Network Working Group Stanislav Shalunov
Internet Draft Internet2
Expiration Date: December 2002 Benjamin Teitelbaum
Advanced Network & Services and Internet2
Matthew J. Zekauskas
Advanced Network & Services
June 2002

A One-way Active Measurement Protocol
<draft-ietf-ippm-owdp-04.txt>

1. Status of this Memo

2. Motivation and Goals

The IETF IP Performance Metrics (IPPM) working group has proposed draft standard metrics for one-way packet delay [RFC2679] and loss [RFC 2680] across Internet paths. Although there are now several measurement platforms that implement collection of these metrics [SURVEYOR], [RIPE], there is not currently a standard that would permit initiation of test streams or exchange of packets to collect singleton metrics in an interoperable manner.

With the increasingly wide availability of affordable global positioning system (GPS) and CDMA based time sources, hosts increasingly have available to them very accurate time sources--either directly or through their proximity to NTP primary (stratum 1) time servers. By standardizing a technique for collecting IPPM one-way active measurements, we hope to create an environment where IPPM metrics may be collected across a far broader mesh of Internet paths than is currently possible. One particularly compelling vision is of widespread deployment of open OWAMP servers that would make measurement of one-way delay as commonplace as measurement of round-trip time using an ICMP-based tool like ping.

Additional design goals of OWAMP include being hard to detect and manipulate, security, logical separation of control and test functionality, and support for small test packets.

OWAMP test traffic is hard to detect, because it is simply a stream of UDP packets from and to negotiated port numbers with potentially nothing static in the packets (size is negotiated, too). Additionally, OWAMP supports an encrypted mode, that further obscures the traffic, at the same time making it impossible to alter timestamps undetectably.

Security features include optional authentication and/or encryption of control and test messages. These features may be useful to prevent unauthorized access to results or man-in-the-middle attackers who attempt to provide special treatment to OWAMP test streams or who attempt to modify sender-generated timestamps to falsify test results.

2.1. Relationship of Test and Control Protocols

OWAMP actually consists of two inter-related protocols: OWAMP-Control and OWAMP-Test. OWAMP-Control is used to initiate, start, stop and retrieve test sessions, while OWAMP-Test is used to exchange test packets between two measurement nodes.

Although OWAMP-Test may be used in conjunction with a control protocol other than OWAMP-Control, the authors have deliberately chosen to include both protocols in the same draft to encourage the implementation and deployment of OWAMP-Control as a common denominator control protocol for one-way active measurements. Having a complete and open one-way active measurement solution that is simple to implement and deploy is crucial to assuring a future in which inter-domain one-way active measurement could become as

commonplace as ping. We neither anticipate nor recommend that OWAMP-Control form the foundation of a general purpose extensible measurement and monitoring control protocol.

OWAMP-Control is designed to support the negotiation of one-way active measurement sessions and results retrieval in a straightforward manner. At session initiation, there is a negotiation of sender and receiver addresses and port numbers, session start time, session length, test packet size, the mean Poisson sampling interval for the test stream, and some attributes of the very general RFC 2330 notion of "packet type", including packet size and per-hop behavior (PHB) [RFC2474], which could be used to support the measurement of one-way active across diff-serv networks. Additionally, OWAMP-Control supports per-session encryption and authentication for both test and control traffic, measurement servers which may act as proxies for test stream endpoints, and the exchange of a seed value for the pseudo-random Poisson process that describes the test stream generated by the sender.

We believe that OWAMP-Control can effectively support one-way active measurement in a variety of environments, from publicly accessible measurement "beacons" running on arbitrary hosts to network monitoring deployments within private corporate networks. If integration with SNMP or proprietary network management protocols is required, gateways may be created.

