

TICTOC
Internet-Draft
Updates: none (if approved)
Intended status: Standards Track
Expires: December 30, 2018

J. Alvarez-Hamelin, Ed.
D. Samaniego
A. Ortega
Universidad de Buenos Aires
R. Geib
Deutsche Telekom
June 28, 2018

Synchronizing Internet Clock frequency protocol (sic)
draft-alvarez-hamelin-tictoc-sic-01

Abstract

Synchronizing Internet Clock Frequency specifies a new secure method to synchronize difference clocks on the Internet, assuring smoothness (i.e., frequency stability) and robustness to man-in-the-middle attacks. In 90% of all cases, Synchronized Internet Clock Frequency is highly accurate, with a Maximum Time Interval Error less than 25 microseconds by a minute. Synchronized Internet Clock Frequency is based on a regular packet exchange and works with commodity terminal hardware.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 30, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. sic frequency protocol overview	3
3. The formal definition of sic frequency protocol	8
3.1. Algorithm description	8
3.2. Protocol definitions	13
3.3. Protocol packet specification	15
3.4. Minimum sic deployment	16
4. Implementation of sic frequency protocol	17
4.1. Evaluation	17
5. Conclusions	19
6. Security Considerations	19
7. IANA Considerations	20
8. Acknowledgements	20
9. References	20
9.1. Normative References	20
9.2. Informative References	20
Appendix A. Example of RTT to NTP servers	21
Authors' Addresses	24

1. Introduction

There are different types of clock synchronization on the Internet. NTP [RFC5905] remains one of the most popular because a potential user does not need any extra hardware, and it is practically a standard in most of the operating systems distributions. Its working principle relies on time servers having some kind of precise clock source, like atomic clocks or GPS based. For most of the needs, NTP provides an accurate synchronization. Moreover, NTP recently incorporates some strategies oriented to avoid man-in-the-middle (MitM) attacks. NTPs potential accuracy is in the order of tens of milliseconds.

Synchronizing Internet Clock frequency (sic frequency) is a protocol providing synchronized difference clocks in two endpoints connected to the Internet. While synchronized absolute clocks aim on a measurement of exact time differences between them, synchronized difference clocks allow measurements during identical time intervals at two locations. This is useful if loads, packet loss or a variation in delay is to be measured.

The sic frequency design is close to TSClocks (see below) but it takes advantage of statistics to perform better. sic frequency synchronization relies on Internet based delay measurements. Route changes are frequent, so we include its detection. Finally, our implementation also contemplates the protection to MitM attacks, including the signature of measurements in each packet. sic frequency does neither put constraints on the quality of a server's clock, nor does it require a limitation of the distance of synchronised end systems.

Another proposal is the TSClocks [ToN2008], which take advantage of the internal computers' clock. This work has been shown a very interesting solution because it is not expensive and can be used in any computer connected to the Internet. This solution was proposed in the beginning at LAN (Local Area Network) level, and then it has been extended to other situations. In [ToN2008] authors report a difference clock error of about half of hundred of microseconds for a WAN connection with 40ms of RTT (Round Trip Time).

When accuracy and stability are needed, further options arise, e.g., the PTP clock [RFC8173] (this mechanism was also defined as the IEEE Std. 1588-2008). The PTP clock however incorporates specialised hardware to provide a highly accurate clock, which is required in each point to be synchronised. Also the GPS (Global Position System) requires specialized hardware in every point of measurement. While GPS may be less expensive than PTP, the GPS unit requires a sky clear view for working. The latter may be costly or impossible in some locations.

Finally, we mention the [ITU-G.8260] shows a methodology to measure delays in networks. It is based on filtering that selects some packets to perform the delay computation. The packet selection is based on the minimum and average RTT, and we show that both of them have some statistical problems to determine (see Section 2).

2. sic frequency protocol overview

Synchronizing Internet Clock frequency (sic frequency) is a protocol providing synchronized difference clocks in two endpoints connected to the Internet. Synchronized difference clocks allow measurements

during identical time intervals at two locations. This is useful if loads, packet loss or a variation in delay is to be measured. The model of typical Internet time-measurement is shown in Figure 1.

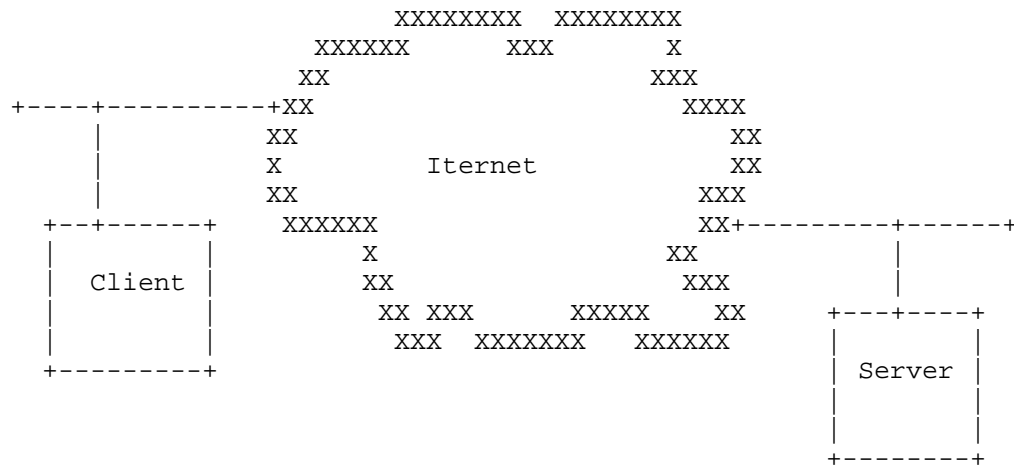


Figure 1: The clock synchronization of sic.--

In this model, sic frequency performs measurements with packets in the way shown in Figure 2.

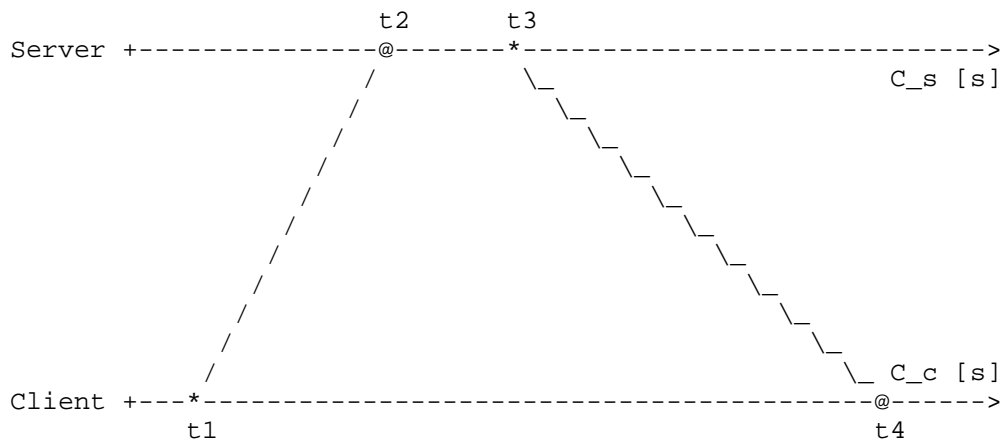


Figure 2: Time line of packets.--

Here, C_s is the server clock, C_c is the client clock and $t_1...t_4$ are timestamps.

Figure 2 shows a horizontal time line for client and server. The diagonal lines depict a packet traversing some physical space (wires, routers, and switches). The packet travel times are not assumed to be identical, because routes and background load may differ in each direction.

The difference between the client clock C_c and the server clock C_s can be modeled as:

$$\begin{aligned} C_c &= C_s + \phi \\ \phi(t) &= C_c(t) - C_s(t) \end{aligned} \quad (1)$$

where ϕ is the absolute clock difference. If RTT is constant (i.e. little or no background load) and routes are symmetric in both directions, the difference between clocks can be computed as:

$$\phi[c \rightarrow s] = t_1 - (t_2 - \text{RTT}/2) \quad (2)$$

$$\phi[c \leftarrow s] = t_4 - (t_3 + \text{RTT}/2) \quad (3)$$

and $\phi[c \rightarrow s] = \phi[c \leftarrow s]$. The general equation for the RTT is:

$$\text{RTT} = (t_2 - t_1) + (t_4 - t_3) \quad (4)$$

Computing Equations 2 and 3 for the this simplified case allows calculation of ϕ as a function of RTT. Note that if routes are not symmetrical it is impossible to determine the absolute clocks' difference.

The sic frequency protocol is based on statistics, background traffic- and network behavior observation behavior. The RTT between two endpoints follows a heavy-tailed distribution. An alpha-stable distribution shows as one possible model [traffic-stable]. This distribution can be characterized by four parameters: the localization "delta," the stretching "gamma," the tail "alpha," and the symmetry "beta," [alfa-estables]. The location parameter is highly related to the mode of the distribution: $\delta > 0$. The stretching is related to the dispersion: $\gamma > 0$. The symmetry, $-1 \leq \beta \leq 1$, indicates if the distribution is skewed to the right (the tail decays to the left) for positive values or the opposite direction for negatives ones. Finally, the tail alpha, defined in $(0,2]$, indicates if the distribution is Gaussian one when $\alpha=2$, a power law without variance for $\alpha < 2$, and also without statistic mean for $\alpha < 1$. The alpha-stable distribution is the generalization of the Central Limit Theorem for any distribution (i.e., it includes the cases without variance or mean).

