



OWAMP (One-Way Active Measurement Protocol)

Jeff Boote (boote@internet2.edu)

Network Performance Workshop

05-Apr-2006

Or... one-way ping

What is it?

OWD or One-Way PING

- A control protocol
- A test protocol
- A sample implementation of both

One-Way latencies or Delay. Think of it as one-way ping.

OWAMP itself is a control protocol, a test packet format, and a sample implementation.

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>

The solution is standards.



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

2006-Apr-05 | 4

Basically the client makes requests for tests with a server.

- Packets can be “open”, “authenticated”, or “encrypted”

The protocol is effectively a packet format.

Applications

- owampd daemon
- owping client

Built upon protocol abstraction library

- Supports one-off applications
- Allows authentication/policy hooks to be incorporated

There is a one-off application in the distribution called powstream that creates a perpetual stream of packets from the server to the client for continuous testing.

- 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

The command-line arguments were made as similar to ping as possible.

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

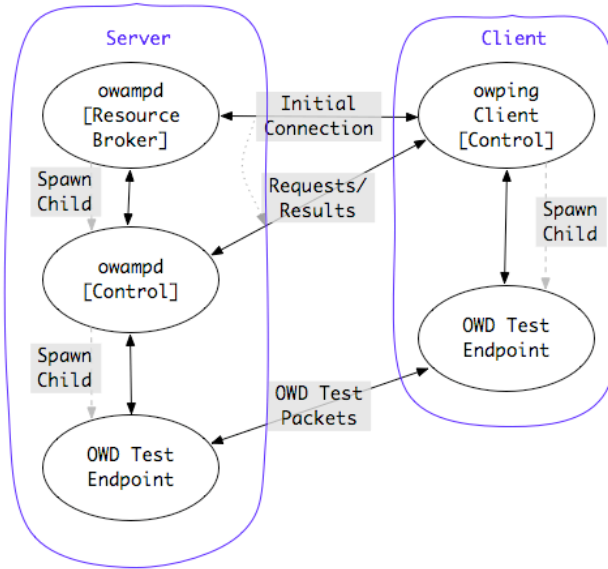
Standard accept/fork style unix daemon

- 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)

The parent owampd keeps track of current resource utilization needed to implement policy.

Architecture



See <http://e2epi.internet2.edu/owamp/details.html>

- 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

Getting a good stable NTP configuration is the most challenging task for obtaining good owamp results.



Supported Systems

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

- 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

Stable system clock is the most important feature.

- 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

The 44 concurrent streams represent intra-abilene testing. Hosts doing tests with external hosts are doing more.

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

These issues will be discussed further in the hands-on session.

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

DoS source:

A compromised owampd server could be used to send packets toward others. The implementation ensures that sessions can not be directed to random hosts in unauthenticated mode. (Only toward the OWAMP-control client.) Reasonable bandwidth limits and well protected AES keys should limit this risk.

DoS target:

Packets directed toward an owampd server can/will affect the precision of the valid test traffic.

Someone might try to effect data plots by targetting hosts that do owamp measurements.

Resource Consumption:

owampd has policy controls to allocate resources to appropriate users.

- 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

On Abilene, we attempt to be open until we can't.

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.

You should expect your results to be valid within 100usec's of the error reported.
(The error reported represents the NTP error, but does not include this error.)



Availability

- Currently available

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

Mail lists:

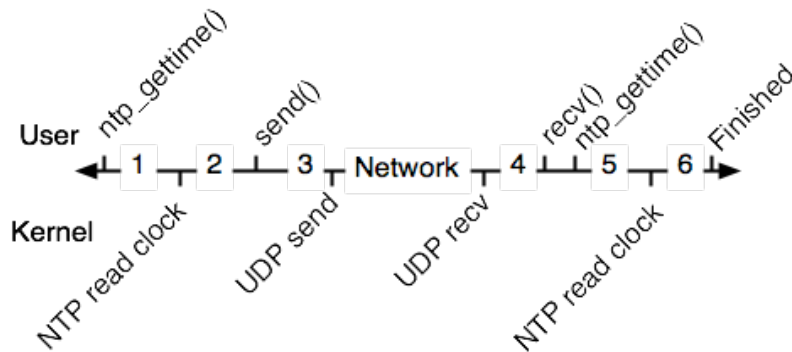
- owamp-users@internet2.edu
- owamp-announce@internet2.edu

<https://mail.internet2.edu/wss/lists/engineering>

INTERNET[®]



www.internet2.edu



This slide just illustrates where most of the methodological error comes from, it is not strictly important to understand from an operational point of view.

There are 6 context switches from when the sender process first requests the time until the receiver process has all the data.

The context switches that affect precision are 2–5 because they occur between the fetching of the system clock values on the respective systems.

Implementation decision was made to do timing in user space to allow for more portability. (custom kernel's not required)

It would be possible to completely remove these 4 context switches if the implementation created its own system call that combined the functionality of reading the timestamp with the sending and receiving of the UDP packet.

This is a possible future enhancement that could be done to improve worst-case accuracy for the subset of installations that need more precision.

Don't go into this unless there is interest and questions.