2.2. Logical Model

Several roles are logically separated to allow for broad flexibility in use. Specifically, we define:

Session-Sender — the sending endpoint of an OWAMP-Test session;

Session-Receiver — the receiving endpoint of an OWAMP-Test session;

Server — an end system that manages one or more OWAMP-Test sessions, is capable of configuring per-session state in session endpoints, and is capable of returning the results of a test session;

Control-Client — an end system that initiates requests for OWAMP-Test sessions, triggers the start of a set of sessions, and may trigger their termination;

Retrieve-Client — an end system that initiates requests to retrieve the results of completed OWAMP-Test sessions;

One possible scenario of relationships between these roles is shown below.

```
    +----------------+               +------------------+
    | Session-Sender |--OWAMP-Test-->| Session-Receiver |
    +----------------+               +------------------+
      ^                                     ^
      |                                     |
      |                                     |
      |                                     |
      |  +----------------+<----------------+
      |  |     Server     |<-------+
      |  +----------------+        |
      |    ^                       |
      |    |                       |
      | OWAMP-Control         OWAMP-Control
      |    |                       |
      v    v                       v
    +----------------+     +-----------------+
    | Control-Client |     | Retrieve-Client |
    +----------------+     +-----------------+
```

(Unlabeled links in the figure are unspecified by this draft and may be proprietary protocols.)

Different logical roles can be played by the same host. For example, in the figure above, there could actually be only two hosts: one playing the roles of Control-Client, Retrieve-Client, and Session-Sender, and the other playing the roles of Server and Session-Receiver. This is shown below.

```
    +-----------------+                   +------------------+
    | Control-Client  |<--OWAMP-Control-->| Server           |
    | Retrieve-Client |                   |                  |
    | Session-Sender  |---OWAMP-Test----->| Session-Receiver |
    +-----------------+                   +------------------+
```

Finally, because many Internet paths include segments that transport IP over ATM, delay and loss measurements can include the effects of ATM segmentation and reassembly (SAR). Consequently, OWAMP has been designed to allow for small test packets that would fit inside the payload of a single ATM cell (this is only achieved in unauthenticated and encrypted modes).

3. Protocol Overview

As described above, OWAMP consists of two inter-related protocols: OWAMP-Control and OWAMP-Test. The former is layered over TCP and is used to initiate and control measurement sessions and to fetch their results. The latter protocol is layered over UDP and is used to send singleton measurement packets along the Internet path under test.

The initiator of the measurement session establishes a TCP connection to a well-known port on the target point and this connection remains open for the duration of the OWAMP-Test sessions. IANA will be requested to allocate a well-known port number for OWAMP-Control sessions. An OWAMP server SHOULD listen to this well-known port.

OWAMP-Control messages are transmitted only before OWAMP-Test sessions are actually started and after they complete (with the possible exception of an early Stop-Session message).

The OWAMP-Control and OWAMP-Test protocols support three modes of operation: unauthenticated, authenticated, and encrypted. The authenticated or encrypted modes require endpoints to possess a shared secret.

All multi-octet quantities defined in this document are represented as unsigned integers in network byte order unless specified otherwise.

4. OWAMP-Control

Each type of OWAMP-Control message has a fixed length. The recipient will know the full length of a message after examining first 16 octets of it. No message is shorter than 16 octets.

If the full message is not received within 30 minutes after it is expected, connection SHOULD be dropped.

4.1. Connection Setup

Before either a Control-Client or a Retrieve-Client can issue commands of a Server, it must establish a connection to the server.

First, a client opens a TCP connection to the server on a well-known port. The server responds with a server greeting:

```
         0                   1                   2                   3
         0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |                                                               |
        |                      Unused (12 octets)                       |
        |                                                               |
        |+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-++-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |                            Modes                              |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |                                                               |
        |                     Challenge (16 octets)                     |
        |                                                               |
        |                                                               |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The following mode values are meaningful: 1 for unauthenticated, 2 for authenticated, 4 for encrypted. The value of the Modes field sent by the server is the bit-wise OR of the mode values that it is willing to support during this session. Thus, last three bits of the Modes 32-bit value are used. The first 29 bits MUST be zero. A client MUST ignore the values in the first 29 bits of the Modes value. (This way, the bits are available for future protocol extensions. This is the only intended extension mechanism.)

If Modes value is zero, the server doesn't wish to communicate with the client and MAY close the connection immediately. The client SHOULD close the connection if it gets a greeting with Modes equal to zero.

Otherwise, the client MUST respond with the following message:

