Network Working Group
Internet Draft
Expiration Date: May 2001

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

November 2000

# A One-way Delay Measurement Protocol
<draft-ietf-ippm-owdp-00.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 to date no standard that would permit initiation of test streams or exchange of packets to collect

singleton metrics in an interoperable manner.

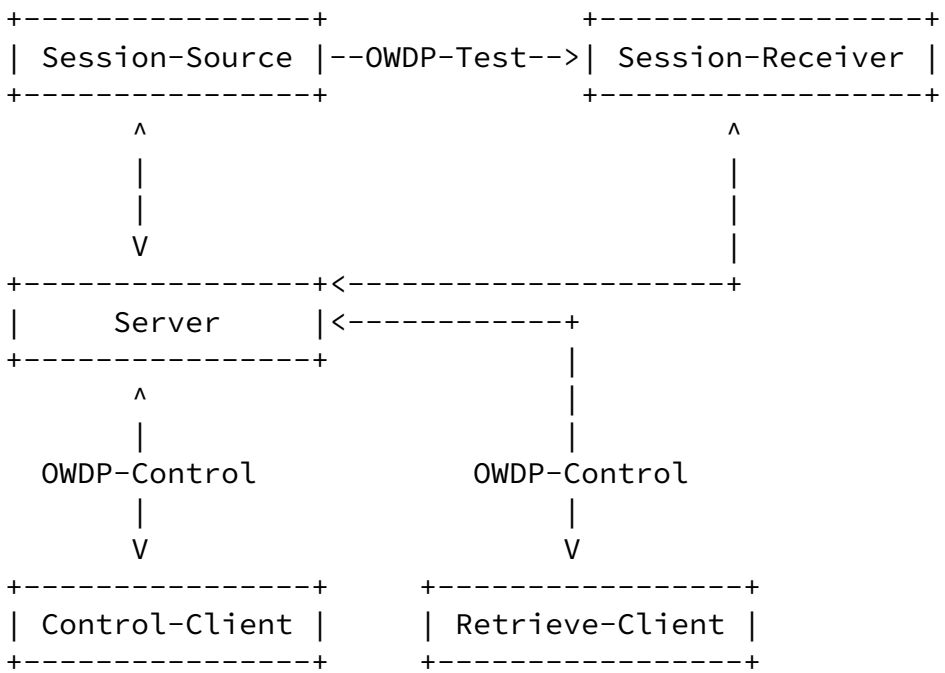
With the increasingly wide availability of affordable global positioning system (GPS) and CDMA based time sources, hosts increasingly have available to them very accurate time sources--either directly or through their proximity to NTP primary (stratum 1) time servers. By standardizing a technique for 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 with several roles logically separated to allow for broad flexibility in use. Specifically, the following roles are logically separate: Control-Client, Retrieve-Client, Server, Session-Source, and Session-Receiver. The relationships between these are shown below.




A Control-Client speaks to a Server and may request test session initiation and may request that accepted test sessions be started and stopped. A Retrieve-Client also speaks to a Server and may request the results of an OWDP test session. The test session itself consists of a stream of singleton OWDP-Test packets sent from Session-Source to Session-Receiver.

Any combination these logical blocks may, in fact, be collocated.

[FIXME: Insert interesting examples.]

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.

## 3. Protocol Overview

OWDP actually 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. OWDP server SHOULD listen to this well-known port.

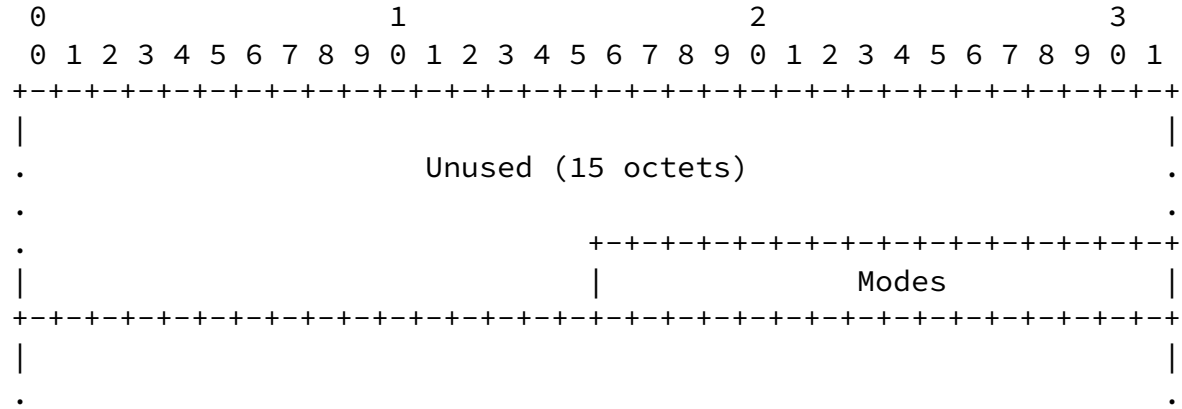
OWDP-Control messages are transmitted only before OWDP-Test sessions actually started and after they complete (with the possible exception of an early Stop-Sessions message).

