Network Working Group
Internet Draft
Expiration Date: June 2001

S. Shalunov
Internet2
B. Teitelbaum
Advanced Network & Services and Internet2
M. Zekauskas
Advanced Network & Services

December 2000

A One-way Delay Measurement Protocol
<draft-ietf-ippm-owdp-01.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 delay 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 OWDP 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 OWDP include stealth, security, logical separation of control and test functionality, and support for small test packets.

Stealth is achieved by making test packet streams look as much as possible like ordinary Internet traffic. Towards this goal, OWDP's test protocol is layered over UDP and allows for a wide range of packet sizes and port numbers. Additionally, OWDP supports an encrypted mode that obscures all transmitted data, making detection of OWDP test activity by Internet service providers very difficult.

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 OWDP test streams or who attempt to modify sender-generated timestamps to falsify test results.

OWDP actually consists of two inter-related protocols: OWDP-Control and OWDP-Test. OWDP-Control is used to initiate, start, stop and retrieve test sessions, while OWDP-Test is the actual one-way delay test protocol that exchanges singleton test packets between two measurement nodes.

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

Session-Sender the sending endpoint of an OWDP-Test session;

Session-Receiver the receiving endpoint of an OWDP-Test session;

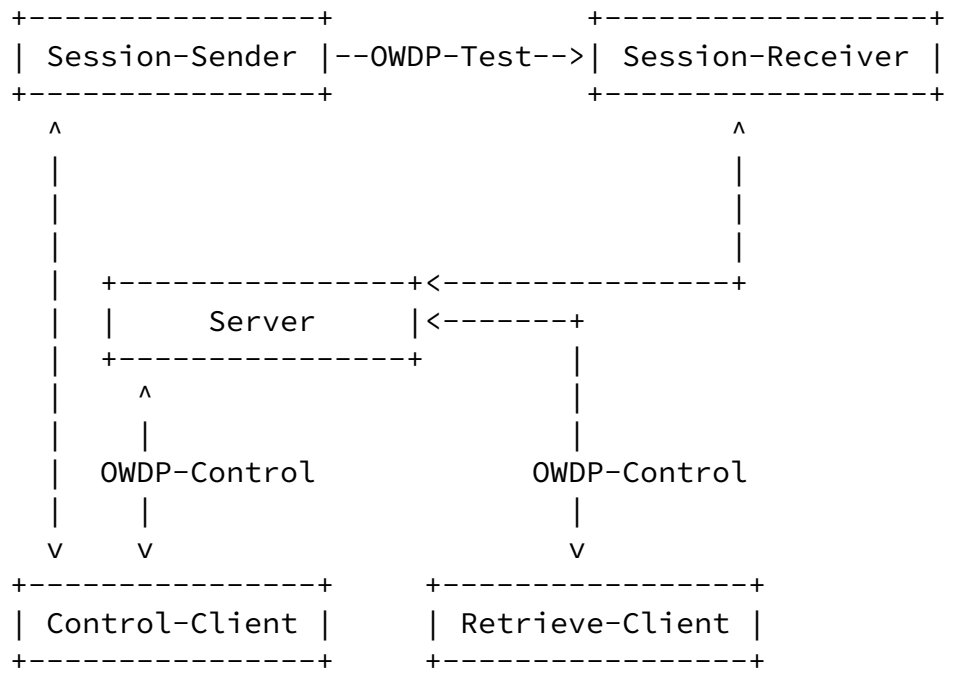
Server an end system that manages one or more OWDP-Test sessions, is capable of configuring per-session state in session endpoints, and is capable of

returning the results of a test session;

Control-Client an end system that initiates requests for OWDP-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 OWDP-Test sessions;

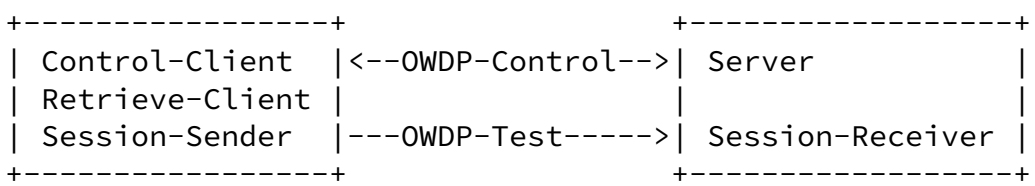
One possible scenario of relationships between these roles is shown below.



(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.



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, OWDP 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

OWDP consists of two inter-related protocols: OWDP-Control and OWDP-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 OWDP-Test sessions. IANA will be requested to allocate a well-known port number for OWDP-Control sessions. An OWDP server SHOULD listen to this well-known port.