Then, the $\phi(t)$ estimation involves the subtraction of two alpha-stable random variables, which yields on another alpha-stable distribution but symmetrical [alfa-estables]. Due to the characteristic of this result, i.e., a fixed mode and symmetry, a good estimator of the mode is the median.

Therefore, sic performs periodic measurements to infer the difference of two clocks in the Internet taking advantage of the empiric observations. The periodicity of RTT measurements is set to 1 second.

The parameters of the simple skew model [ToN2008] are estimated by the following equation:

$$\phi(t) = K + F * t, \quad (5)$$

where $\phi(t) = C_c - C_s$, K is a constant representing the absolute difference of time of client clock C_c and server clock C_s , and F is the rate parameter. As sic frequency is a difference clock, we only estimate the frequency parameter " F ."

Note that the " K " parameter cannot be estimated using just endpoints measurements. Estimating the " K " parameter accurately is out of scope, and we use $K = \min(RTT)/2$, as it used in several synchronization protocols under the assumption of symmetric paths. Considering the following asymmetry definition,

$$A = 1 - \frac{t[c \rightarrow s]}{t[c \leftarrow s]}, \quad (6)$$

where $t[c \rightarrow s]$ is the minimum delay measured from the client to the server. The maximum asymmetry A of equation 6 is $A=1$, which is unlucky, and this establishes the hard bound for the error of K as $\min(RTT)$: if $t[c \rightarrow s]$ approaches RTT , $t[c \rightarrow s]$ approaches zero. The difference between the two is $\phi(t)$, and this difference hence is close to $\min(RTT)$, if $A=1$. In our experiments the error in estimation $\phi(t)$ was always less than $\min(RTT)/2$.

Another problem with most of the synchronization protocols is the estimation of the minimum RTT, which depends upon the time-window within which the RTT is captured. A minimum RTT can only be measured in the absence of any cross traffic. In a first step, the minimum RTT measured during a window of 10 minutes ($mRTT_{10m}$) is captured. Based on these values, the minimum RTT over a week ($mRTT_w$) is determined. RTT_{ee} is defined as $mRTT_{10m} - mRTT_w$. Figure 3 shows the the RTT estimation error captured during an experiment where the minimum latency between probes was 9431 microseconds during one week,

i.e., $mRTT_w=9431$ microseconds. Notice that $mRTT_{10m}$ varies a lot, and the observed values can be more than 450 microseconds above the minimum RTT over a week. This error is a consequence of the statistical behavior of the RTT which can be modeled by the alfa-stable distribution.

Finally, it is mostly believed there always exist NTP servers at less than five hops with few milliseconds of RTT, because of the NTP deployment. In Appendix A we show a typical case in Latin America region where the RTT differ notably from host in the same city (Buenos Aires). This example reveals that in some countries could be not possible to have this desired situation and other synchronization tools are needed.

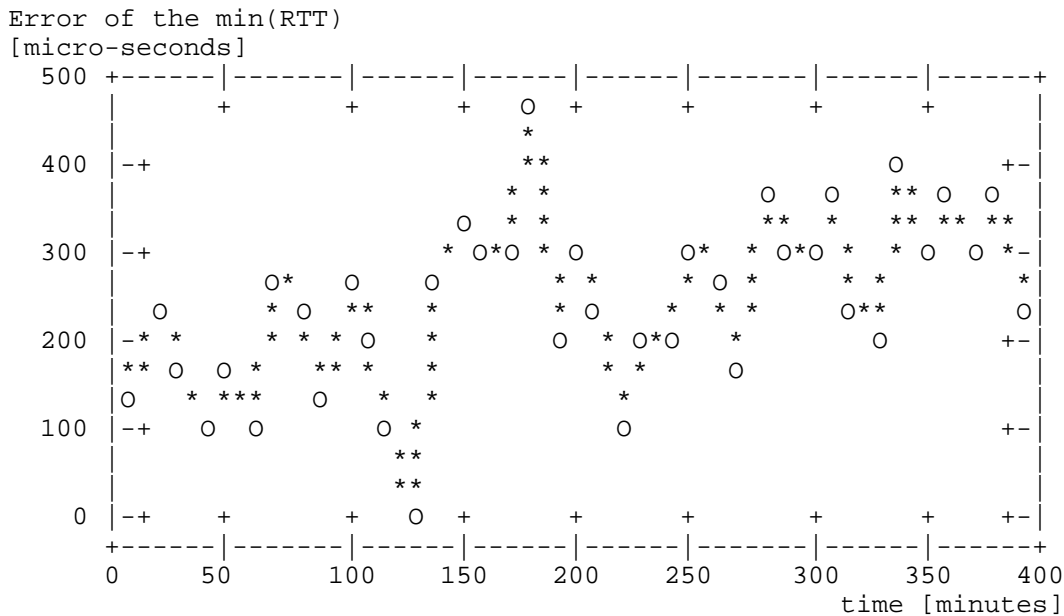


Figure 3: Min RTT error, estimated every 10 minutes along 7 hours.--

The sic frequency protocol estimates $\phi(t)$ of Equation 5 using measurement statistics and taking advantage of the inherent RTT properties, i.e., the heavy tail distribution and its alfa-stable distribution model. The basic sic frequency operation is to periodically send packets, estimate $\phi(t)$, and correct the local clock with:

$$t_c = t + \phi(t) , \quad (7)$$

where t_c is the corrected time and t the local clock time (notice that $\phi(t)$ is calculated according to Equation 1).

The sic protocol also detects route changes by seeking a nonnegligible difference between the minimum RTT of the actual and past round trip measurement. The next section also discusses different mechanisms to detect route changes by RTT evaluation.

3. The formal definition of sic frequency protocol

Section 3.1 presents the sic frequency algorithm. In addition, parameters and their definitions are introduced. Finally, formal packet formats are provided.

The sic frequency protocol MUST sign the packets with the deterministic Elliptic Curve Digital Signature Algorithm (ECDSA) specified by [RFC6979] to protect sic frequency from MitM attacks. To avoid delays when a packet is signed, sic frequency signs them in a deferred fashion. That is, in each packet carries the signature of the previous packet (see algorithms in Figure 6 and Figure 5).

3.1. Algorithm description

sic frequency implementations MUST support the formal description specified by this section. Once activated, the sic frequency protocol MUST operate permanently while a client and a receiver exchange measurement packets. sic frequency works with three states: NOSYNC, PRESYNC, and SYNC. These states are triggered by the variables `errsync`, `presync`, and `sync`.

Lines 1 to 4 of the pseudocode in Figure 4 initialize the required data structures needed and set the sic frequency state to NOSYNC. In NOSYNC state, a complete measurement window estimates ϕ 's by Equation 2 (see line 8). Notice that also Equation 3 can be used, or an average of both Equations. During the experiments, using a single equation only resulted in estimations with a smaller error. The possible explanation is that measurements are affected by the same type of traffic.

The median of the measurement window is also computed in line 9, while lines 10-12 are used to verify if there is a path change in the measurements. When an appreciable difference is detected (bounded by `errRTT`) in line 13, the "else" clause is executed and the systems re-initiates the cycle (see lines 17-22). Notice that line 13 verifies if the absolute value of the minimum RTTs is lower than a percentage of minimum over the complete RTT window.

The sic frequency algorithm specification is presented by three tables of pseudocode. The parameters are explained after the third table.

```

=====
|               sic frequency algorithm               |
=====
1  Wmedian <-0, Wm <-0, WRTT <-0, actual_m <-0, actual_c <-0
2  presync <- INT_MAX - P, epochsync <- INT_MAX - P, n_to <-0
3  synck <- false, errsync <- epoch, set(0, 0, NOSYNC), e_prev<-epoch
4  send_sic_packet(SERVER_IP, TIMEOUT)
5  for each timer(RUNNING_TIME) == 0
6      (epoch, t1, t2, t3, t4, to) <- send_sic_p(SERVER_IP, TIMEOUT)
7      if (to == false) then
8          Wm <- t1 - t2 + (t2 - t1 + t4 - t3)/2
9          Wmedian <- median(Wm)
10         WRTT <- t4 - t1 size(W)
11         RTTf <- min(WRTT[size(WRTT)/2, size(WRTT)])
12         RTTl <- min(WRTT[0, size(WRTT)/2])
13         if ((|RTTf - RTTl| <= errRTT * min(WRTT)) then
14             if (epoch >= presynck + P)) then
15                 presynck <- true
16             end if
17         else
18             synck <- false, Wmedian <- 0
19             Wm <- 0, errsync <- epoch, n_to <- 0
20             epoch_sync <- INT_MAX - P, pre_sync <- INT_MAX - P
21             set(0, 0, NOSYNC)
22         end if
23         if ((synck == true) && (epoch >= epochsync + P)) then
24             (m, c) <- linear_fit(Wmedian)
25             actual_c <- c
26             actual_m <- (1-alpha) * m + alpha * actual_m
27             epochsync <- epoch, n_to <- 0
28             set(actual_m, actual_c, SYNC)
29         else
30             if (epoch == errsync + MEDIAN_MAX_SIZE) then
31                 presync <- epoch
32             end if
33             if (epoch >= presync + P) then
34                 (actual_m, actual_c) <- linear_fit(Wmedian)
35                 synck <- true , epoch_sync <- epoch
36                 set(actual_m, actual_c, PRESYNC)
37             end if
38         end if
39     else
40         to <- false
41     end if
42 end for
=====

```

Figure 4: Formal description of sic.--

Several conditions should be verified to pass from NOSYNC to PRESYNC. First, the "else" condition of line 29 should occur, and also the elapsed time between errsync and actual epoch should be MEDIAN_MAX_SIZE (30-32). Therefore, when it also P time is passed from presync, the condition on line 33 is true, and the system arrives at PRESYNC, providing an initial estimation of ϕ .