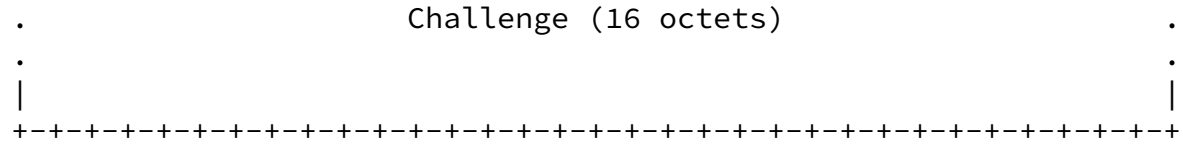
The protocol allows negotiating three modes of operation of OWDP-Control and OWDP-Test: unauthenticated, authenticated, and encrypted. If authenticated or encrypted mode is desired, endpoints must possess a shared secret.

### 3.1. OWDP-Control

The client opens a TCP connection to the server on a well-known port.

The server responds with server greeting:





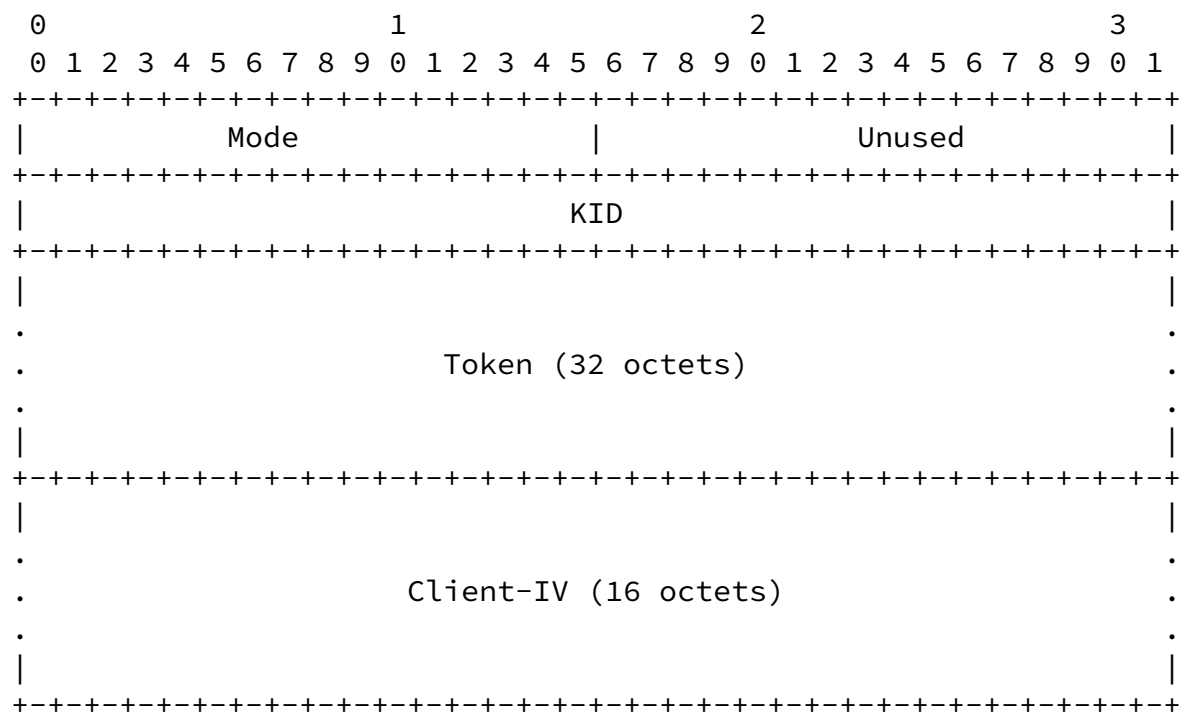

The following mode values are meaningful: 1 for unauthenticated, 2 for authenticated, 4 for encrypted. The value of the Modes field sent by the server is the bit-wise OR of the mode values it is willing to support during this session. If Modes is 1, the Challenge field MAY be committed.

