TEIN2 Kick Off Workshop

Yasuichi Kitamura (kita@jp.apan.net)
APAN Tokyo XP/NICT

iperf

bandwidth

- wire rate
  - the bandwidth of the network device interface
- available bandwidth
  - the maximum bandwidth from one host to the other
  - the rest of the bandwidth in the actual connetion

Why you need to check the available bandwidth?

- Wire rate doesn't mean the available bandwidth.
- The performance of routers and switches is the key of the available bandwidth.
- To improve the performance based on TCP

iperf

- server(?) client(?) style application
  - same as ttcp
- http://dast.nlanr.net/Projects/Iperf/
  - source code and binary

bwtcl

BWCTL (Bandwidth Test Control)

Jeff Boote (boote@internet2.edu)

Network Performance Workshop

10-Jun-05

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

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

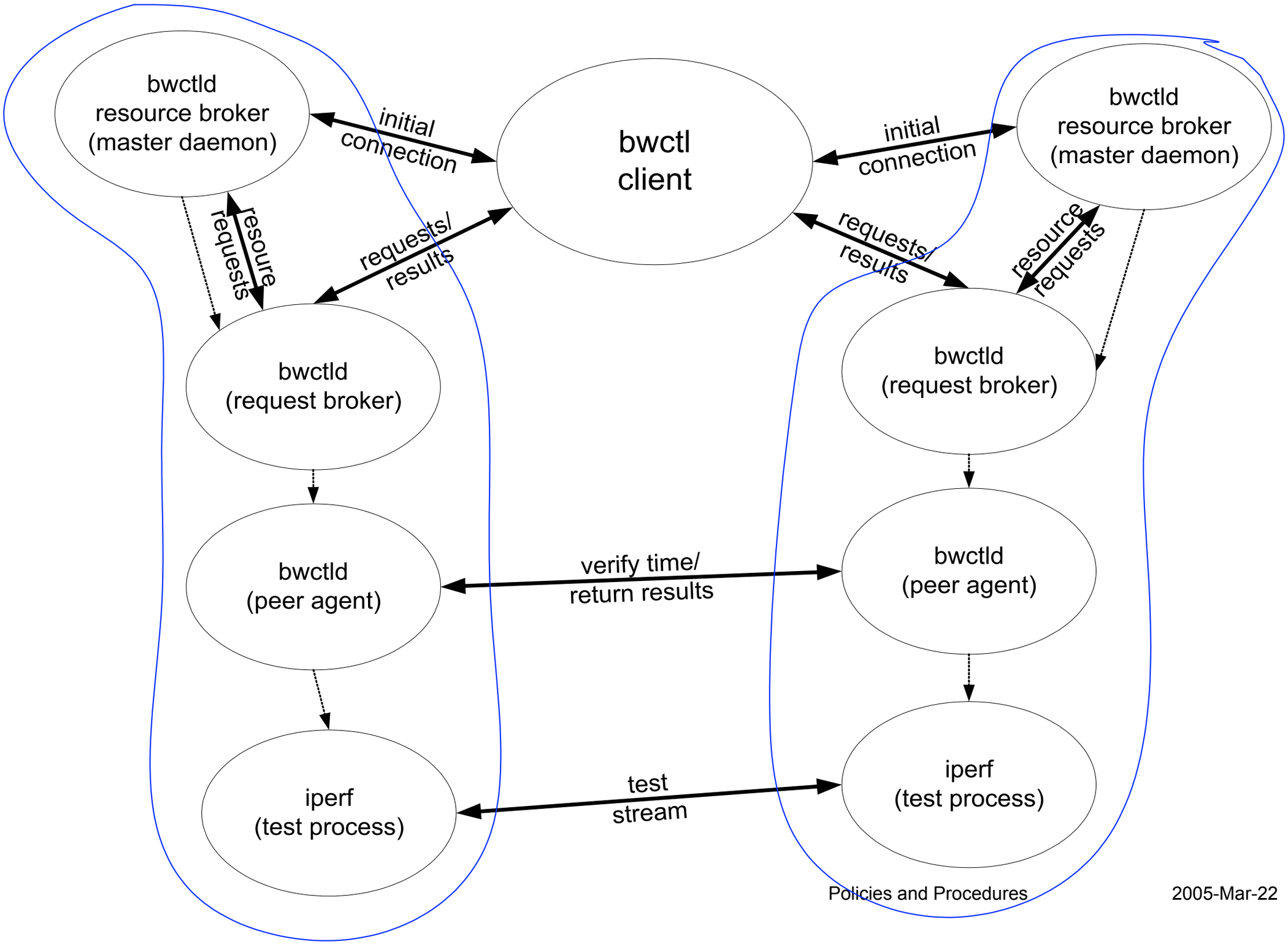
Scheduling

A time slot is simply a time-dependant resource that needs to be allocated just like any other resource. It therefore follows the resource allocation model.