Then, if there is no route change, the condition in line 14 will be true when the time was increased in another P period. Then, the system is in SYNC state, and it provides the estimation of $\phi(t)$ in line 28. Notice that every P time the estimation of $\phi(t)$ is computed unless a route change occurs (lines 13 and 17-22).

The function in line 6: (epoch, t1, t2, t3, t4, to) <- send_sic_packet(SERVER_IP, TIMEOUT), has a special treatment. It sends the packets specified in Section 3.3, which have signatures. To avoid the processing delay caused by the signature computation, we implemented a policy to send the signature of the previous packet, and if an error is detected, we can stop the synchronization just one loop ahead.

Figure 5 illustrates how the client side MUST implement the function send_sic_p (SERVER_IP, TIMEOUT). This function computes the timestamp t1 in line 1, build and send the UDP packet in lines 2-3. Then, if there is no timeout, it calculates the t4 timestamp (line 5), and if no packets were lost, verifies the signature of the previous one in lines 8-18. If the signature is not valid with the received certificate, then the system MUST change to NOSYNC state immediately (see line 11). NOSYNC state MUST also be set, if the limit of time without receiving packets MAX_to is reached. Finally, it stores the received packet into prev_rcv_pck (a global variable) to use in the next packet (line 19). Notice that n_to, the lost packets, is a global variable, as well as the epoch of the previous packet: e_prev.

```

=====
|               function: send_sic_p(server, TIMEOUT)               |
=====
1  t1 <- get_timestamp()
2  sic_P <- sic_pck(t1, 0, 0, prev_sig)
3  (to, rcv_sic_pck) <- send(sic_P,UDP_PORT, SERVER_IP, TIMEOUT)
4  if (to == false) then
5      |   t4 <- get_timestamp()
6      |   epoch <- trunc_to_seconds(t1)
7      |   prev_sig <- get_signature(sic_P)
8      |   if (epoch - e_prev <= RUNNING_TIME) then
9      |       |   if (n_to < MAX_to) then
10     |           |   if (verify(prev_rcv_pck,rcv_sic.CERT) == false) then
11     |               |   |   set(0, 0, NOSYNC)
12     |               |   |   else
13     |               |   |   |   n_to <- 0,  e_prev <- epoch
14     |               |   |   |   end if
15     |               |   |   else
16     |               |   |   |   set(0, 0, NOSYNC)
17     |               |   |   |   end if
18     |               |   |   end if
19     |               |   prev_rcv_pck <- rcv_sic_pck
20     |               |   t2 <- rcv_sic_pck.t2
21     |               |   t3 <- rcv_sic_pck.t3
22 else
23     |   n_to <- n_to + 1
24 end if
25 return (epoch, t1, t2, t3, t4, to)
=====

```

Figure 5: The send_sic_p function.--

The server sic algorithm is presented in Figure 6. It uses `prev_sic_P{}`, which is a structure to store the received previous signatures, indexed by the IP client addresses (`CLIENT_add` contains its IP and UDP port); and the same for `prev_sig{}` with the previously sent signatures. Line 6 verifies either signature is null because it is the first packet, or it is a valid signature. In both cases, the algorithm process the packet computing `t3`, building up the sic frequency packet, sending it and computing its signature (stored to send in the next reply) in lines 7-11. Next, the actual packet is stored in the `prev_sic_P{}` structure, line 13.

```

=====
|                               sic Server alorihm                               |
=====
1  prev_sic_P{} <- null, prev_sig{} <-- null
2  while (RUNNING == true) then
3      if (receive() == true) then
4          t2 <- get_timestamp()
5          prev_sig <- get_signature(prev_sic_P{receive().CLIENT_add})
6          if (prev_sig == null) ||
              (verify(prev_sig, CLIENT_add.CERT) == true) then
7              t3 <- get_timestamp()
8              sic_P<-sic_pack(t1, t2, t3, prev_sig)
9              send(sic_P, CLIENT_add.UDP, CLIENT_add.IP, TIMEOUT)
10             prev_sig <- get_signature(sic_P)
11             prev_sig{receive().CLIENT_add} <- prev_sig
12         end if
13     prev_sic_P{receive().CLIENT_add} <- receive().sic_pack
14 end if
15 end while
=====

```

Figure 6: Algorithm sic for the Server.--

3.2. Protocol definitions

We provide a formal definition of each used constant and variables; the RECOMMENDED values are displayed in parentheses at the end of the description. These constant and variables MUST be represented in a sic frequency implementation. All the types MUST be respected. They are expressed in "C" programming language running on a 64-bit processor.

a. Constants used for the sic frequency algorithm (Figure 4)

1. RUNNING_TIME: is the period between sic packets are sent (1 second).
2. MEDIAN_MAX_SIZE: is the window size used to compute the median of the measurements (600).
3. P: is the period between phi's estimation (60).
4. alpha: is a float in the [0,1], the coefficient of the autoregressive estimation of the slope of phi(t) (0.05).
5. TIMEOUT: is the maximum time in seconds that a sic packet reply is expected (0.8 seconds).

6. SERVER_IP: is the IP address of the server (@IP in version 4 or 6).
 7. errRTT: is a float that bounds the maximum difference to detect a route change (0.2).
 8. MAX_to: is an integer representing the maximum number of packet lost ($P/10$).
 9. CERT: is a public certificate of the other end, it is used to verify signs of the packets.
 10. UDP_PORT: is an integer with the port UDP where the service is running on the server. (4444)
 11. SERVER_IP: is the IP address of the server.
 12. CLIENT_IP: is the IP address of the client.
- b. States used for the sic frequency algorithm (Figure 4)
1. NOSYNC: a boolean indicates that it is not possible to correct the local time.
 2. PRESYNC: an integer indicates that sic is almost (P RUNNING_TIME) seconds from the synchronization.
 3. SYNC: a boolean indicates that sic is synchronized.
- c. Variables used for the sic frequency algorithms (Figure 4, Figure 5 and Figure 6)
1. errsync: is an integer with the UNIX timestamp epoch of the initial NOSYNC cycle. It is used to complete the window or measurements (W_m) to compute their medians.
 2. presync: is an integer with the UNIX timestamp epoch of the initial PRESYNC cycle. It is used to wait until (P RUNNING_TIME) seconds to the linear fit of $\phi(t)$.
 3. syncck: is an integer with the UNIX timestamp epoch of the initial SYNC cycle. Every P RUNNING_TIME) seconds the $\phi(t)$ function is estimated.
 4. epochsync: is an integer with the last UNIX timestamp epoch of synchronization. It is used to compute a new estimation of $\phi(t)$, every (P RUNNING_TIME) seconds.

5. epoch: is an integer with UNIX timestamp in seconds. It carries the initial epoch of each sic measurement packet.
6. t1, t2, t3, t4: are long long integers to store the t UNIX timestamps in microseconds.
7. actual_m : is a double with the slope for the $\phi(t)$ estimation.
8. actual_c: is a double with the intercept for the $\phi(t)$ estimation.
9. Wm: is an array of doubles of MEDIAN_MAX_SIZE. It stores the instantaneous estimates of $\phi(t)$.
10. Wmedian: is an array of doubles of P size. It saves the computed medians of Wm every RUNNING_TIME.
11. WRTT: is an array of doubles of (2 P) size. It stores the calculated RTT of last measurements.
12. RTTl: is a double with the minimum of last P RTTs. It is used to detect changes on the route from the client to the server.
13. RTTf: is a double with the minimum of previous P RTTs. It is used to detect changes on the route from the client to the server.
14. n_to: is an integer representing the number of lost packets in the actual synchronization window P.
15. e_prev: is an integer with the UNIX timestamp epoch of the last valid packet.
16. prev_rcv_pck: is a sic packet structure, the previous received one.

3.3. Protocol packet specification

The sic frequency uses UNIX microsecond format timestamps. Regarding Figure 2, the client takes a timestamp t1 just before it sends the packet. When the server receives the packet, it immediately computes t2, and just before it is sent back to the client, it computes t3. When the client receives the packet, it calculates t4.

The server does not need the timestamp t_1 because the proposed protocol synchronizes a client with the server clock. This information could however be useful for the server for future use.

The packets are shown in Figure 7. They MUST be sent as UDP data, and it MUST have five fields. The first three correspond to t_1 (client), t_2 (server), and t_3 (server); the last one is the signature of the previous message of the sender (client o server) with its private key. The timestamps t_1 , t_2 , and t_3 MUST be the UNIX timestamp in microseconds represented with a long long integer of 64-bit C language.

The client and server certificates SHOULD be valid and signed ones (only for experimentation user MAY use autogenerated ones).