If Modes octet is zero (server doesn't wish to communicate with this client), the server MAY close the connection after this message. 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.

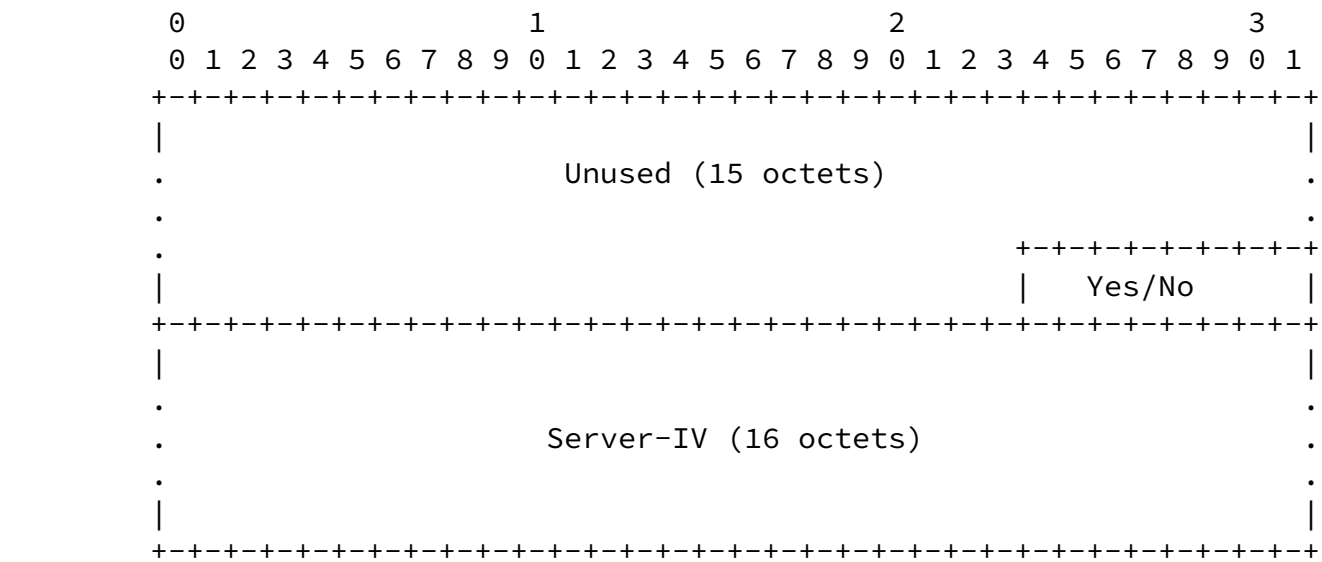
Otherwise, KID (key ID) is a 4-octet indicator of which shared secret the client wishes to use to authenticate or encrypt and Token is the concatenation of a 16-octet challenge and a 16-octet Session-key, encrypted using the AES (Advanced Encryption Standard) [AES] in Cipher Block Chaining (CBC). Encryption 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:







Here "Yes/No" is either 1 or 0. Yes (0) means that the server accepts the authentication and is willing to conduct further transactions. No (any non-zero value) means that the server doesn't accept 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.

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, End-Sessions, Retrieve-Session. The command End-Sessions is available to both client and server.

[FIXME: move next two paragraphs below?] After Start-Sessions is sent/received by the client/server, and before it both sends and receives End-Sessions (order unspecified), it is said to be conducting active measurements.

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





## 3.2. Creating Test Sessions

Individual one-way delay measurement sessions are established using a simple request/response protocol. An OWDP client, may issue one 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.





Represents can have three values: Source (0), Dest (1), and Third-

Party(2). It tells the server on whose behalf the client is speaking.

The meaning of Port depends on the value of Represents. If it is Source, Port is the port to expect OWDP-Test packets from. Is it is Dest, Port is the port to send OWDP-Test packets to. Port is unused in the case of a Third-Party client.

The Source Address and Dest Address fields contain respectively the source and destination addresses of the end points of the Internet path over which an OWDP test session is requested. The IPVN field contains the IP version number of the source and destination addresses that follow. In the case of IPVN=4, twelve unused octets follow each address.

