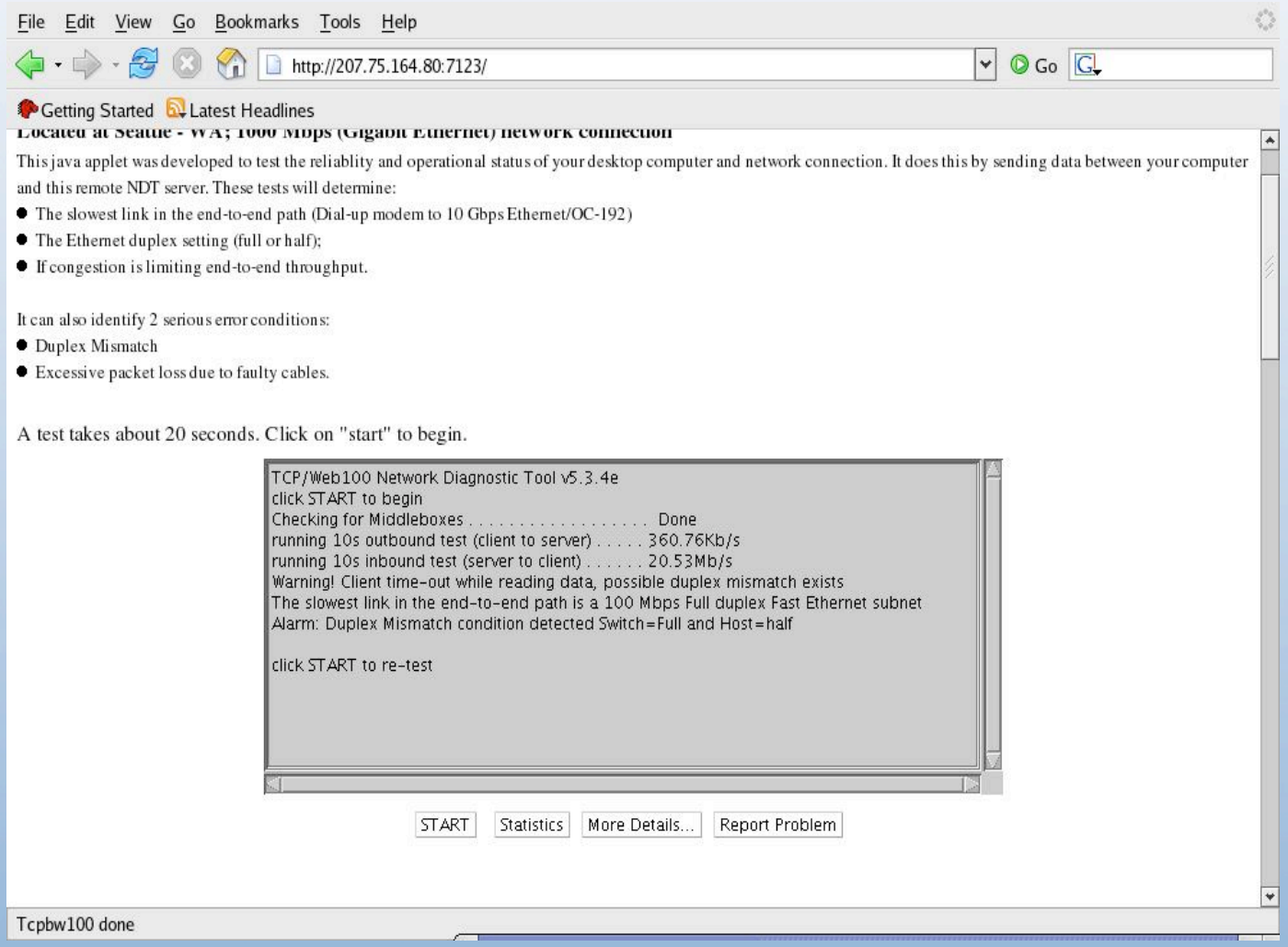
NDT Goals

- 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