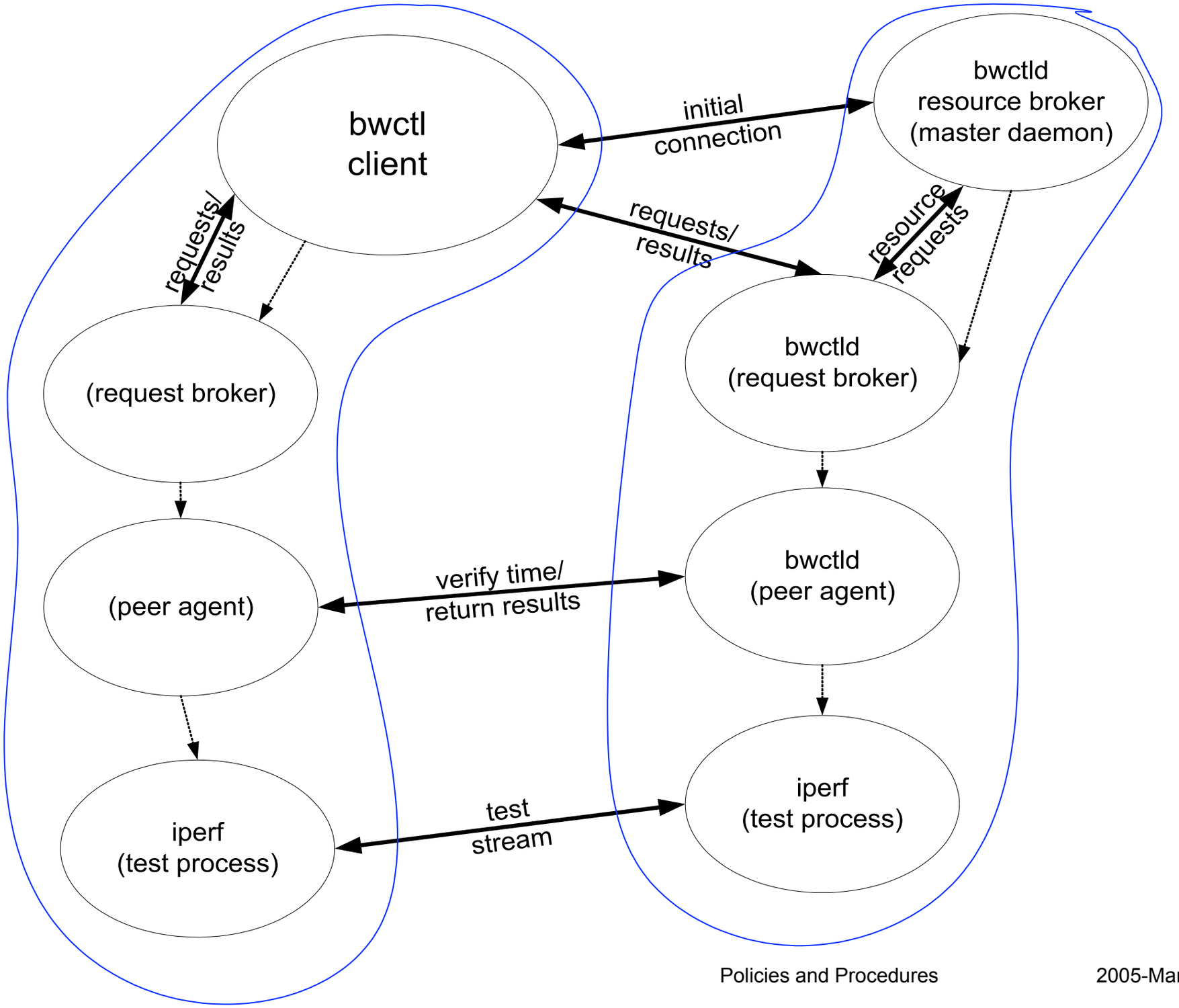
Resource Allocation (bwctld)

- 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)

BWCTL: 3-party Interaction



BWCTL: No Local Server



Iperf is the “tester”

- Well known – widely used
- Problems of integration
  - Iperf server initialization (port number allocation)
  - Iperf error conditions
  - End of session
  - No indication of partial progress (How full was the send buffer when the session was killed?)

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)

Recommended Hardware

- Highly dependent upon the network tests
  - Any system that can support an iperf test of a given intensity will be able to handle the additional burden of BWCTL
- To support 990 Mbps TCP flows on Abilene we use:
  - Intel SCB2 motherboard
    - 2 x 1.266 GHz PIII, 512 KB L2 cache, 133 MHz FSB
    - 2 x 512 MB ECC registered RAM (one/slot to enable interleaving)
    - 2 x Seagate 18 GB SCSI (ST318406LC)
    - SysConnect Gigabit Ethernet SK-9843 SX

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 statstics 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

www.internet2.ed
u

APAN area

- http://www.jp.apan.net/NOC/bwctl/
- Get the key today

OWAMP

OWAMP (One-Way Active Measurement Protocol)

Jeff Boote (boote@internet2.edu)

Network Performance Workshop

10-Jun-2005

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/internet-drafts/draft-ietf-ippm-owdp-014.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

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 reciever
- 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