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

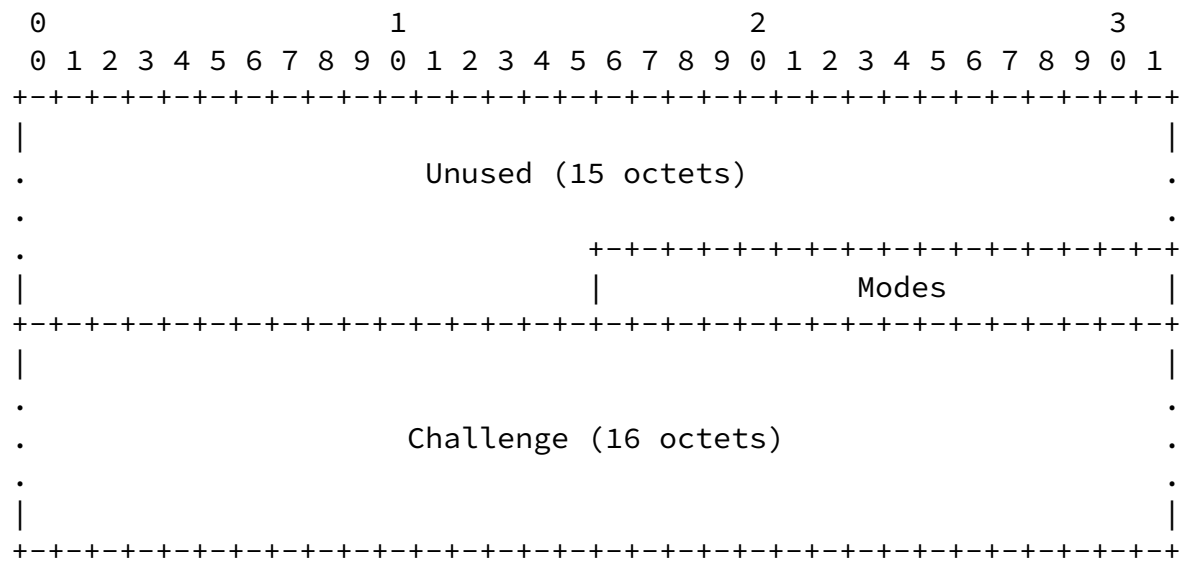
The OWDP-Control and OWDP-Test protocols support three modes of operation: unauthenticated, authenticated, and encrypted. The authenticated or encrypted modes require endpoints to possess a shared secret.

4. OWDP-Control

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:

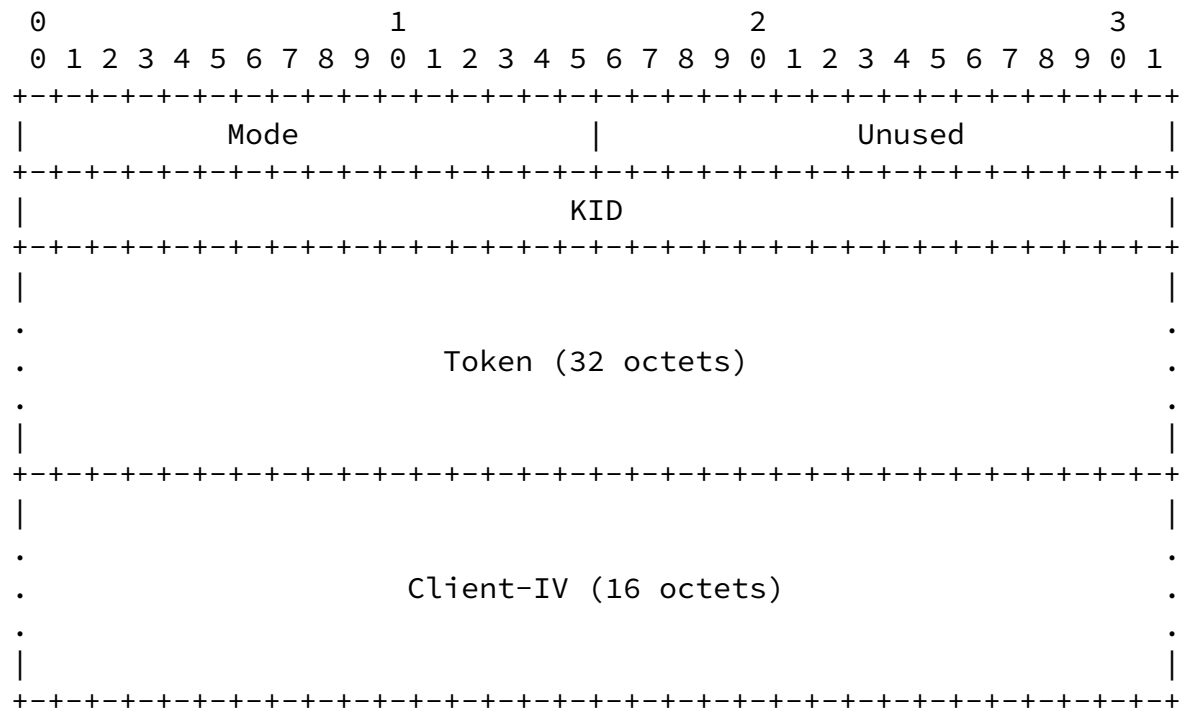


The following mode values are meaningful: 1 for unauthenticated, 2 for authenticated, 4 for encrypted. The value of the Modes field sent by the server is the bit-wise OR of the mode values that it is willing to support during this session.