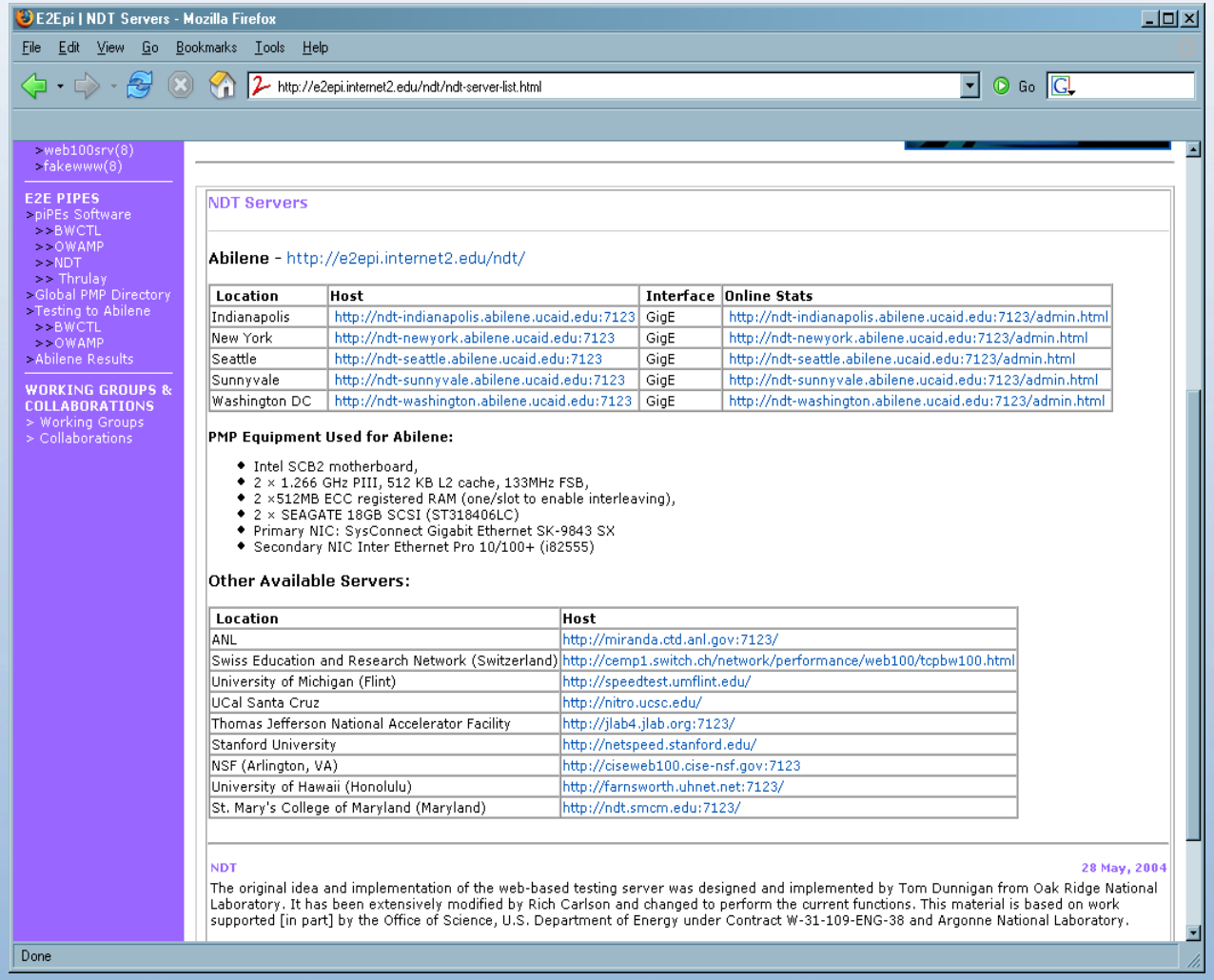
Finding Results of Interest

- 