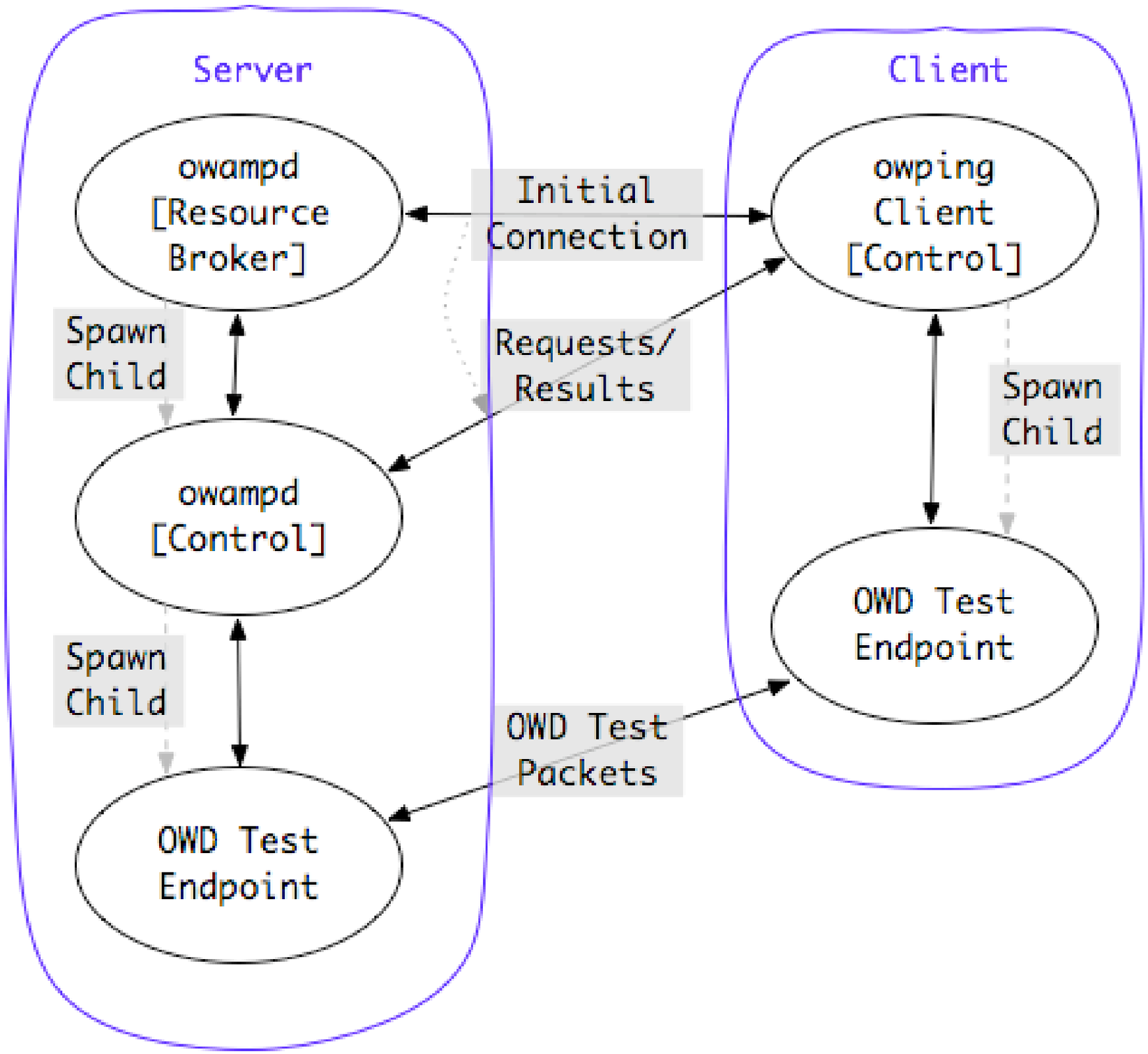
- Accepts requests for OWD tests
- Responds with accepted/denied
- Tests are formally started with a StartSessions message from the client.
- Runs tests
- Sessions with packets received at the server are buffered for later retrieval

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…
- NTP system calls available
- gnumake for build process

Supported Systems

- FreeBSD 4.x, 5.x
- Linux 2.4,2.6
- (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

Example Hardware

- Intel SCB2 motherboard
  - 2 x 1.266 GHz PIII, 512 KB L2 cache, 133 MHz FSB
  - 2 x 512 MB ECC registered RAM (one/slot to enable interleaving)
  - 2 x Seagate 18 GB SCSI (ST318406LC) Inter Ethernet Pro
  - 10/100+ (i82555) (on-motherboard)

We use these systems to support more than 44 concurrent streams of 10 packets/second

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 reciever

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

Methodological Errors

Our tests indicate a methodological error of 73 usec *

- Experiments with two systems connected via cross-over cable
- Two concurrent sessions (send,recv)
  - 10 packets/second
    - Intel SCB2 motherboard
    - 2x512 MB ECC registered RAM
    - Intel PRO/100+ integrated NIC

* 95% confidence level (RFC 2679)

* Error is specific to this hardware/intensity level

* Old version of owamp, should be even better now.

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

www.internet2.ed
u

Precision Related Context Switches