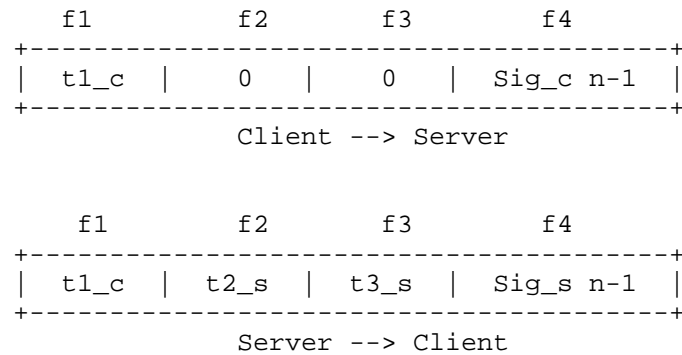


Figure 7: Packet format for the sic protocol.--

3.4. Minimum sic deployment

To deploy the sic frequency algorithm, as a minimum a Server and one Client are needed. The Server can support multiple clients. The maximum number of clients is for further study. The Server clock is considered the master one, and all clients synchronize with it. The Server side runs sic frequency as a server with a UDP_PORT number, as specified by the algorithm shown in Figure 6.

Client sic runs the algorithm shown in Figure 4 and also SHOULD provide the corrected time as

$$t = \text{actual_c} + \text{actual_m} * \text{timestamp} \quad (8)$$

Figure 8

Different ways of doing this task are possible:

Providing a client capable of reading the variables `actual_m` and `actual_c` in shared memory and producing the result of Equation 8.

Providing a service in a UDP port answering the correcter timestamp queries with Equation 8.

Other solution.

4. Implementation of sic frequency protocol

In this section we present the prove of the sic concept through some test that we already performed, and the current implementation of sic in C language. Our implementation is publicly available [sic-implementation]. Currently, the authentication process requiring transport of packet signatures is under development.

@@@ We started with a version to test sic without the MitM protection; soon we will finish with the secured version.

This protocol implements protection against MitM attacks. The identity of endpoints is guarantee by signed certificates using the deterministic Elliptic Curve Digital Signature Algorithm (ECDSA) specified in the [RFC6979]. Server and Client should use signed and valid ECDSA certificates to ensure their identity, and each side has is responsible to verify the public certificate of the other side before to run the algorithm in Figure 4.

4.1. Evaluation

To verify the sic proposal, we tested it using three hosts with GPS units. The first two were located at Buenos Aires, and the third at Los Angeles. We slightly modified the algorithm in Figure 4 to trigger each measurement using the PPS (pulse per second) signal provided by the GPS unit. Then, recording the client and server clocks with the PPS signal, we can determine the real phi function of Equation 1, within the GPS error (it is several orders of magnitude smaller than the error of the sic frequency protocol).

We use MTIE defined as follows (Maximum Time Interval Error, see [ToIM1996]):

$$MTIE = \max [\phi(t')] - \min [\phi(t)] , \quad (9)$$

for every t' and t in the interval $[t, t+s]$; and we chose $s=60$ seconds. We first used two host (RaspBerriesPI-2) connected back to back to analyze the minimum achievable precisiton, yielding a MTIE of 15.8 microseconds for the 90 percentile. Then, we selected two real cases of study, one national and other international. In Figure 9 we

show the result of the MTIE, evaluated in 60 seconds intervals, for the experiment Buenos Aires-Buenos Aires (RTT of 10ms) and Buenos Aires-Los Angeles (RTT of 198ms). The percentile 90 corresponds to 18.35 microseconds for the Buenos Aires case, and 25.4 microseconds for the Los Angeles case. The percentile 97.5 corresponds to 30 microseconds for the Buenos Aires case, and 42 microseconds for the Los Angeles case. We display the quartiles in Figure 10 . These measurements were performed during a week in each case.

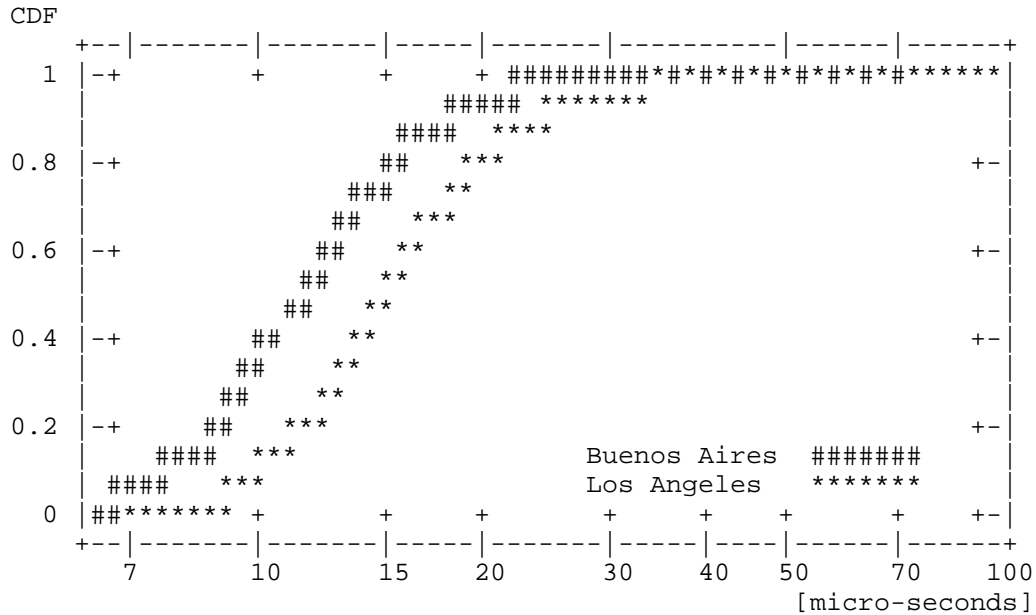


Figure 9: Cumulative distribution function of the MTIE (60s).--

	Buenos Aires (10ms)	Los Angeles (198ms)
Q3	14.69	19.29
Q2	11.60	14.93
Q1	9.41	12.26

Figure 10: Table with MTIE quartiles for two RTT cases (the numbers indicate microseconds).--

5. Conclusions

This document presents the sic algorithm to synchronize host clock frequency using the Internet and resistant to MitM attacks. It also shows the complete specification, implementation, and experiments results that support its working principle. In particular, sic frequency provides a clock rate stability of less than 1ppm for most of the time.

6. Security Considerations

Following [RFC7384] enumeration of Time Protocols in packet-switched networks, the proposed encryption of timing packets, based on a mechanism of secure key distribution, provides the following characteristics:

- 3.2.1 Packet Manipulation: Prevented by packet signature.
- 3.2.2 Spoofing: Prevented by packet signature and secure key distribution.
- 3.2.3 Replay Attack : Prevented by chain signing of packets.
- 3.2.4 Rogue Master Attack: Prevented by secure key distribution.
- 3.2.5 Packet Interception and Removal: If several packets are removed, the protocol does not arrive to SYNC state.
- 3.2.6 Packet Delay Manipulation: Not prevented. Future versions may prevent this using over-specification of timing (using redundant masters)
- 3.2.7 L2/L3 DoS attacks: Not prevented. This can be prevented in future versions using over-specification of timing and redundant masters time servers.
- 3.2.8 Cryptographic performance attacks: Not an issue in ECDSA.
- 3.2.9 DoS attacks against the time protocol: Prevented by secure key distribution.
- 3.2.10 Grandmaster Time source attack (GPS attacks): Not prevented. Future versions may prevent this using over-specification of timing (using several time servers) .
- 3.2.11 Exploiting vulnerabilities in the time protocol: Not prevented, future vulnerabilities are unknown.

3.2.12 Network Reconnaissance: Not provented in this version. No contermesasures were done in node anonymization.

7. IANA Considerations

This memo makes no requests of IANA.

8. Acknowledgements

The authors thank to Ethan Katz-Bassett, Zahaib Akhtar, the USC and CAIDA for lodging the testbed of sic frequency.

9. References

9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <<https://www.rfc-editor.org/info/rfc5905>>.
- [RFC6979] Pornin, T., "Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA)", RFC 6979, DOI 10.17487/RFC6979, August 2013, <<https://www.rfc-editor.org/info/rfc6979>>.
- [RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", RFC 7384, DOI 10.17487/RFC7384, October 2014, <<https://www.rfc-editor.org/info/rfc7384>>.
- [RFC8173] Shankarkumar, V., Montini, L., Frost, T., and G. Dowd, "Precision Time Protocol Version 2 (PTPv2) Management Information Base", RFC 8173, DOI 10.17487/RFC8173, June 2017, <<https://www.rfc-editor.org/info/rfc8173>>.

9.2. Informative References

- [alfa-estables] Uchaikin, V. and V. Zolotarev, "Chance and stability: stable distributions and their applications.", Walter de Gruyter. (Book), 1999.

[ITU-G.8260]

"Definitions and terminology for synchronization in packet networks (Recommendation ITU-T G.8260)", August 2015.

[sic-implementation]

Samariego, E., Ortega, A., and J. Alvarez-Hamelin,
"Synchronizing Internet Clocks",
<https://github.com/CoNexDat/SIC>, February 2018.

[ToIM1996]

Bregni, S., "Measurement of maximum time interval error for telecommunications clock stability characterization",
Bregni S. Measurement of maximum time interval error for telecommunicIEEE transactions on instrumentation and measurement. 45(5):900-906, October 1996.