```
        0                   1                   2                   3
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                              Mode                             |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                               |
       .                                                               .
       .                      Username (16 octets)                     .
       .                                                               .
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                               |
       .                                                               .
       .                        Token (32 octets)                      .
       .                                                               .
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                               |
       .                                                               .
       .                      Client-IV (16 octets)                    .
       .                                                               .
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Here Mode is the mode that the client chooses to use during this OWAMP-Control session. It will also be used for all OWAMP-Test sessions started under control of this OWAMP-Control session. In Mode, one or zero bits MUST be set within last three bits. The first 29 bits of Mode MUST be zero. A server MUST ignore the values of the first 29 bits.

In unauthenticated mode, Username, Token, and Client-IV are unused.

Otherwise, Username is a 16-octet indicator of which shared secret the client wishes to use to authenticate or encrypt and Token is the concatenation of a 16-octet challenge and a 16-octet Session-key, encrypted using the AES (Advanced Encryption Standard) [AES] in Cipher Block Chaining (CBC). Encryption MUST be performed using an Initialization Vector (IV) of zero and a key value that is the shared secret associated with Username. The shared secret will typically be provided as a passphrase; in this case, not the actual passphrase SHOULD be used as a key for encryption by the client and decryption by the server, but the MD5 sum [RFC1321] of the passphrase (without possible newline character(s) at the end of the passphrase; the passphrase also SHOULD not contain newlines).

Session-key and Client-IV are generated randomly by the client.

The server MUST respond with the following message:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                      Unused, MBZ (15 octets)                  |
|                                                               |
|                                               +-+-+-+-+-+-+-+-+
|                                               |   Accept      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                     Server-IV (16 octets)                     |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Unused 15-octet part MUST be zero. The server MUST ignore its value.

A zero value in the Accept field means that the server accepts the authentication and is willing to conduct further transactions. A value of 1 means that the server does not accept the authentication provided by the client or, for some other reason, is not willing to conduct further transactions in this OWAMP-Control session. All other values are reserved. The server MUST interpret all values of Accept other than 0 and 1 as 1. This way, other values are available for future extensions. If a negative response is sent, the server MAY and the client SHOULD close the connection after this message.

The previous transactions constitute connection setup.

4.2. OWAMP-Control Commands

In authenticated or encrypted mode (which are identical as far as OWAMP-Control is concerned, and only differ in OWAMP-Test) all further communications are encrypted with the Session-key, using CBC mode. The client encrypts its stream using Client-IV. The server encrypts its stream using Server-IV.

The following commands are available for the client: Request-Session, Start-Sessions, Stop-Session, Retrieve-Session. The command Stop-Session is available to both client and server.

After Start-Sessions is sent/received by the client/server, and before it both sends and receives Stop-Session (order unspecified), it is said to be conducting active measurements.

While conducting active measurements, the only command available is Stop-Session.

These commands are described in detail below.

4.3. Creating Test Sessions

Individual one-way active measurement sessions are established using a simple request/response protocol. An OWAMP client MAY issue zero or more Request-Session messages to an OWAMP server, which MUST respond to each with an Accept-Session message. An Accept-Session message MAY refuse a request.

The format of Request-Session message is as follows:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |      1        |IPVN-S | IPVN-R| Conf-Sender   | Conf-Receiver |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                        Sender Address                         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |              Sender Address (cont.) or Unused                 |
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                        Receiver Address                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |              Receiver Address (cont.) or Unused               |
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |          Sender Port          |         Receiver Port         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                        SID (16 octets)                        |
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                          Inv-Lambda                           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                            Packets                            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                         Padding Length                        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                           Start Time                          |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                       Type-P Descriptor                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                   Zero Padding (16 octets)                    |
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Here the first octet (1) indicates that this is Request-Session command.

IPVN-S and IPVN-R are IP version numbers for Sender and Receiver. In the case of IP version number being 4, twelve unused octets follow the four-octet address.

Conf-Sender and Conf-Receiver can be 0 or 1. If 1, the server is being asked to configure the corresponding agent (sender or receiver). In this case, the corresponding Port value SHOULD be disregarded by the server. At least one of Conf-Sender and Conf-Receiver MUST be 1. (Both can be set, in which case the server is being asked to perform a session between two hosts it can configure.)