If the Modes octet 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:



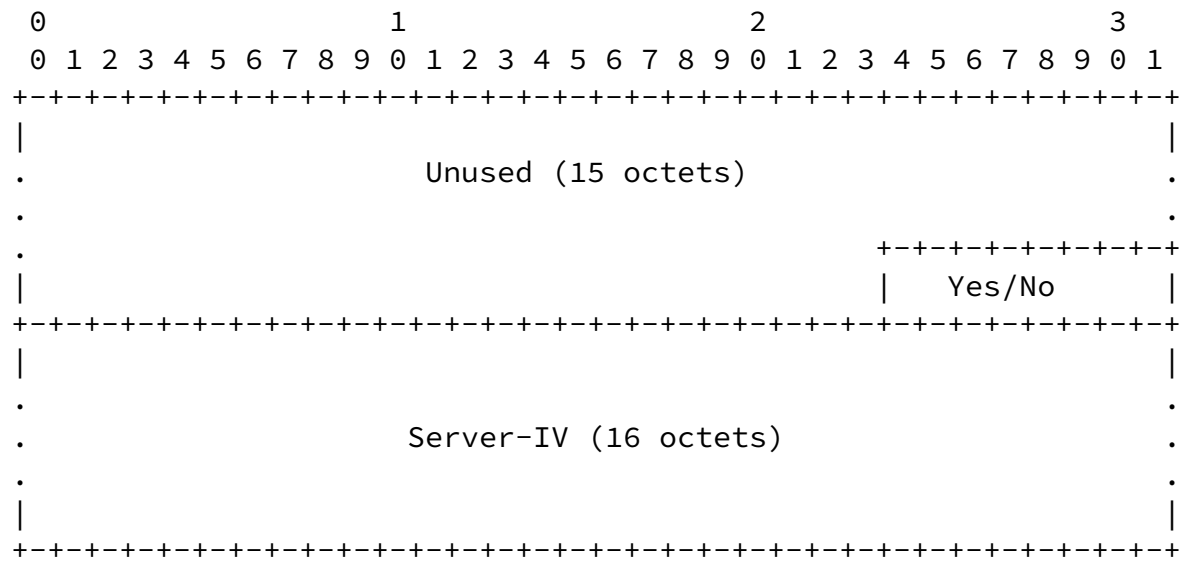
Here Mode is the mode that the client chooses to use during this OWDP-Control session. It will also be used for all OWDP-Test sessions started under control of this OWDP-Control session.

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

Otherwise, KID (key ID) is a 4-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 KID.

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

The server MUST respond with the following message:



A zero value in the "Yes/No" field means that the server accepts the authentication and is willing to conduct further transactions. Any non-zero value 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 OWDP-Control session. If a "No" response is sent, the server MAY close the connection after this message. The client SHOULD close the connection if it gets message that says "No" at this stage.

The previous transactions constitute connection setup.

4.2. OWDP-Control Commands

In authenticated or encrypted mode (which are identical as far as OWDP-Control is concerned, and only differ in OWDP-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.

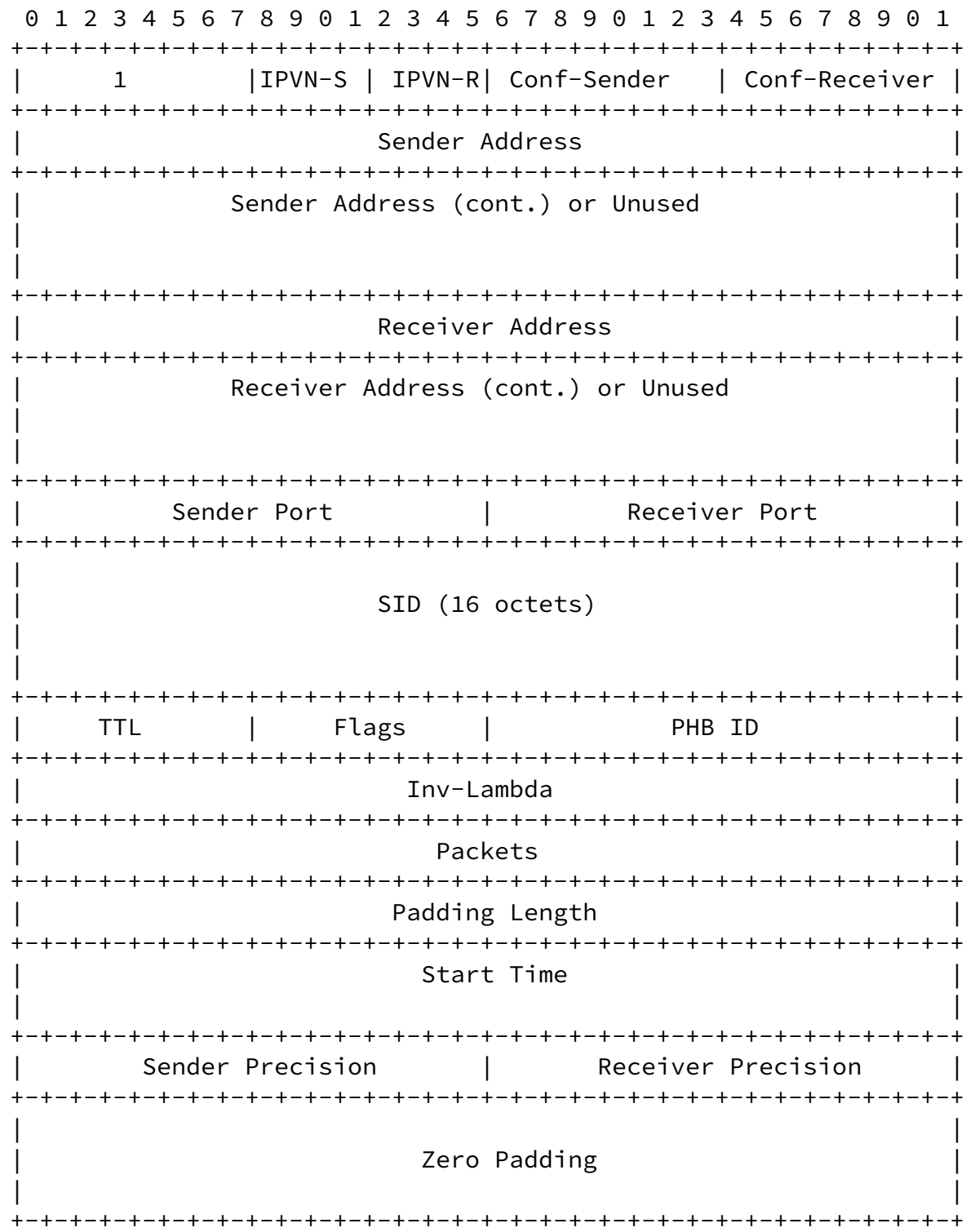
While conducting active measurements, the only command available is Stop-Session.