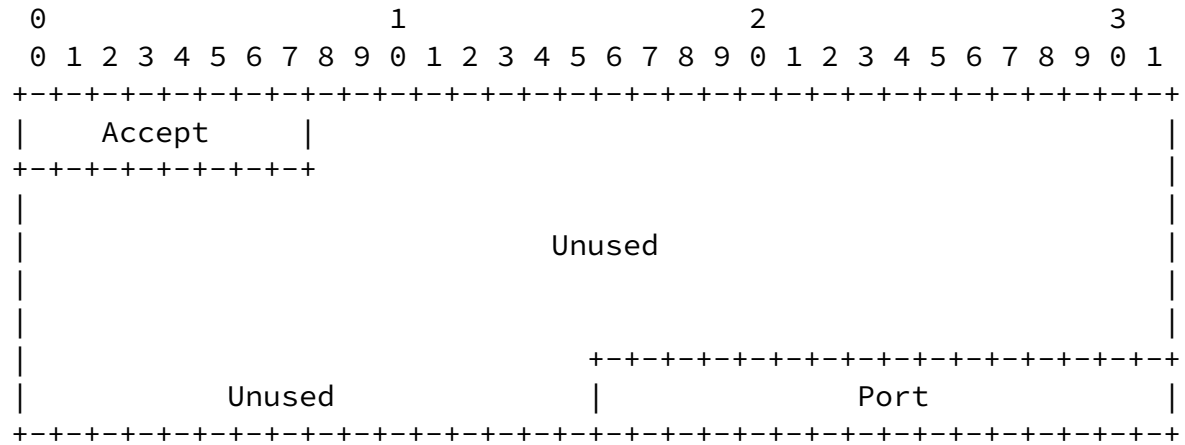
SID is the session identifier. It can be used in later sessions as an argument for Retrieve-Session command. It is meaningful only if Represents is Dest.

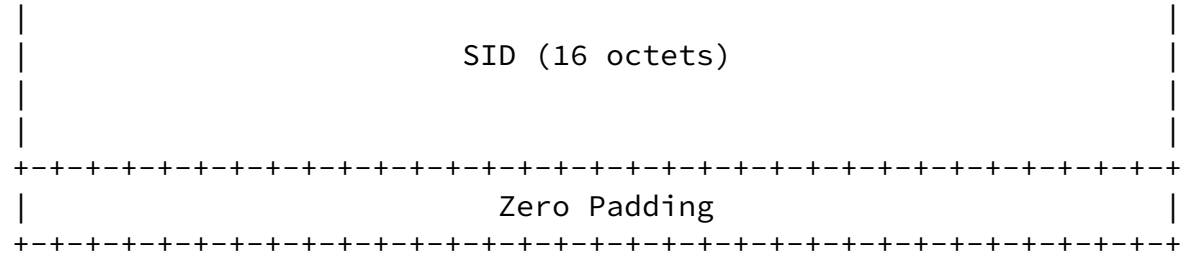
The field Inv-Lambda is an unsigned integer and is the scaled reciprocal in microseconds of rate at which the Poisson test stream is to be generated. This allows the average Poisson sampling interval for the requested test session to be set to between 1 microsecond and over an hour.

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).

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 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.

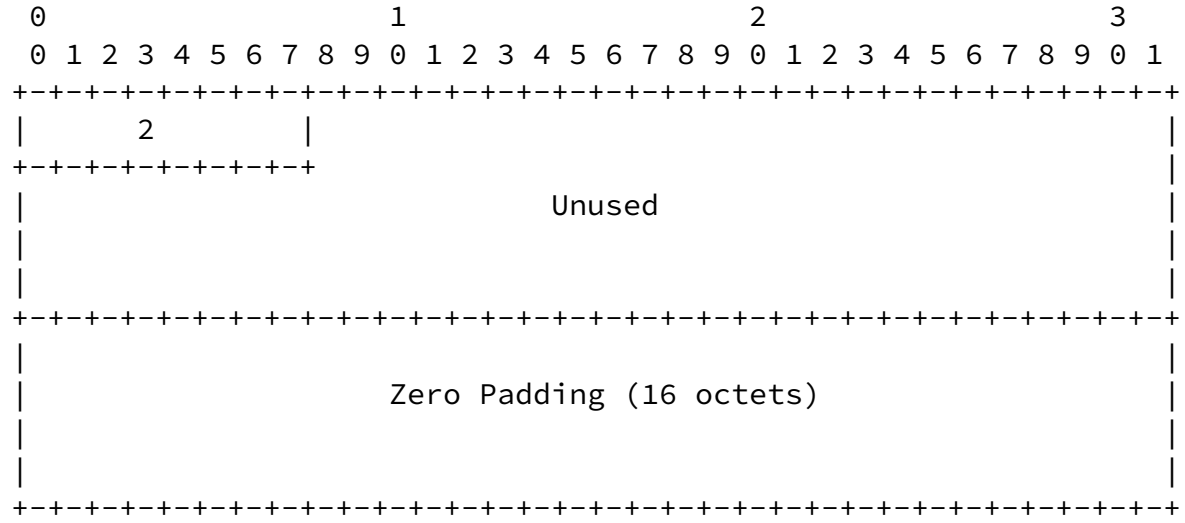
The meaning of Port depend on the value of Represents in the query that solicited the response. If it was Dest, Port is the port to expect OWDP-Test packets from. Is it was Source, Port is the port to send OWDP-Test packets to. If is was Third-Party, the Port field is unused.