The Sender Address and Receiver Address fields contain respectively the sender and receiver addresses of the end points of the Internet path over which an OWAMP test session is requested.

If Conf-Sender is not set, Sender Port is the UDP port OWAMP-Test packets will be sent from. If Conf-Receiver is not set, Receiver Port is the UDP port OWAMP-Test packets are requested to be sent to.

SID is the session identifier. It can be used in later sessions as an argument for Retrieve-Session command. It is meaningful only if Conf-Receiver is 1.

The field Inv-Lambda is an unsigned integer and is the scaled reciprocal of rate (in microseconds) at which the Poisson test stream is to be generated. This allows the average Poisson sampling interval for the requested test session to be set to between 1 microsecond and over an hour.

The value Packets is the number of active measurement packets to be sent during this OWAMP-Test session (note that both server and client can abort the session early).

Padding length is the number of octets to be appended to normal OWAMP-Test packet (see more on padding in discussion of OWAMP-Test).

Start Time is the time when the session is to be started (but not before Start-Sessions command is issued). This timestamp is in the same format as OWAMP-Test timestamps.

Type-P Descriptor covers only a subset of (very large) Type-P space. If the first two bits of Type-P Descriptor are 00, then subsequent 6 bits specify the requested Differentiated Services Codepoint (DSCP) value of sent OWAMP-Test packets as defined in RFC 2474. If the first two bits of Type-P descriptor are 01, then subsequent 16 bits specify the requested Per Hop Behavior Identification Code (PHB ID) as defined in RFC 2836.

Therefore, the value of all zeros specifies the default best-effort service.

If Conf-Sender is set, Type-P Descriptor is to be used to configure

the sender to send packets according to its value. If Conf-Sender is not set, Type-P Descriptor is a declaration of how the sender will be configured.

If Conf-Sender is set and the server doesn't recognize Type-P Descriptor, cannot or does not wish to set the corresponding attributes on OWAMP-Test packets, it SHOULD reject the session request. If Conf-Sender is not set, the server SHOULD accept the session regardless of the value of Type-P Descriptor.

Zero Padding MUST be all zeros in this and all subsequent messages that use zero padding. The recipient of a message where zero padding is not zero MUST reject the message as it is an indication of tampering with the content of the message by an intermediary (or brokenness). If the message is part of OWAMP-Control, the session MUST be terminated and results invalidated. If the message is part of OWAMP-Test, it MUST be silently ignored. This will ensure data integrity.

To each Request-Session message, an OWAMP server MUST respond with an Accept-Session message:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Accept     |  Unused       |            Port               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
|                                                               |
|                        SID (16 octets)                        |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                    Zero Padding (12 octets)                   |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Zero in the Accept field means that the server is willing to conduct the session. A value of 1 indicates rejection of the request. All other values are reserved.

If the server rejects a Request-Session command, it SHOULD not close the TCP connection. The client MAY close it if it gets negative response to Request-Session.

The meaning of Port depend on the values of Conf-Sender and Conf-Receiver in the query that solicited the response. If both were set, Port field is unused. If only Conf-Sender was set, Port is the port

to expect OWAMP-Test packets from. If only Conf-Receiver was set, Port is the port to send OWAMP-Test packets to.

If only Conf-Sender was set, SID is unused. Otherwise, SID is a unique server-generated session identifier. It can be used later as handle to retrieve the results of a session.

SIDs SHOULD be constructed by concatenation of 4-octet IPv4 IP number belonging to the generating machine, 8-octet timestamp, and 4-octet random value. Note that SID is always chosen by the receiver.

4.4. Starting Test Sessions

Having requested one or more test sessions and received affirmative Accept-Session responses, an OWAMP client may start the execution of the requested test sessions by sending a Start-Sessions message to the server.