These commands are described in detail below.

4.3. Creating Test Sessions

Individual one-way delay measurement sessions are established using a simple request/response protocol. An OWDP client MAY issue zero or more Request-Session messages to an OWDP 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:



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.

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 OWDP test session is requested.

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.

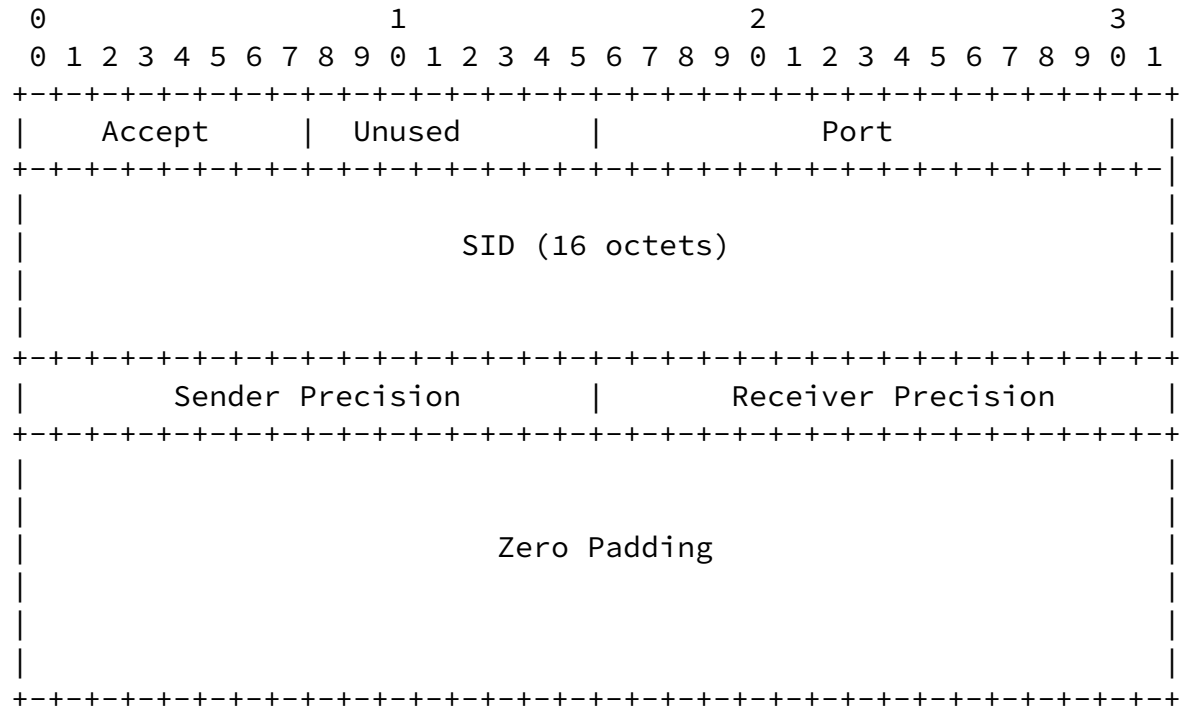
The value Packets is the number of active measurement packets to be sent during this OWDP-Test session (note that both server and client can abort the session early).

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

Start Time is the time when the session is to be started (but not before Start-Sessions command is issued).

Sender Precision and Receiver Precision are signed integers in the range +32 to -32 indicating the precision of the corresponding clocks, in seconds to the nearest power of two, as described in RFC 958. Sender Precision is meaningful only if Conf-Sender is not set. Receiver Precision is meaningful only if Conf-Receiver is not set.