SID is a locally-unique server-generated session identifier. It can be used later as handle to retrieve the results of a session. An OWDP server MUST return an SID, if Represents was Source or Third-Party. It is not meaningful if Represents was Dest.

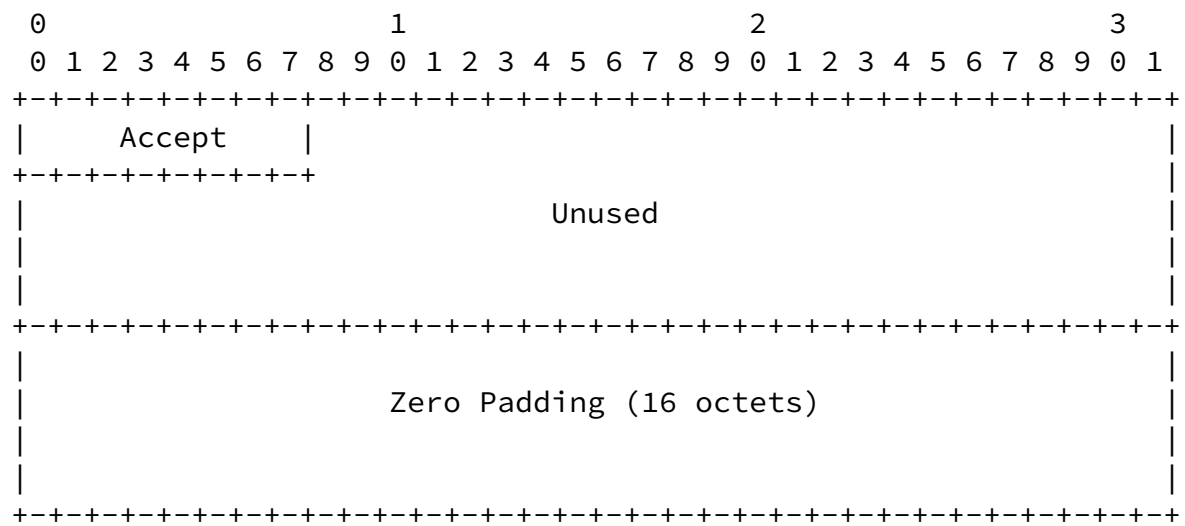
## 3.3. Starting Test Sessions

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.

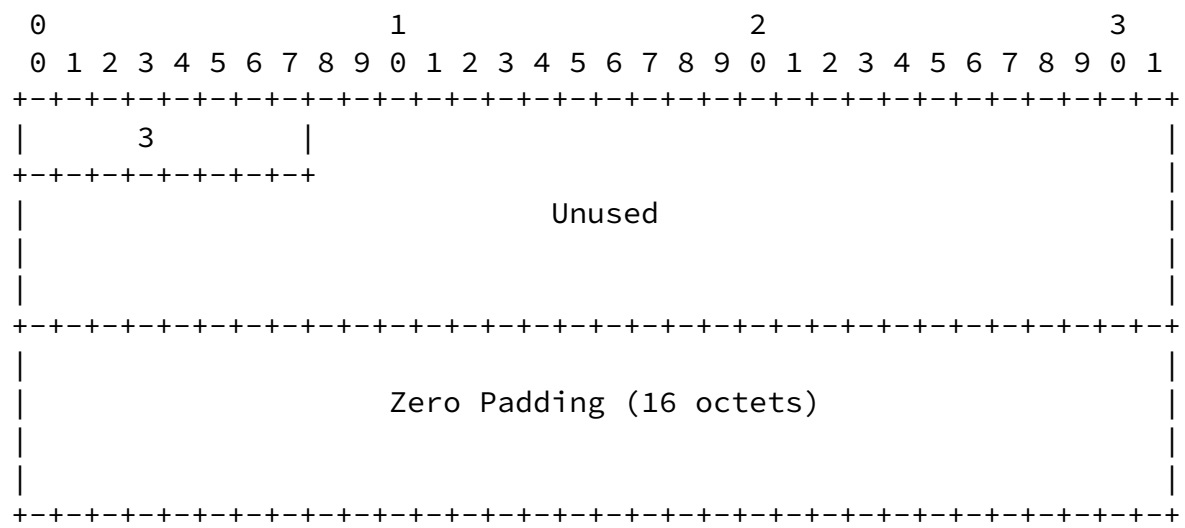
The server SHOULD start all OWDP-Test streams immediately after it sends the response or immediately after their specified start times, whichever is later. (Note that a client can effect an immediate start by specifying in Request-Session a Start Time in the past.) The client represents a Source, the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