[ToN2008]

Veitch, D., Ridoux, J., and S. Korada, "Robust synchronization of absolute and difference clocks over networks.", IEEE.ACM Transactions on Networking (TON) 17.2 (2009): 417-430, 2009.

[traffic-stable]

Carisimo, E., Grynberg, S., and J. Alvarez-Hamelin,
"Influence of traffic in stochastic behavior of latency.",
7th PhD School on Traffic Monitoring and Analysis (TMA),
Ireland, Dublin,, June 2017.

Appendix A. Example of RTT to NTP servers

This appendix shows an experiment to measure the RTT and the distance in hops from four different points to a time server in Buenos Aires city (the capital of Argentina). We did the measures two times from the four points, and we used one hundred packets to determine some statistical parameters. Next traceroute measurements show that the number of hops and RTT are very different from each point also changes a lot. For instance, taking a distinctive look at the STD, average, and maximum is possible to detect huge variations. We provide here a case in Argentina, trying to reach an NTP server from 4 different points at the Buenos Aires city.

```
host1$ mtr -r -c 100 time.afip.gov.ar
```

```
Start: Tue Mar 27 19:03:51 2018
```

HOST:	raspbian-server	Loss%	Snt	Last	Avg	Best	Wrst	StDev
1.	-- gw-vlan-srv.innova-red.ne	0.0%	100	2.2	2.8	2.1	37.7	4.9
2.	-- rnoc5.BUENOS-AIRES.innova	0.0%	100	2.3	3.8	2.1	55.8	7.9
3.	-- 10.5.10.2	0.0%	100	2.5	2.6	2.2	3.1	0.0

4.	-- 200.0.17.104	0.0%	100	3.1	3.1	2.4	13.7	1.1
5.	-- 172.18.2.53	0.0%	100	4.8	5.7	3.8	12.4	1.7
6.	-- time.afip.gob.ar	0.0%	100	5.2	5.2	3.8	12.0	1.3

```
host1$ mtr -r -c 100 time.afip.gov.ar
```

```
Start: Tue Mar 27 18:57:06 2018
```

HOST:	raspbian-server	Loss%	Snt	Last	Avg	Best	Wrst	StDev
1.	-- gw-vlan-srv.innova-red.ne	0.0%	50	2.4	2.8	2.0	34.2	4.5
2.	-- rnoc5.BUENOS-AIRES.innova	0.0%	50	2.1	3.8	2.1	52.8	7.7
3.	-- 10.5.10.2	0.0%	50	2.7	2.9	2.2	15.6	1.8
4.	-- 200.0.17.104	0.0%	50	2.8	3.0	2.3	3.9	0.0
5.	-- 172.18.2.53	0.0%	50	4.5	5.8	3.8	17.8	2.2
6.	-- time.afip.gob.ar	0.0%	50	4.7	9.9	4.2	238.5	33.0

```
host2$ mtr -r -c 100 time.afip.gov.ar
```

```
Start: Tue Mar 27 19:03:47 2018
```

HOST:	ws-david	Loss%	Snt	Last	Avg	Best	Wrst	StDev
1.	-- 10.10.96.1	0.0%	100	0.3	0.2	0.2	0.3	0.0
2.	-- 200.16.116.171	0.0%	100	1.0	5.9	0.6	158.4	22.9
3.	-- static.33.229.104.190.cps	1.0%	100	1.6	2.5	1.5	80.6	8.0
4.	-- static.129.192.104.190.cp	0.0%	100	2.1	1.9	1.8	3.0	0.1
5.	-- 200.0.17.104	1.0%	100	2.0	2.2	1.8	9.4	0.7
6.	-- 172.18.2.53	0.0%	100	3.2	4.2	3.1	12.0	1.5
7.	-- auth.afip.gob.ar	0.0%	100	4.2	4.5	3.3	9.8	1.2

```
host2$ mtr -r -c 100 time.afip.gov.ar
```

```
Start: Tue Mar 27 18:57:00 2018
```

HOST:	ws-david	Loss%	Snt	Last	Avg	Best	Wrst	StDev
1.	-- 10.10.96.1	0.0%	50	0.3	0.3	0.2	0.7	0.0
2.	-- 200.16.116.171	0.0%	50	0.9	6.7	0.7	196.5	29.1
3.	-- static.33.229.104.190.cps	2.0%	50	1.6	1.7	1.5	2.2	0.0
4.	-- static.129.192.104.190.cp	0.0%	50	2.1	2.0	1.7	2.4	0.0
5.	-- 200.0.17.104	0.0%	50	2.0	2.1	1.8	2.6	0.0
6.	-- 172.18.2.53	0.0%	50	4.8	4.3	3.2	9.1	1.3
7.	-- time.afip.gob.ar	0.0%	50	4.3	9.4	3.3	234.9	32.7

```
host3$ mtr -r -c 100 time.afip.gov.ar
```

```
Start: 2018-03-27T19:03:51-0300
```

HOST:	aleph.local	Loss%	Snt	Last	Avg	Best	Wrst	StDev
1.	-- 10.17.71.254	0.0%	100	4.7	30.3	3.5	280.4	52.4
2.	-- 10.255.254.250	0.0%	100	2.5	31.1	1.8	285.4	59.2
3.	-- 209.13.133.10	0.0%	100	30.5	43.9	2.3	237.2	64.9
4.	-- host169.advance.com.ar	3.0%	100	36.0	64.8	3.1	404.4	86.9
5.	-- 200.32.33.33	2.0%	100	106.9	70.6	2.8	315.0	80.4

6.	-- 200.32.34.66	5.0%	100	93.1	56.8	2.7	336.1	74.5
7.	-- 200.32.33.38	7.0%	100	42.8	58.0	2.9	357.8	76.7
8.	-- 209.13.139.211	4.0%	100	46.2	56.7	2.8	298.8	69.9
9.	-- 209.13.139.209	1.0%	100	84.5	57.1	2.6	277.7	72.3
10.	-- 209.13.166.211	1.0%	100	43.4	63.5	3.2	390.6	78.7
11.	-- 200.32.34.137	2.0%	100	68.7	64.1	3.7	259.5	75.5
12.	-- 200.32.33.37	0.0%	100	4.1	56.9	3.3	249.6	64.3
13.	-- 200.32.34.121	3.0%	100	10.9	65.0	4.1	415.7	85.1
14.	-- 200.32.33.241	2.0%	100	252.6	61.8	3.8	355.9	74.5
15.	-- 200.16.206.198	3.0%	100	188.0	54.6	3.1	461.7	74.9
16.	-- 172.18.2.53	2.0%	100	133.4	53.1	4.3	402.1	69.2
17.	-- time.afip.gob.ar	4.0%	100	72.5	54.1	4.9	343.2	66.9

host3\$ mtr -r -c 100 time.afip.gov.ar

Start: 2018-03-27T18:57:05-0300

HOST: aleph.local	Loss%	Snt	Last	Avg	Best	Wrst	StDev	
1.	-- 10.17.71.254	0.0%	50	125.6	88.1	3.7	392.4	79.3
2.	-- 10.255.254.250	0.0%	50	62.1	54.8	2.1	333.2	68.0
3.	-- 209.13.133.10	0.0%	50	4.0	33.9	2.4	280.8	51.3
4.	-- host169.advance.com.ar	2.0%	50	4.1	21.3	2.9	236.7	40.4
5.	-- 200.32.33.33	2.0%	50	4.5	32.2	3.2	341.3	69.4
6.	-- 200.32.34.66	4.0%	50	7.7	26.0	3.5	278.8	55.8
7.	-- 200.32.33.38	2.0%	50	4.8	29.4	3.0	221.3	43.4
8.	-- 209.13.139.211	0.0%	50	84.8	40.3	3.1	250.4	53.0
9.	-- 209.13.139.209	0.0%	50	25.1	35.0	2.8	249.2	55.4
10.	-- 209.13.166.211	0.0%	50	3.7	33.5	2.6	188.9	54.3
11.	-- 200.32.34.137	0.0%	50	5.6	28.2	3.7	224.3	51.1
12.	-- 200.32.33.37	0.0%	50	3.7	24.2	3.5	189.5	44.9
13.	-- 200.32.34.121	0.0%	50	4.7	30.8	4.0	213.2	51.6
14.	-- 200.32.33.241	0.0%	50	14.4	33.1	3.9	364.6	67.2
15.	-- 200.16.206.198	0.0%	50	5.0	58.2	3.1	300.7	88.5
16.	-- 172.18.2.53	0.0%	50	9.4	117.8	4.4	315.1	103.4
17.	-- time.afip.gob.ar	0.0%	50	199.6	120.2	5.2	484.0	96.2

host4\$ mtr -r -c 100 time.afip.gov.ar

Start: 2018-03-27T19:03:51-0300

HOST: cnet	Loss%	Snt	Last	Avg	Best	Wrst	StDev	
1.	-- 157.92.58.1	0.0%	100	6.6	2.8	0.3	12.8	2.5
2.	-- ???	100.0	100	0.0	0.0	0.0	0.0	0.0
3.	-- ???	100.0	100	0.0	0.0	0.0	0.0	0.0
4.	-- host98.131-100-186.static	0.0%	100	5.7	5.6	1.5	9.4	2.2
5.	-- host130.131-100-186.stati	0.0%	100	6.5	6.3	2.5	10.3	2.2
6.	-- 200.0.17.104	0.0%	100	2.4	2.7	2.3	15.6	1.4
7.	-- ???	100.0	100	0.0	0.0	0.0	0.0	0.0
8.	-- time.afip.gob.ar	0.0%	100	4.9	7.6	3.9	243.0	23.9

```
host4$ mtr -r -c 100 time.afip.gov.ar
```

```
Start: Tue Mar 27 18:41:40 2018
```