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



Zero in the Accept field means that the server is willing to conduct the session. Any non-zero value indicates rejection of the request.

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

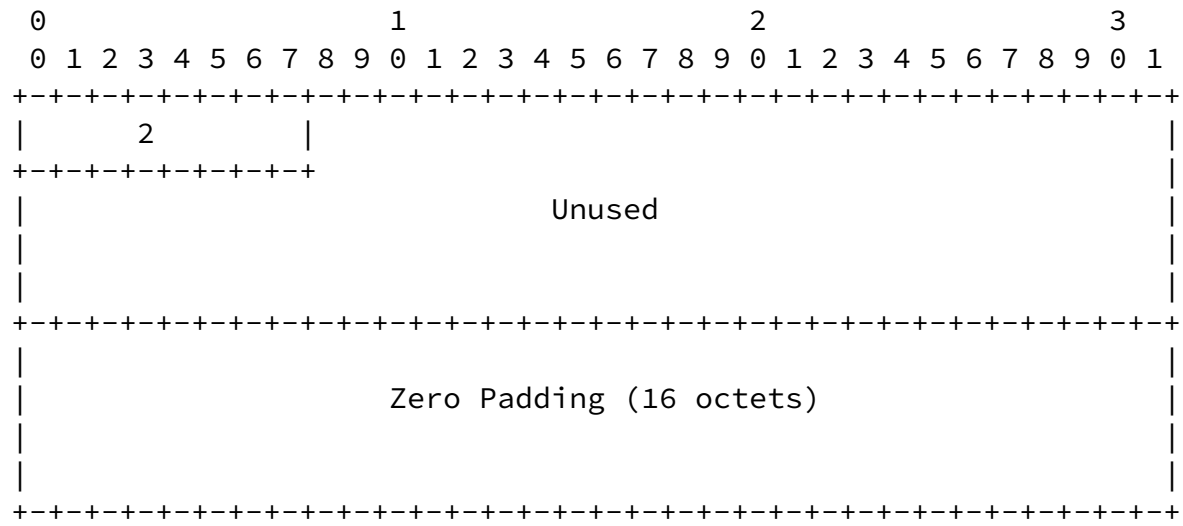
random value.

Sender Precision and Receiver Precision have the same meaning as in Request-Session command. Sender Precision is meaningful only if Conf-Sender is set. Receiver Precision is meaningful only if Conf-Receiver is set.

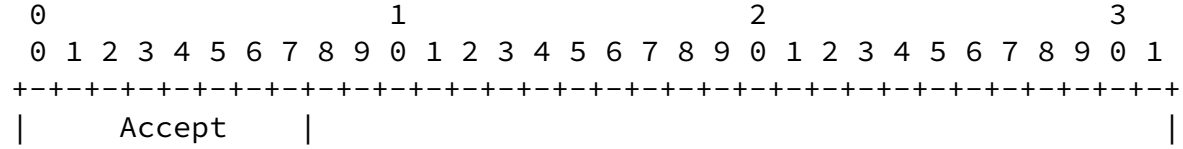
4.4. Starting Test Sessions

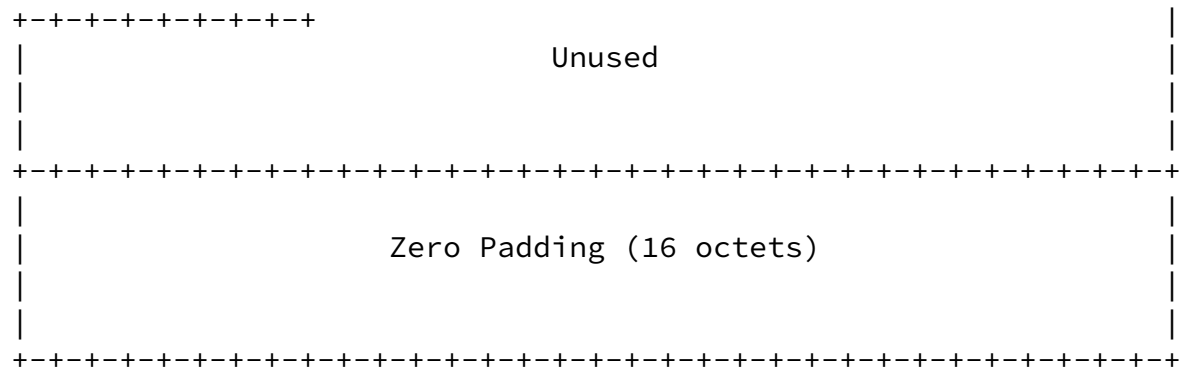
Having requested one or more test sessions and received affirmative Accept-Session responses, an OWDP client may start the execution of the requested test sessions by sending a Start-Sessions message to the server.