## 3.4. 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:

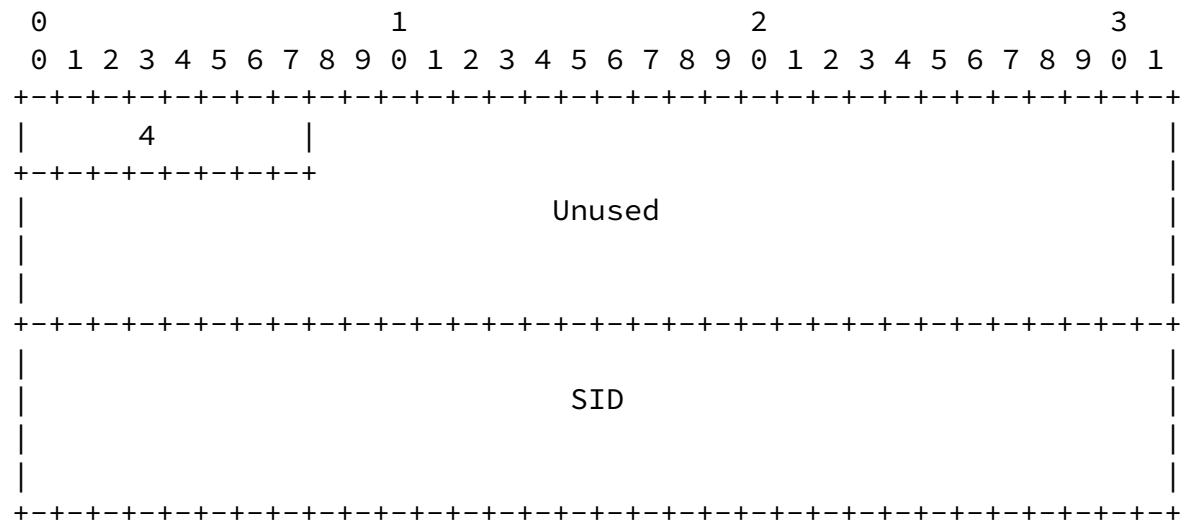



Normally, the client SHOULD send this command after the OWDP-Test streams have completed. However, either client or server MAY send it prematurely.

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

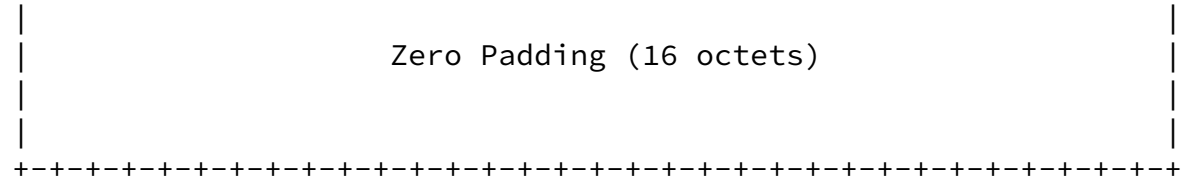
## 3.5. Retrieve-Session

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.

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.

## 4. OWDP-Test

This section describes OWDP-Test protocol. It runs over UDP using

source and destination 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.)