HOST: cnet	Loss%	Snt	Last	Avg	Best	Wrst	StDev
1. -- 157.92.58.1	0.0%	50	4.0	1.6	0.3	9.1	1.6
2. -- ???	100.0	50	0.0	0.0	0.0	0.0	0.0
3. -- ???	100.0	50	0.0	0.0	0.0	0.0	0.0
4. -- host98.131-100-186.static	0.0%	50	4.7	5.5	1.5	10.9	2.4
5. -- host130.131-100-186.stati	0.0%	50	8.4	6.5	2.6	10.5	2.2
6. -- 200.0.17.104	0.0%	50	2.5	2.8	2.3	11.0	1.2
7. -- ???	100.0	50	0.0	0.0	0.0	0.0	0.0
8. -- time.afip.gob.ar	0.0%	50	4.9	9.2	3.8	226.7	31.4

Authors' Addresses

Jose Ignacio Alvarez-Hamelin (editor)
Universidad de Buenos Aires
Av. Paseo Colon 850
Buenos Aires C1063ACV
Argentina

Phone: +54 11 5285-0716
Email: ihameli@cnet.fi.uba.ar
URI: <http://cnet.fi.uba.ar/ignacio.alvarez-hamelin/>

David Samaniego
Universidad de Buenos Aires
Av. Paseo Colon 850
Buenos Aires C1063ACV
Argentina

Phone: +54 11 5285-0716
Email: dsamanie@fi.uba.ar

Alfredo A. Ortega
Universidad de Buenos Aires
Av. Paseo Colon 850
Buenos Aires C1063ACV
Argentina

Phone: +54 11 5285-0716
Email: ortegaalfredo@gmail.com

Ruediger Geib
Deutsche Telekom
Heinrich-Hertz-Str. 3-7
Darmstadt 64297
Germany

Phone: +49 6151 5812747
Email: Ruediger.Geib@telekom.de

ippm
Internet-Draft
Intended status: Standards Track
Expires: May 23, 2021

F. Brockners, Ed.
Cisco
S. Bhandari
Thoughtspot
V. Govindan
C. Pignataro, Ed.
N. Nainar, Ed.
Cisco
H. Gredler
RtBrick Inc.
J. Leddy

S. Youell
JMPC

T. Mizrahi
Huawei Network.IO Innovation Lab

P. Lapukhov
Facebook

B. Gafni
A. Kfir

Mellanox Technologies, Inc.

M. Spiegel

Barefoot Networks, an Intel company

November 19, 2020

Geneve encapsulation for In-situ OAM Data
draft-brockners-ippm-ioam-geneve-05

Abstract

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information in the packet while the packet traverses a path between two points in the network. This document proposes a new Geneve tunnel option and outlines how IOAM data fields are carried in the option data field.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 23, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Conventions	3
2.1. Requirement Language	3
2.2. Abbreviations	3
3. IOAM Data Field Encapsulation in Geneve	3
4. Considerations	5
4.1. Discussion of the encapsulation approach	5
4.2. IOAM and the use of the Geneve O-bit	6
4.3. Transit devices	6
4.4. Additional Encapsulation Consideration	7
5. IANA Considerations	7
6. Security Considerations	7
7. Acknowledgements	7
8. Normative References	7
Authors' Addresses	8

1. Introduction

In-situ OAM (IOAM) records OAM information within the packet while the packet traverses a particular network domain. The term "in-situ" refers to the fact that the IOAM data fields are added to the data packets rather than is being sent within packets specifically dedicated to OAM. This document proposes a new Geneve tunnel option and defines how IOAM data fields are transported as part of the

tunnel option in the Geneve [I-D.ietf-nvo3-geneve] encapsulation. The IOAM data fields are defined in [I-D.ietf-ippm-ioam-data].

2. Conventions

2.1. Requirement Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2.2. Abbreviations

Abbreviations used in this document:

IOAM: In-situ Operations, Administration, and Maintenance

OAM: Operations, Administration, and Maintenance

Geneve: Generic Network Virtualization Encapsulation

3. IOAM Data Field Encapsulation in Geneve

Geneve is defined in [I-D.ietf-nvo3-geneve]. When Geneve encapsulation header is used, IOAM data fields can either be carried after the Geneve header, identified by the next protocol field or can be carried in the Geneve header itself as a tunnel option. The former approach is defined in [I-D.weis-ippm-ioam-eth] while the latter approach is defined in this document.

IOAM data fields are carried using a single Geneve Option Class TBD_IOAM. The different IOAM data fields defined in [I-D.ietf-ippm-ioam-data] are added as TLVs using that Geneve Option Class. In an administrative domain where IOAM is used, insertion of the IOAM header in Geneve is enabled at the Geneve tunnel endpoints, which also serve as IOAM encapsulating/decapsulating nodes by means of configuration.

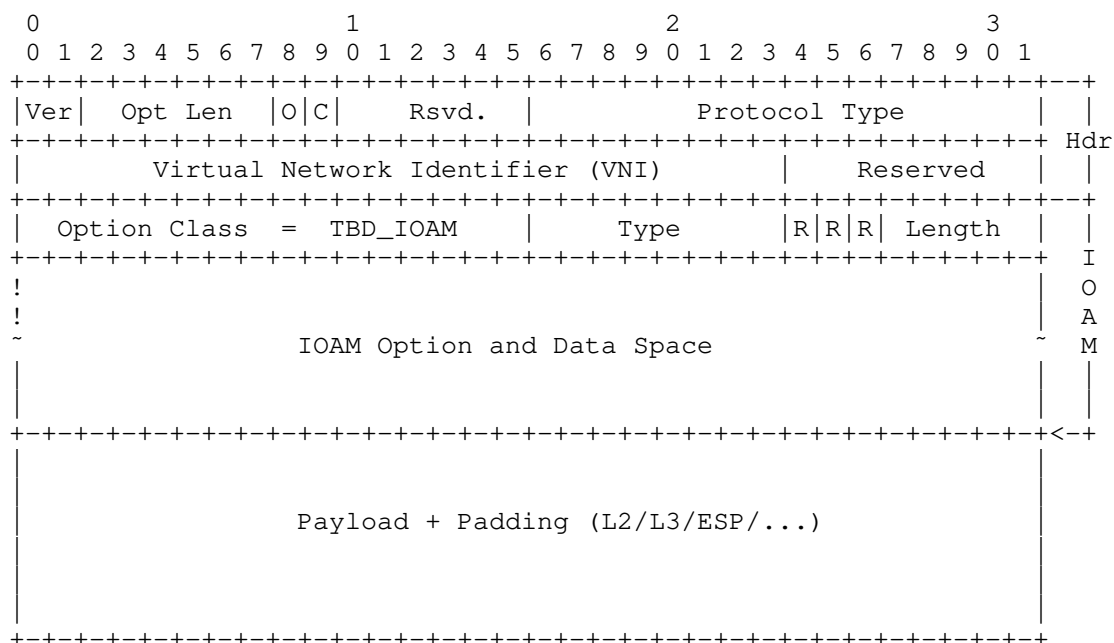


Figure 1: IOAM data encapsulation in Geneve

The Geneve header and fields are defined in [I-D.ietf-nvo3-geneve]. The Geneve Option Class value for use with IOAM is TBD_IOAM.

The fields related to the encapsulation of IOAM data fields in Geneve are defined as follows:

Option Class: 16-bit unsigned integer that determines the IOAM option class. The value is from the IANA registry setup for Geneve option classes as defined in [I-D.ietf-nvo3-geneve].

Type: 8-bit field defining the IOAM Option type, as defined in Section 7.2 of [I-D.ietf-ippm-ioam-data]. The 'C' bit MUST be set to zero.

R (3 bits): Option control flags reserved for future use. The flags MUST be set to zero on transmission and ignored on receipt.

Length: 5-bit unsigned integer. Length of the IOAM HDR in 4-octet units.

IOAM Option and Data Space: IOAM option header and data is present as defined by the Type field, and is defined in Section 4 of [I-D.ietf-ippm-ioam-data].

Multiple distinct IOAM Option-Types MAY be included within the same Geneve encapsulation. Each IOAM Option-Type MUST occur at most once within the same Geneve encapsulation. For example, if a Geneve encapsulation contains two IOAM Option-Types before a data payload, there would be two fields with TBD_IOAM Option Class each, differentiated by the Type field which specifies the type of the IOAM data included.

4. Considerations

This section summarizes a set of considerations on the overall approach taken for IOAM data encapsulation in Geneve, as well as deployment considerations.

4.1. Discussion of the encapsulation approach

This section is to support the working group discussion in selecting the most appropriate approach for encapsulating IOAM data fields in Geneve.

An encapsulation of IOAM data fields in Geneve should be friendly to an implementation in both hardware as well as software forwarders and support a wide range of deployment cases, including large networks that desire to leverage multiple IOAM data fields at the same time.