The format of this message is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      2        |                                               |
+-+-+-+-+-+-+-+-+                                               |
|                      Unused (15 octets)                       |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                    Zero Padding (16 octets)                   |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The server MUST respond with an Control-Ack message (which SHOULD be sent as quickly as possible). Control-Ack messages have the following format:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Accept    |                                               |
    +-+-+-+-+-+-+-+-+                                               |
    |                       Unused (15 octets)                      |
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                     Zero Padding (16 octets)                  |
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

If Accept is 1, the Start-Sessions request was rejected; zero means that the command was accepted. All other values are reserved. The server MAY and the client SHOULD close the connection in the case of a negative response.

The server SHOULD start all OWAMP-Test streams immediately after it sends the response or immediately after their specified start times, whichever is later. (Note that a client can effect an immediate start by specifying in Request-Session a Start Time in the past.) If the client represents a Sender, the client SHOULD start its OWAMP-Test streams immediately after it sees the Control-Ack response from the Server.

4.5. Stop-Sessions

The Stop-Sessions message may be issued by either the Control-Client or the Server. The format of this command is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      3        |    Accept     |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                         Unused (14 octets)                    |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                      Zero Padding (16 octets)                 |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Normally, the client SHOULD send this command after the OWAMP-Test streams have completed. However, either client or server MAY send it prematurely.

Value of 1 of Accept indicates a failure of some sort. Zero values indicates normal (but possibly premature) completion. All other values are reserved. If Accept had non-zero value (from either party), or if it was not transmitted at all (for whatever reason, including TCP connection used for OWAMP-Control breaking), results of all OWAMP-Test sessions spawned by this OWAMP-Control session SHOULD be considered invalid, even if Retrieve-Session with SID from this session works during a different OWAMP-Control session.

The party that receives this command MUST stop its OWAMP-Test streams and respond with a Stop-Sessions message. Any non-zero value in Accept field means something went wrong. A zero value means OWAMP-Test streams have been successfully stopped.

4.6. Retrieve-Session

The format of this client command is as follows:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |      4        |                                               |
   +-+-+-+-+-+-+-+-+                                               |
   |                      Unused (17 octets)                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          Begin Seq                            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                           End Seq                             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   |                        SID (16 octets)                        |
   |                                                               |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   |                    Zero Padding (16 octets)                   |
   |                                                               |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Begin Seq is the sequence number of the first requested packet. End Seq is the sequence number of the last requested packet. If Begin Seq is all zeros and End Seq is all ones, complete session is said to be requested.

If a complete session is requested and the session is still in progress, or has terminated in any way other than normal, the request to retrieve session results MUST be denied. If an incomplete session is requested, all packets received so far that fall into the requested range SHOULD be returned.

The server MUST respond with a Control-Ack message. Again, 1 in the Accept field means rejection of command. Zero means that data will follow. All other values are reserved.

If Accept was 0, the server then MUST send the OWAMP-Test session data in question, followed by 16 octets of zero padding.

The transmission starts with 4 octets that contain the number of records that will follow, each record representing one received packet. This is followed by 4 octets of Type-P Descriptor and 8 octets of zero padding.

Each packet is represented with 20 octets, and includes 4 octets of sequence number, 8 octets of send timestamp, and 8 octets of receive timestamp.

The last (possibly full, possibly incomplete) block (16 octets) of data is padded with zeros if necessary. A zero padding consisting of 16 octets is then appended.

5. OWAMP-Test

This section describes OWAMP-Test protocol. It runs over UDP using sender and receiver IP and port numbers negotiated during Session-Prepare exchange.

As OWAMP-Control, OWAMP-Test has three modes: unauthenticated, authenticated, and encrypted. All OWAMP-Test sessions spawned by an OWAMP-Control session inherit its mode.

OWAMP-Control client, OWAMP-Control server, OWAMP-Test sender, and OWAMP-Test receiver can potentially all be different machines. (In a typical case we expect that there will be only two machines.)

5.1. Sender Behavior

The sender sends the receiver a stream of packets with exponential distribution of times between packets. The format of the body of a UDP packet in the stream depends on the mode being used.