The format of this message is as follows:



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



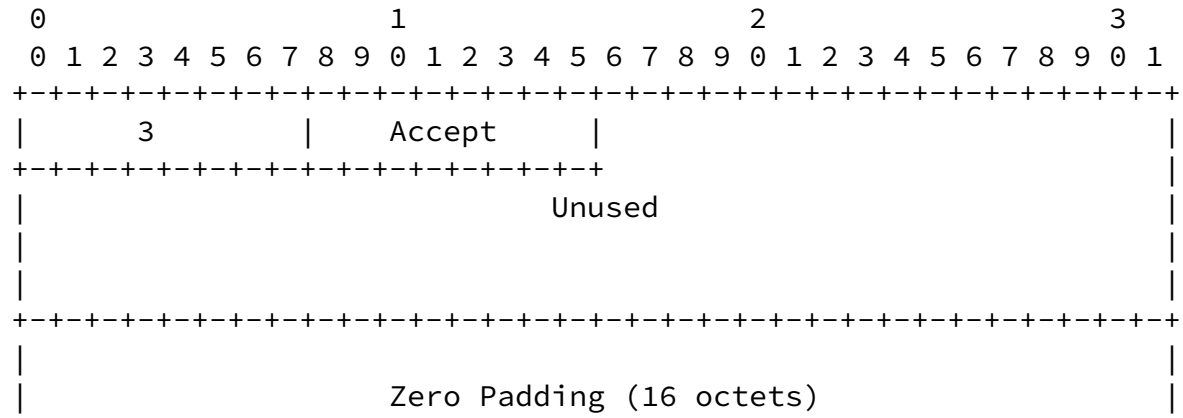


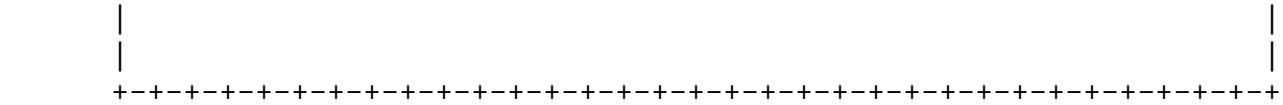
If Accept has any non-zero value, the Start-Sessions request was rejected; zero means that the command was accepted. The server MAY and the client SHOULD close the connection in the case of a negative response.

The server SHOULD start all OWDP-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 OWDP-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:





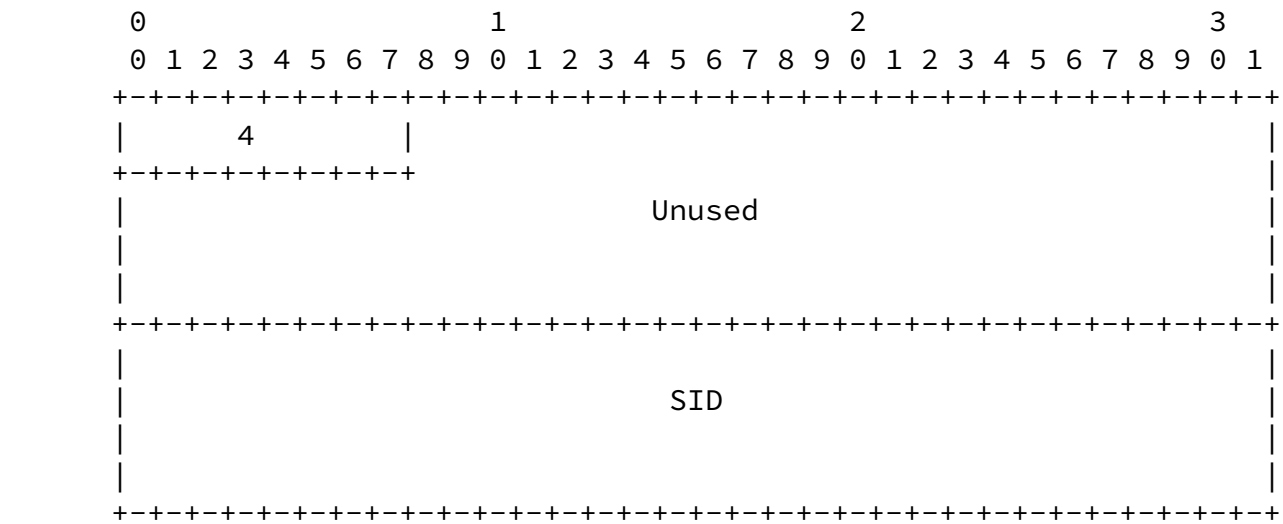
Normally, the client SHOULD send this command after the OWDP-Test streams have completed. However, either client or server MAY send it prematurely.

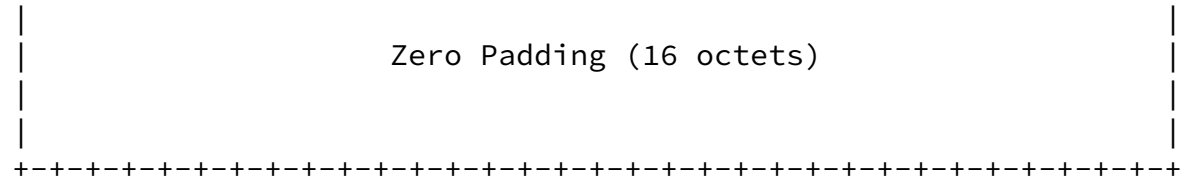
Non-zero value of Accept indicates a failure of some sort. Zero values indicates normal (but possibly premature) completion. If Accept had non-zero value (from either party), or if it was not transmitted at all (for whatever reason, including TCP connection used for OWDP-Control breaking), results of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

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

4.6. Retrieve-Session

The format of this client command is as follows:





The server MUST respond with a Control-Ack message. Again, any non-zero value in the Accept field means rejection of command. Zero means that data will follow.