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



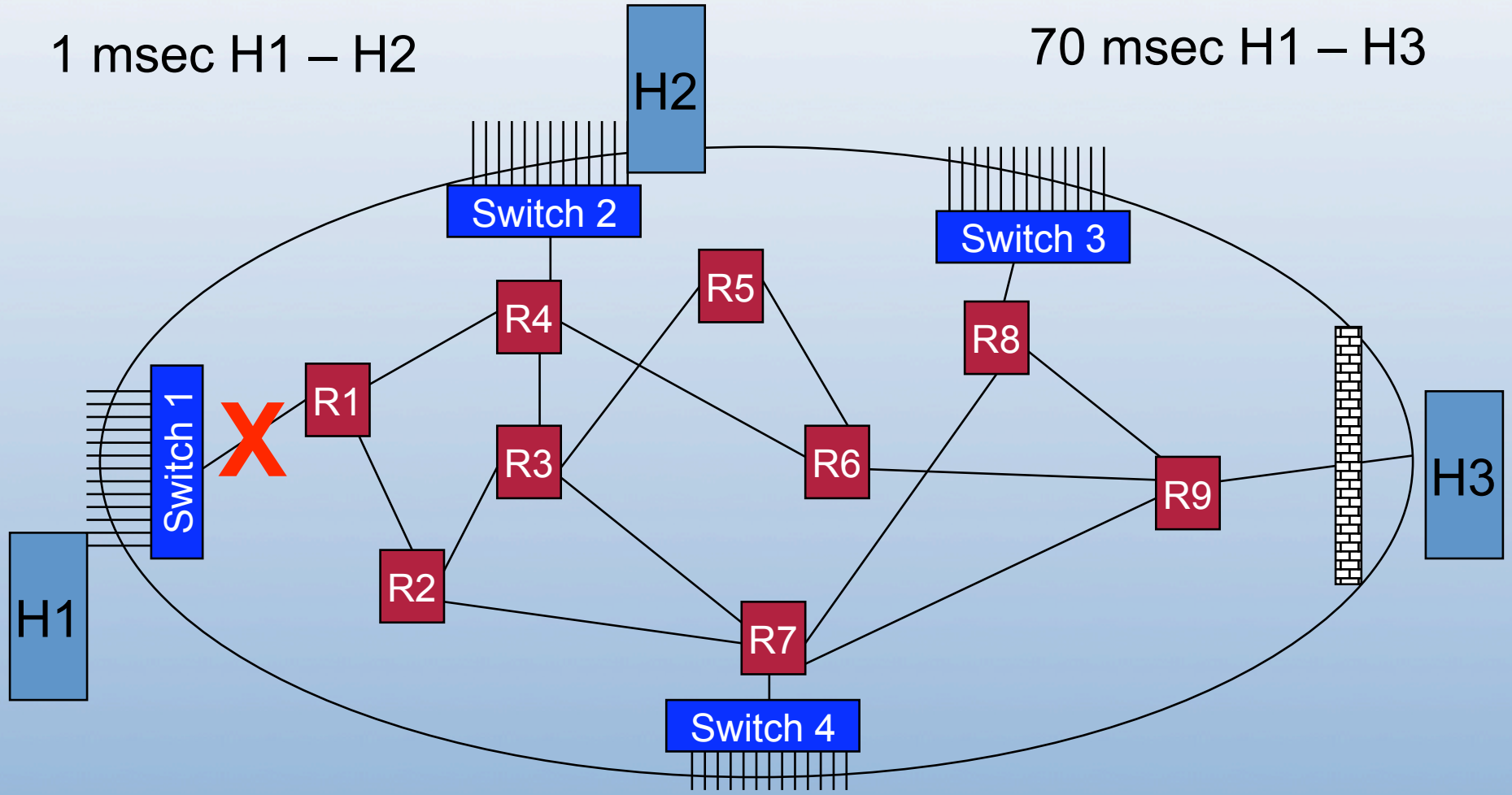
NPAD/pathdiag: What is it?