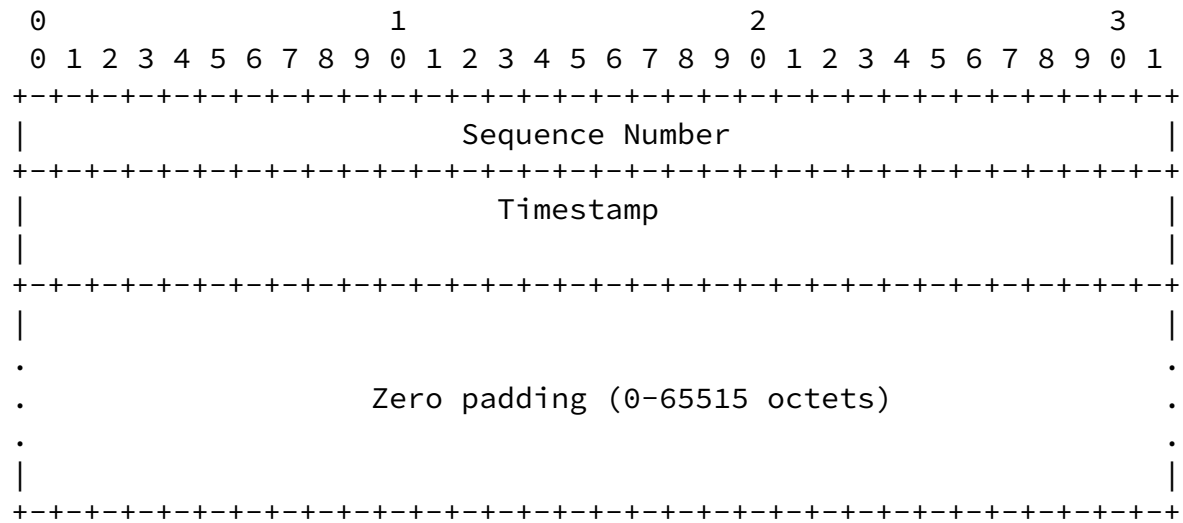
## 4.1. Sender Behavior

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.

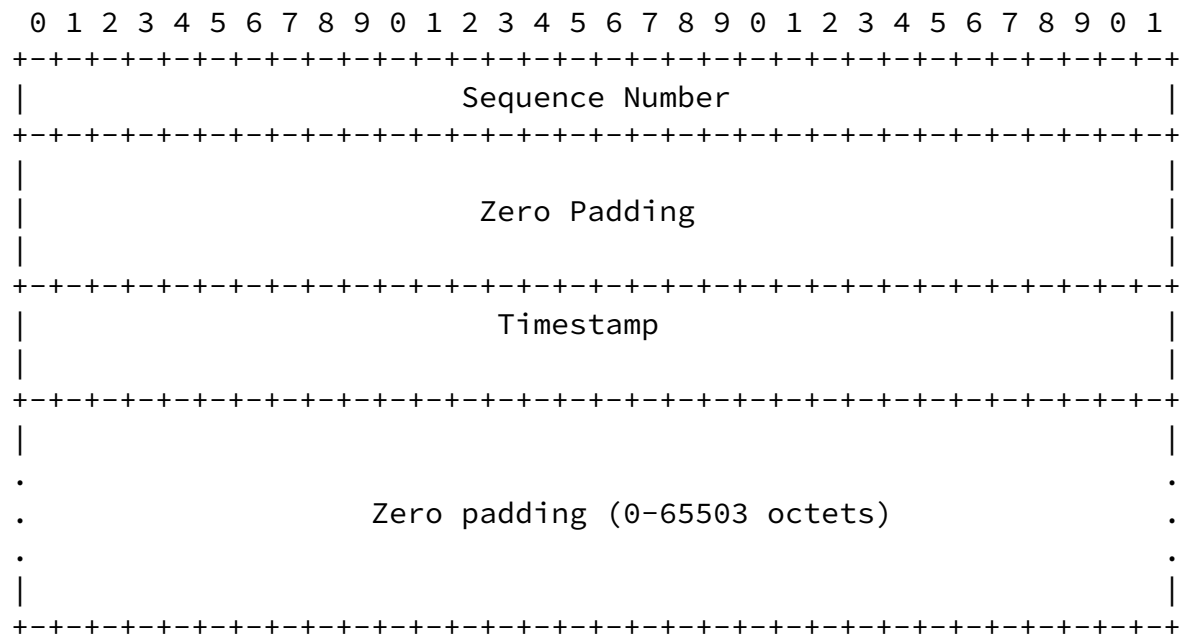
For unauthenticated mode:



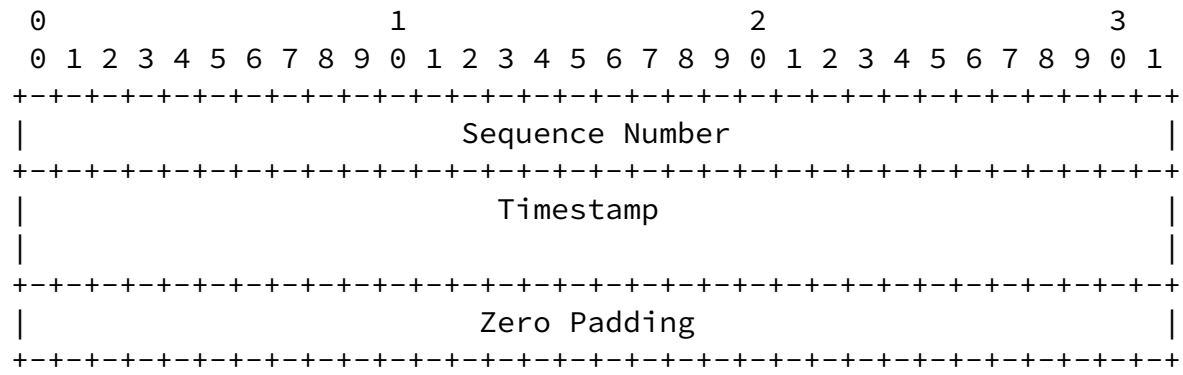





For authenticated mode:

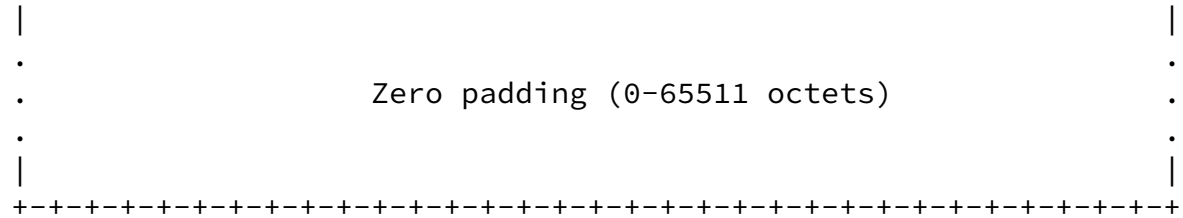



For encrypted mode:





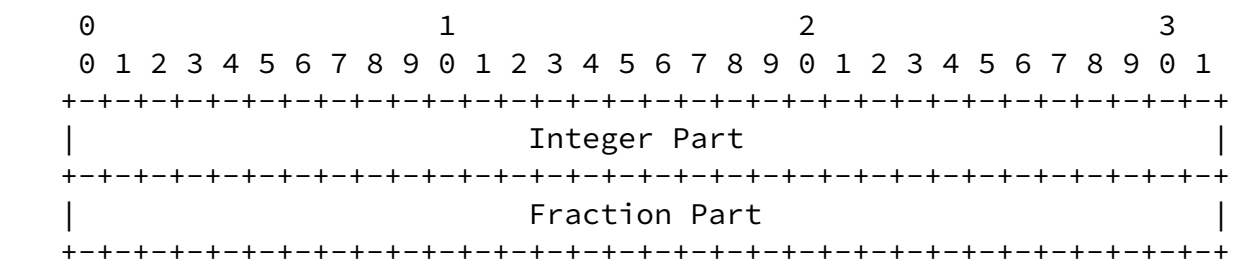



The format of timestamp is the same as that of NTP v3 protocol [RFC958]. Quoting from RFC 958:

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 (Poisson) distribution. As suggested in RFC 2330, the ith sampling interval Ei may be computed using inverse transform:

```
Ei = -log(Ui) / lambda
```





where Ui is uniformly distributed between 0 and 1 and 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.) and lambda is the desired mean rate of the sampling distribution.

[FIXME: should state precisely how the 16 byte block is interpreted as a number between 0 and 1].

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!]

## 4.2. Receiver Behavior

FIXME: Expand this sketch.

As packets are received,

+ Timestamp the received packet.

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

+ Packets not received within parameter Tl, the loss threshold are considered lost. FIXME: loss threshold not mentioned above. also need to decide if the receiver knows which packets are lost, and if so how is it represented in the results perhaps (seqno presumed send time, receive time of 0).

# 5. Security Considerations

The goal of authenticated mode to let one be able to 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.

FIXME: More stuff to go here.

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.

## 6. 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.

[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

## 7. 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: kalidindi@advanced.org

Expiration date: May 2001