Hardware and software friendly implementation: Hardware forwarders benefit from an encapsulation that minimizes iterative look-ups of fields within the packet: Any operation which looks up the value of a field within the packet, based on which another lookup is performed, consumes additional gates and time in an implementation - both of which are desired to be kept to a minimum. This means that flat TLV structures are to be preferred over nested TLV structures. IOAM data fields are grouped into three option categories: Trace, proof-of-transit, and edge-to-edge. Each of these three options defines a TLV structure. A hardware-friendly encapsulation approach avoids grouping these three option categories into yet another TLV structure, but would rather carry the options as a serial sequence.

Total length of the IOAM data fields: The total length of IOAM data can grow quite large in case multiple different IOAM data fields are used and large path-lengths need to be considered. If for example an operator would consider using the IOAM trace option and capture node-id, app_data, egress/ingress interface-id, timestamp seconds, timestamps nanoseconds at every hop, then a total of 20 octets would be added to the packet at every hop. In case this particular deployment would have a maximum path length

of 15 hops in the IOAM domain, then a maximum of 300 octets of IOAM data were to be encapsulated in the packet.

Concerns with the current encapsulation approach:

Hardware support: Using Geneve tunnel options to encapsulate IOAM data fields leads to a nested TLV structure. Each IOAM data field option (trace, proof-of-transit, and edge-to-edge) represents a type, with the different IOAM data fields being TLVs within this the particular option type. Nested TLVs require iterative look-ups, a fact that creates potential challenges for implementations in hardware. It would be desirable to offer a way to encapsulate IOAM in a way that keeps TLV nesting to a minimum.

Length: Geneve tunnel option length is a 5-bit field in the current specification [I-D.ietf-nvo3-geneve] resulting in a maximum option length of 128 ($2^5 \times 4$) octets which constrains the use of IOAM to either small domains or a few IOAM data fields only. Support for large domains with a variety of IOAM data fields would be desirable.

4.2. IOAM and the use of the Geneve O-bit

[I-D.ietf-nvo3-geneve] defines an "O bit" for Control packets. Per [I-D.ietf-nvo3-geneve] the O bit indicates that the packet contains a control message instead of data payload. Packets that carry IOAM data fields in addition to regular data payload / customer traffic must not set the O bit. Packets that carry only IOAM data fields without any payload must set the O bit.

4.3. Transit devices

If IOAM is deployed in domains where UDP port numbers are not controlled and do not have a domain-wide meaning, such as on the global Internet, transit devices MUST NOT attempt to modify the IOAM data contained in the IOAM option class. In case UDP port numbers are not controlled there might be UDP packets, which leverage the UDP port number that Geneve utilizes, i.e. 6081, but the payload of these packets isn't Geneve. The scenario and associated reasoning is discussed in [RFC7605] which states that "it is important to recognize that any interpretation of port numbers -- except at the endpoints -- may be incorrect, because port numbers are meaningful only at the endpoints."

4.4. Additional Encapsulation Consideration

Geneve encapsulation header supports carrying IOAM data fields either as part of the tunnel option or as the protocol data unit that follows the Geneve header. An operator may choose to enable either of the options but it is not recommended to include both in the same data packet.

5. IANA Considerations

IANA is requested to allocate a Geneve "option class" numbers for IOAM:

Option Class	Description	Reference
x	TBD_IOAM	This document

6. Security Considerations

The security considerations of Geneve are discussed in [I-D.ietf-nvo3-geneve], and the security considerations of IOAM in general are discussed in [I-D.ietf-ippm-ioam-data].

IOAM is considered a "per domain" feature, where one or several operators decide on leveraging and configuring IOAM according to their needs. Still, operators need to properly secure the IOAM domain to avoid malicious configuration and use, which could include injecting malicious IOAM packets into a domain.

7. Acknowledgements

The authors would like to thank Eric Vyncke, Nalini Elkins, Srihari Raghavan, Ranganathan T S, Karthik Babu Harichandra Babu, Akshaya Nadahalli, Stefano Previdi, Hemant Singh, Erik Nordmark, LJ Wobker, Andrew Yourtchenko and Nagendra Kumar Nainar for the comments and advice.

8. Normative References

[I-D.ietf-ippm-ioam-data]

Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-10 (work in progress), July 2020.

[I-D.ietf-nvo3-geneve]

Gross, J., Ganga, I., and T. Sridhar, "Geneve: Generic Network Virtualization Encapsulation", draft-ietf-nvo3-geneve-16 (work in progress), March 2020.

[I-D.weis-ippm-ioam-eth]

Weis, B., Brockners, F., Hill, C., Bhandari, S., Govindan, V., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A., Gafni, B., Lapukhov, P., and M. Spiegel, "EtherType Protocol Identification of In-situ OAM Data", draft-weis-ippm-ioam-eth-04 (work in progress), May 2020.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <<https://www.rfc-editor.org/info/rfc2784>>.

[RFC3232] Reynolds, J., Ed., "Assigned Numbers: RFC 1700 is Replaced by an On-line Database", RFC 3232, DOI 10.17487/RFC3232, January 2002, <<https://www.rfc-editor.org/info/rfc3232>>.

[RFC7605] Touch, J., "Recommendations on Using Assigned Transport Port Numbers", BCP 165, RFC 7605, DOI 10.17487/RFC7605, August 2015, <<https://www.rfc-editor.org/info/rfc7605>>.

Authors' Addresses

Frank Brockners (editor)
Cisco Systems, Inc.
Hansaallee 249, 3rd Floor
DUESSELDORF, NORDRHEIN-WESTFALEN 40549
Germany

Email: fbrockne@cisco.com

Shwetha Bhandari
Thoughtspot
3rd Floor, Indiqube Orion, 24th Main Rd, HSR Layout
Bangalore, KARNATAKA 560 102
India

Email: shwetha.bhandari@thoughtspot.com

Vengada Prasad Govindan
Cisco Systems, Inc.

Email: venggovi@cisco.com

Carlos Pignataro (editor)
Cisco Systems, Inc.
7200-11 Kit Creek Road
Research Triangle Park, NC 27709
United States

Email: cpignata@cisco.com

Nagendra Kumar Nainar (editor)
Cisco Systems, Inc.
7200-11 Kit Creek Road
Research Triangle Park, NC 27709
United States

Email: naikumar@cisco.com

Hannes Gredler
RtBrick Inc.

Email: hannes@rtbrick.com

John Leddy
United States

Email: john@leddy.net

Stephen Youell
JP Morgan Chase
25 Bank Street
London E14 5JP
United Kingdom

Email: stephen.youell@jpmorgan.com

Tal Mizrahi
Huawei Network.IO Innovation Lab
Israel

Email: tal.mizrahi.phd@gmail.com

Petr Lapukhov
Facebook
1 Hacker Way
Menlo Park, CA 94025
US

Email: petr@fb.com

Barak Gafni
Mellanox Technologies, Inc.
350 Oakmead Parkway, Suite 100
Sunnyvale, CA 94085
U.S.A.

Email: gbarak@mellanox.com

Aviv Kfir
Mellanox Technologies, Inc.
350 Oakmead Parkway, Suite 100
Sunnyvale, CA 94085
U.S.A.

Email: avivk@mellanox.com

Mickey Spiegel
Barefoot Networks, an Intel company
4750 Patrick Henry Drive
Santa Clara, CA 95054
US

Email: mickey.spiegel@intel.com

ippm
Internet-Draft
Intended status: Standards Track
Expires: May 7, 2020

F. Brockners
S. Bhandari
V. Govindan
C. Pignataro
Cisco
H. Gredler
RtBrick Inc.
J. Leddy

S. Youell
JMPC
T. Mizrahi
Huawei Network.IO Innovation Lab
A. Kfir
B. Gafni
Mellanox Technologies, Inc.
P. Lapukhov
Facebook
M. Spiegel
Barefoot Networks
November 4, 2019

VXLAN-GPE Encapsulation for In-situ OAM Data
draft-brockners-ippm-ioam-vxlan-gpe-03

Abstract

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information in the packet while the packet traverses a path between two points in the network. This document outlines how IOAM data fields are encapsulated in VXLAN-GPE.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 7, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Conventions	3
2.1. Requirement Language	3
2.2. Abbreviations	3
3. IOAM Data Field Encapsulation in VXLAN-GPE	3
4. Considerations	5
4.1. Discussion of the encapsulation approach	5
4.2. IOAM and the use of the VXLAN O-bit	6
4.3. Transit devices	6
5. IANA Considerations	6
5.1. VXLAN-GPE Next Protocol Value	6
5.2. LISP-GPE Next Protocol Value	7
6. Security Considerations	7
7. Acknowledgements	7
8. References	7
8.1. Normative References	7
8.2. Informative References	8
Authors' Addresses	9

1. Introduction

In-situ OAM (IOAM) records OAM information within the packet while the packet traverses a particular network domain. The term "in-situ" refers to the fact that the IOAM data fields are added to the data packets rather than being sent within packets specifically dedicated to OAM. This document defines how IOAM data fields are transported as part of the VXLAN-GPE [I-D.ietf-nvo3-vxlan-gpe] encapsulation. The IOAM data fields are defined in [I-D.ietf-ippm-ioam-data]. An implementation of IOAM which leverages VXLAN-GPE to carry the IOAM

data is available from the FD.io open source software project [FD.io].