- 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



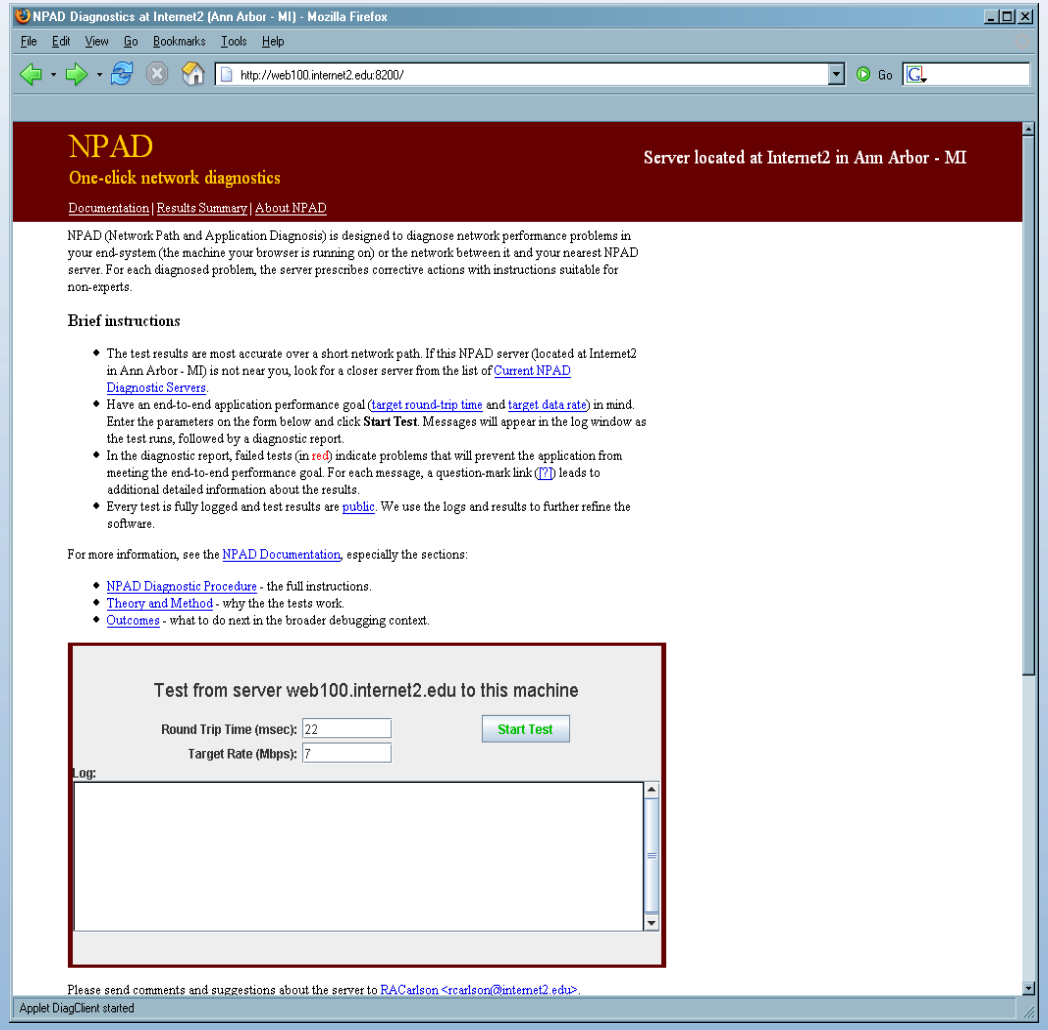
Long Path Problem

- E2E application performance is dependant 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