If Yes/No was 0, the server then MUST send the OWDP-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 2 octets of Sender Precision, 2 octets of Receiver precision, 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. A zero padding consisting of 16 octets is then appended.

5. OWDP-Test

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

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

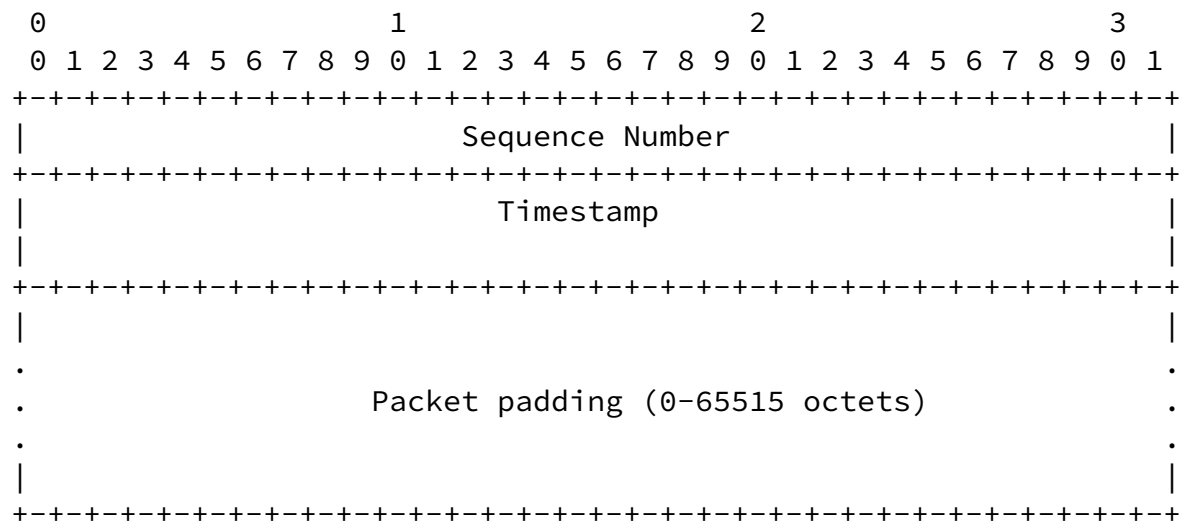
OWDP-Control client, OWDP-Control server, OWDP-Test sender, and OWDP-

Test receiver can potentially all be different machines. (In a typical case we expect that there will be only two machines.)

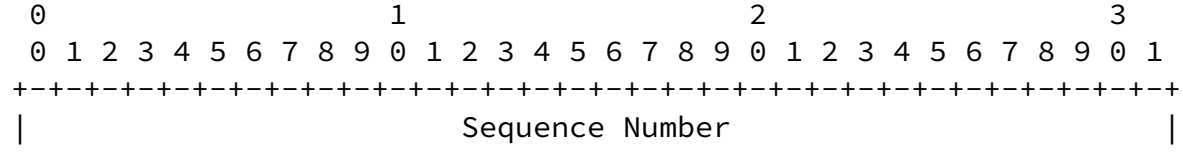
5.1. Sender Behavior

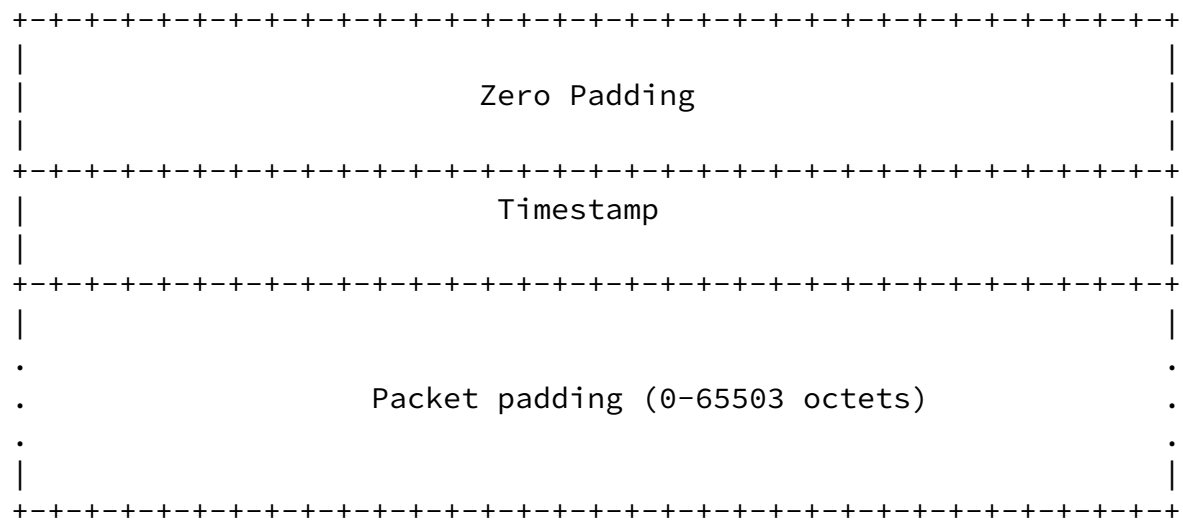
The sender sends the receiver a stream of packets with Poisson distribution of times between packets. The format of the body of a UDP packet in the stream depends on the mode being used.