For unauthenticated mode:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Timestamp                            |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                                                               .
.                    Packet padding (0-65515 octets)            .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

For authenticated mode:

```
      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                        Sequence Number                        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                                               |
     |                   Zero Padding (12 octets)                    |
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                          Timestamp                            |
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                                               |
     .                                                               .
     .                   Packet padding (0-65503 octets)             .
     .                                                               .
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

For encrypted mode:

```
      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                        Sequence Number                        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                          Timestamp                            |
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                         Zero Padding                          |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                                               |
     .                                                               .
     .                   Packet padding (0-65511 octets)             .
     .                                                               .
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The format of the timestamp is influenced by RFC 1305 and is as follows: first 32 bits represent the unsigned integer number of seconds elapsed since 0h on 1 January 1900; next 24 bits represent the fractional part of a second that has elapsed since then (so, first 56 bits of the timestamp would be the same as the corresponding bits of NTP v3 timestamp). The remaining octet specifies synchronization and precision. The first bit is set if the party generating the timestamp has a clock that is synchronized to an external source (e.g., the bit should be set if GPS hardware is used and it indicates that it has acquired current position and time or if

NTP is used and it indicates that it has synchronized to an external source, which includes stratum 0 source, etc.); if there is no notion of external synchronization for the time source (e.g., a cesium oscillator is used directly), the bit SHOULD be set. The next bit is currently unused and may be set to an arbitrary value. The remaining six bits form an unsigned integer, which is the number of bits in the time-specifying main part of the timestamp that the party generating timestamp believes to be correct (this should be set conservatively). When generating a timestamp, one MUST ensure that this number falls into the range from 0 to 56; when interpreting a timestamp, one MUST treat numbers in the range 57 to 63 identically to the number 56.

More rigorous semantics of precision indicators are out of scope of OWAMP, but may be negotiated out-of-band.

So, timestamp is represented as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Integer part of seconds                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Fractional part of seconds                |S|U| Prec      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where S is the synchronization bit, U is currently unused, and Prec is the unsigned integer in the range from 0 to 56 discussed above.

Sequence numbers start with 0 and are incremented by 1 for each subsequent packet.

The minimum data segment length is therefore 12 octets in unauthenticated mode, 24 octets in authenticated mode, and 16 octets in encrypted mode.

In authenticated and encrypted mode, the first block (16 octets) of each packet is encrypted using AES ECB mode.

In unauthenticated mode, no encryption is applied.

The time elapsed between packets is pseudo-random, with exponential distribution (resulting in a Poisson stream of packets). As suggested in RFC 2330, the ith sampling interval Ei may be computed using inverse transform:

    Ei = -ln(Ui) * Inv-Lambda

where Ui is uniformly distributed between 0 and 1 and lambda is the desired mean time between packets.

Pseudo-random stream of bits is obtained using AES with SID as the key, running in counter mode (first encrypted block is 0, second encrypted block is 1 in network octet order, etc.) Each block of 64 bits is used to obtain one pseudo-random number uniformly distributed between 0 and 1. If the bits are Bj (j=1..64, numbered left to right), the resulting value is

U = B1*2^{-1} + B2*2^{-2} + ... B64*2^{-64}

The parameter lambda is has the value requested in the Request-Session message of the OWAMP-Control negotiation that spawned the session.

The logarithm and division in the formula above MUST be computed using IEEE 754 standard floating point arithmetic. [HELP WANTED!: Someone with a stronger background in numerical analysis to specify how to compute the sampling intervals precisely and portably!]

Finally, Packet Padding SHOULD be pseudo-random (generated independently of any other pseudo-random numbers mentioned in this document). However, implementations MUST provide a configuration parameter, an option, or a different means of making Packet Padding consist of all zeros.

5.2. Receiver Behavior

Receiver knows when the sender will send packets. The following parameter is defined: loss threshold. It SHOULD be 10 minutes and MAY be more, but not more than 60 minutes.

As packets are received,

+ Timestamp the received packet.

+ In authenticated or encrypted mode, decrypt first block (16 octets) of packet body.

+ Store the packet sequence number, send times, and receive times for the results to be transferred.

+ Packets not received within the loss threshold are considered lost. They are recorded with their seqno, presumed send time, and receive time consisting of a string of zero bits.