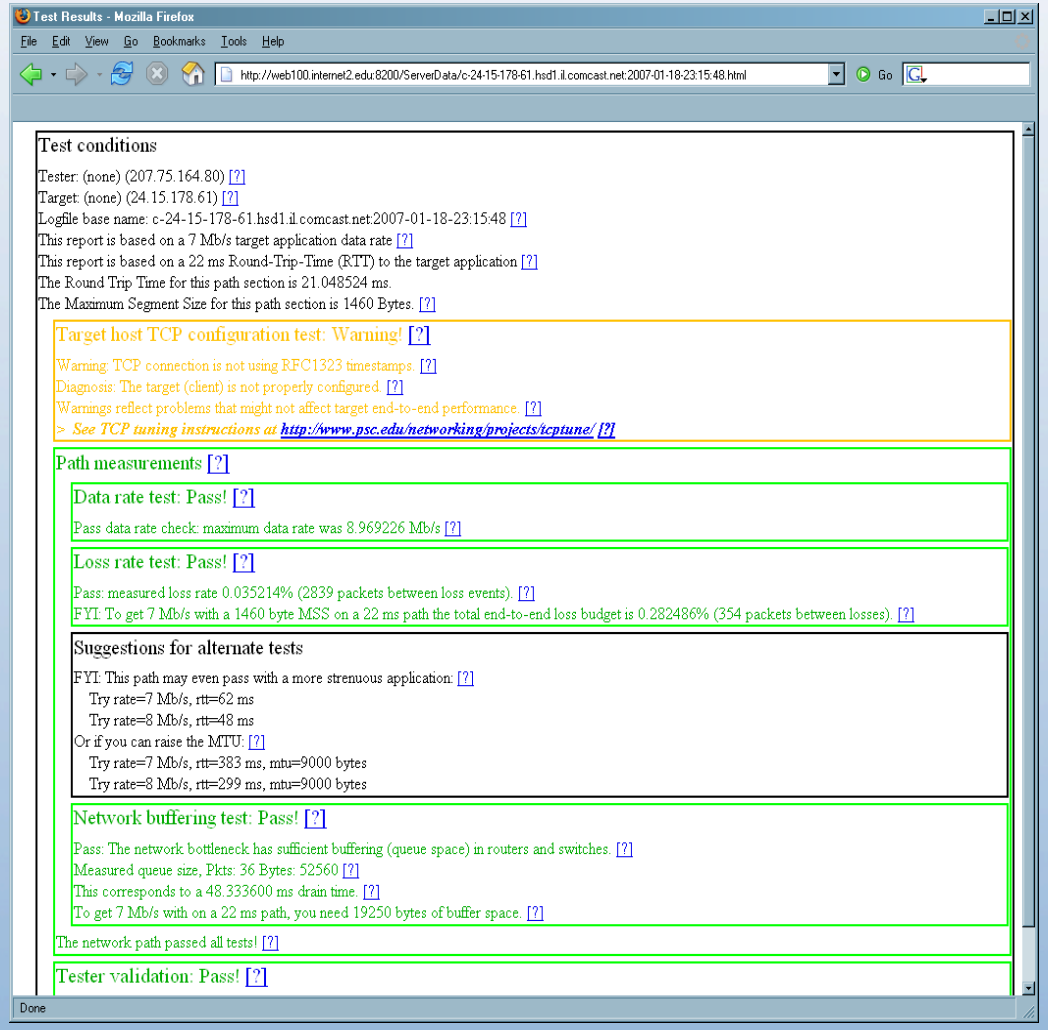
TCP Congestion Avoidance

- Cut number of packets by ½
- 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