For unauthenticated mode:

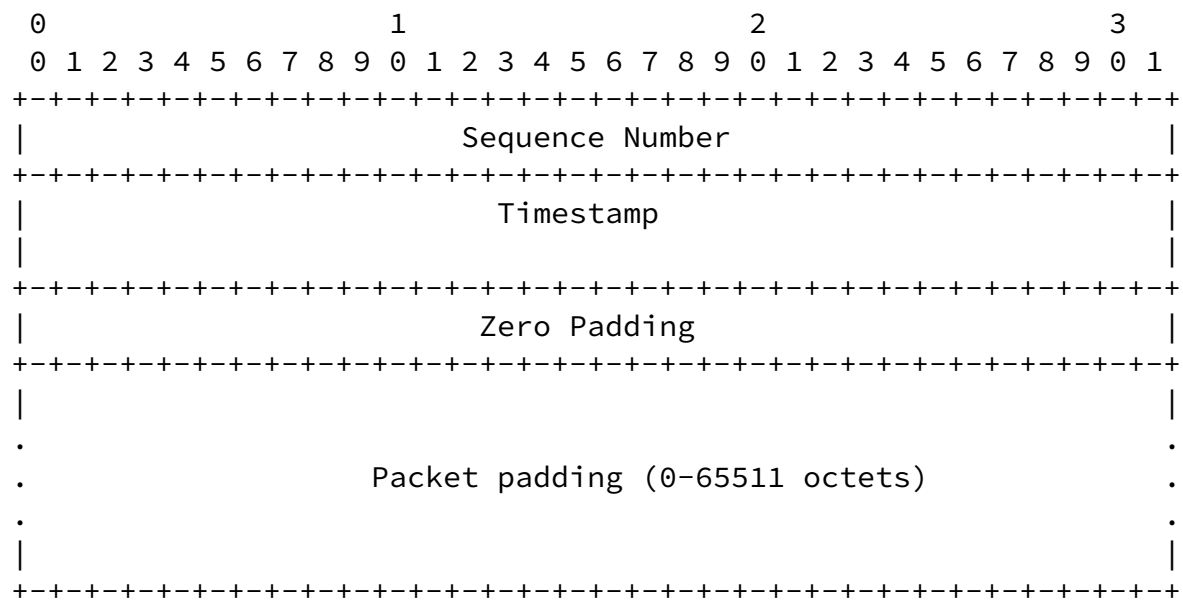


For authenticated mode:



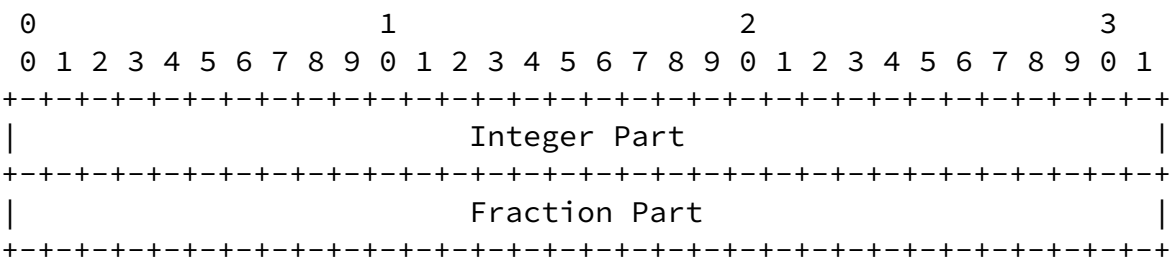


For encrypted mode:



The format of timestamp is the same as that of NTP v3 protocol [RFC958]. Quoting from RFC 958:

> NTP timestamps are represented as a 64-bit fixed-point number, in seconds relative to 0000 UT on 1 January 1900. The integer part is in the first 32 bits and the fraction part in the last 32 bits, as shown in the following diagram.



This format allows convenient multiple-precision arithmetic and conversion to Time Protocol representation (seconds), but does complicate the conversion to ICMP Timestamp message representation (milliseconds). The low-order fraction bit increments at about 0.2-nanosecond intervals, so a free-running one-millisecond clock will be in error only a small fraction of one part per million, or less than a second per year.

Sequence numbers start with 0.

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 OWDP-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 password-protect service provided by a particular OWDP-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 OWDP-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 OWDP-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.

OWDP-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 OWDP-Control client.

OWDP-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/

[RFC958]D. Mills, "Network Time Protocol (NTP)", RFC 958, September 1985.

[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.

[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 Test-Traffic Home page, 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/ripencc/mem-services/ttm/Talks/9805_nluug.ps.gz. (NOTE: it's actually postscript, not gzip'd postscript.)

[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: June 2001