Packets that have send time in the future MUST be recorded normally, without changing their send timestamp, unless they have to be discarded.

If any of the following is true, packet MUST be discarded:

+ Send timestamp is more than loss threshold in the past or in the future.

+ Send timestamp differs by more than loss threshold from the time when the packet should have been sent according to its seqno.

+ In authenticated or encrypted mode, any of the bits of zero padding inside the first 16 octets of packet body is non-zero.

6. Security Considerations

The goal of authenticated mode to let one passphrase-protect service provided by a particular OWAMP-Control server. One can imagine a variety of circumstances where this could be useful. Authenticated mode is designed to prohibit theft of service.

Additional design objective of authenticated mode was to make it impossible for an attacker who cannot read traffic between OWAMP-Test sender and receiver to tamper with test results in a fashion that affects the measurements, but not other traffic.

The goal of encrypted mode is quite different: To make it hard for a party in the middle of the network to make results look "better" than they should be. This is especially true if one of client and server doesn't coincide with neither sender nor receiver.

Encryption of OWAMP-Control using AES CBC mode with blocks of zeros after each message aims to achieve two goals: (i) to provide secrecy of exchange; (ii) to provide authentication of each message.

OWAMP-Test sessions directed at an unsuspecting party could be used for denial of service (DoS) attacks. In unauthenticated mode servers should limits receivers to hosts they control or to the OWAMP-Control client.

OWAMP-Test sessions could be used as covert channels of information. Environments that are worried about covert channels should take this into consideration.

Notice that AES in counter mode is used for pseudo-random number generation, so implementation of AES MUST be included even in a server that only supports unauthenticated mode.

7. References

[AES]   Advanced Encryption Standard (AES),
http://csrc.nist.gov/encryption/aes/

[RFC1305]D. Mills, "Network Time Protocol (Version 3) Specification, Implementation and Analysis", RFC 1305, March 1992.

[RFC1321]   R. Rivest, "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[RFC2026]S. Bradner, "The Internet Standards Process -- Revision 3", RFC 2026, October 1996.

[RFC2119]S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

[RFC2330]   V. Paxon, G. Almes, J. Mahdavi, M. Mathis, "Framework for IP Performance Metrics" RFC 2330, May 1998.

[RFC2474]   K. Nichols, S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[RFC2679]G. Almes, S. Kalidindi, and M. Zekauskas, "A One-way Delay Metric for IPPM", RFC 2679, September 1999.

[RFC2680]G. Almes, S. Kalidindi, and M. Zekauskas, "A One-way Packet Loss Metric for IPPM", RFC 2680, September 1999.

[RFC2836]S. Brim, B. Carpenter, F. Le Faucheur, "Per Hop Behavior Identification Codes", RFC 2836, May 2000.

[RIPE]   RIPE NCC Test-Traffic Measurements home,
http://www.ripe.net/test-traffic/.

[RIPE-NLUUG]H. Uijterwaal and O. Kolkman, "Internet Delay Measurements Using Test-Traffic", Spring 1998 Dutch Unix User Group Meeting, http://www.ripe.net/test-traffic/Talks/9805_nluug.ps.gz.

[SURVEYOR]   Surveyor Home Page, http://www.advanced.org/surveyor/.

[SURVEYOR-INET]S. Kalidindi and M. Zekauskas, "Surveyor: An Infrastructure for Network Performance Measurements", Proceedings of INET'99, June 1999.
http://www.isoc.org/inet99/proceedings/4h/4h_2.htm

8. Authors' Addresses

Stanislav Shalunov
Internet2 / UCAID
200 Business Park Drive
Armonk, NY 10504
USA

Phone: +1 914 765 1182
EMail: shalunov@internet2.edu

Benjamin Teitelbaum
Advanced Network & Services
200 Business Park Drive
Armonk, NY 10504
USA

Phone: +1 914 765 1118
EMail: ben@advanced.org

Matthew J. Zekauskas
Advanced Network & Services, Inc.
200 Business Park Drive
Armonk, NY 10504
USA

Phone: +1 914 765 1112
EMail: matt@advanced.org

Expiration date: December 2002