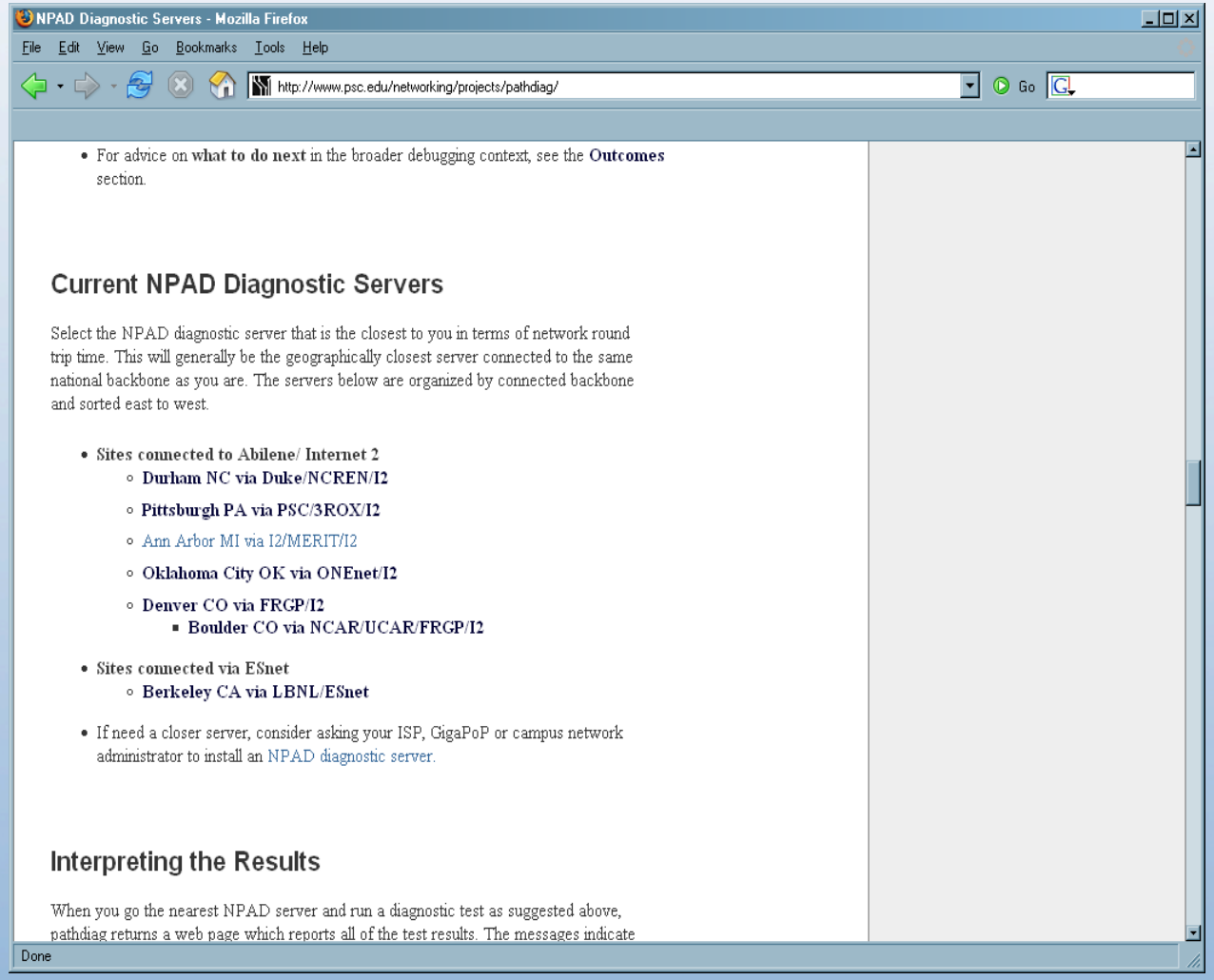
NPAD Server main page



NPAD Sample results



Finding NPAD servers



Try these tools

Network Performance Toolkit

http://e2epi.internet2.edu/network-performance-toolkit.html

Knoppix disk (OS on a CD) that has:

Iperf, thrulay, bwctl, owamp, NDT, NPAD, reverse-traceroute/ping…

Questions?

www.internet2.edu