2. Conventions

2.1. Requirement Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2.2. Abbreviations

Abbreviations used in this document:

IOAM: In-situ Operations, Administration, and Maintenance

OAM: Operations, Administration, and Maintenance

VXLAN-GPE: Virtual eXtensible Local Area Network, Generic Protocol Extension

3. IOAM Data Field Encapsulation in VXLAN-GPE

VXLAN-GPE is defined in [I-D.ietf-nvo3-vxlan-gpe]. IOAM data fields are carried in VXLAN-GPE using a next protocol value of TBD_IOAM. An IOAM header is added containing the different IOAM data fields defined in [I-D.ietf-ippm-ioam-data]. In an administrative domain where IOAM is used, insertion of the IOAM header in VXLAN-GPE is enabled at the VXLAN-GPE tunnel endpoints, which also serve as IOAM encapsulating/decapsulating nodes by means of configuration.

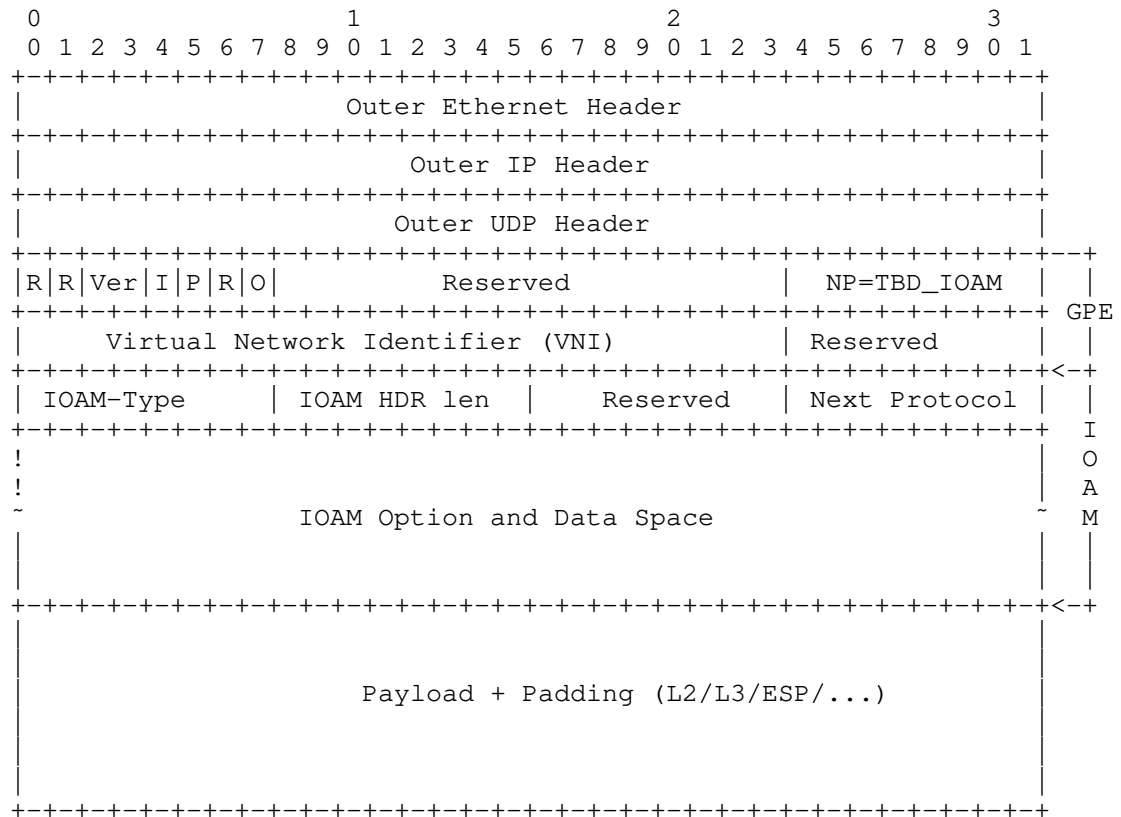


Figure 1: IOAM data encapsulation in VXLAN-GPE

The VXLAN-GPE header and fields are defined in [I-D.ietf-nvo3-vxlan-gpe]. The VXLAN Next Protocol value for IOAM is TBD_IOAM.

The IOAM related fields in VXLAN-GPE are defined as follows:

IOAM-Type: 8-bit field defining the IOAM Option type, as defined in Section 7.2 of [I-D.ietf-ippm-ioam-data].

IOAM HDR len: 8-bit unsigned integer. Length of the IOAM HDR in 4-octet units not including the first 4 octets.

Reserved: 8-bit reserved field MUST be set to zero upon transmission and ignored upon receipt.

Next Protocol: 8-bit unsigned integer that determines the type of header following IOAM protocol. The value is from the IANA

registry setup for VXLAN GPE Next Protocol defined in [I-D.ietf-nvo3-vxlan-gpe].

IOAM Option and Data Space: IOAM option header and data is present as specified by the IOAM-Type field, and is defined in Section 4 of [I-D.ietf-ippm-ioam-data].

Multiple IOAM options MAY be included within the VXLAN-GPE encapsulation. For example, if a VXLAN-GPE encapsulation contains two IOAM options before a data payload, the Next Protocol field of the first IOAM option will contain the value of TBD_IOAM, while the Next Protocol field of the second IOAM option will contain the VXLAN "Next Protocol" number indicating the type of the data payload.

4. Considerations

This section summarizes a set of considerations on the overall approach taken for IOAM data encapsulation in VXLAN-GPE, as well as deployment considerations.

4.1. Discussion of the encapsulation approach

This section is to support the working group discussion in selecting the most appropriate approach for encapsulating IOAM data fields in VXLAN-GPE.

An encapsulation of IOAM data fields in VXLAN-GPE should be friendly to an implementation in both hardware as well as software forwarders. Hardware forwarders benefit from an encapsulation that minimizes iterative look-ups of fields within the packet: Any operation which looks up the value of a field within the packet, based on which another lookup is performed, consumes additional gates and time in an implementation - both of which are desired to be kept to a minimum. This means that flat TLV structures are to be preferred over nested TLV structures. IOAM data fields are grouped into three option categories: Trace, proof-of-transit, and edge-to-edge. Each of these three options defines a TLV structure. A hardware-friendly encapsulation approach avoids grouping these three option categories into yet another TLV structure, but would rather carry the options as a serial sequence.

Two approaches for encapsulating IOAM data fields in VXLAN-GPE could be considered:

1. Use a single GPE protocol type for all IOAM types: IOAM would receive a single GPE protocol type code point. A "sub-type" field would then specify what IOAM options type (trace, proof-of-transit, edge-to-edge) is carried.

2. Use one GPE protocol type per IOAM options type: Each IOAM data field option (trace, proof-of-transit, and edge-to-edge) would be specified by its own "next protocol", i.e. each IOAM options type becomes its own GPE protocol type with a dedicated code point. This implies that in case additional IOAM option types would be added in the future, additional GPE protocol type code points would need to be allocated.

The first option has been chosen here. Multiple back-to-back IOAM options can be encoded as a succession of IOAM headers, with the same single GPE protocol type appearing as the next protocol before each IOAM header, but different sub-types within each IOAM header.

4.2. IOAM and the use of the VXLAN O-bit

[I-D.ietf-nvo3-vxlan-gpe] defines an "O bit" for OAM packets. Per [I-D.ietf-nvo3-vxlan-gpe] the O bit indicates that the packet contains an OAM message instead of data payload. Packets that carry IOAM data fields in addition to regular data payload / customer traffic must not set the O bit. Packets that carry only IOAM data fields without any payload must set the O bit.

4.3. Transit devices

If IOAM is deployed in domains where UDP port numbers are not controlled and do not have a domain-wide meaning, such as on the global Internet, transit devices MUST NOT attempt to modify the IOAM data contained in the IOAM header following the VXLAN-GPE header. In case UDP port numbers are not controlled there might be UDP packets specifying the same UDP port number that VXLAN-GPE utilizes, i.e. 4790, but with a payload that is not VXLAN-GPE. The scenario and associated reasoning is discussed in [RFC7605] which states that "it is important to recognize that any interpretation of port numbers -- except at the endpoints -- may be incorrect, because port numbers are meaningful only at the endpoints."

5. IANA Considerations

5.1. VXLAN-GPE Next Protocol Value

IANA is requested to allocate a value in the VXLAN-GPE "Next Protocol" registry for IOAM, which is defined in [I-D.ietf-nvo3-vxlan-gpe].

Next Protocol	Description	Reference
0x81	IOAM	This document

5.2. LISP-GPE Next Protocol Value

IANA is requested to allocate a value in the LISP-GPE "Next Protocol" registry for IOAM, which is defined in [I-D.ietf-lisp-gpe].

Next Protocol	Description	Reference
0x81	IOAM	This document

6. Security Considerations

The security considerations of VXLAN-GPE are discussed in [I-D.ietf-nvo3-vxlan-gpe], and the security considerations of IOAM in general are discussed in [I-D.ietf-ippm-ioam-data].

IOAM is considered a "per domain" feature, where one or several operators decide on leveraging and configuring IOAM according to their needs. Still, operators need to properly secure the IOAM domain to avoid malicious configuration and use, which could include injecting malicious IOAM packets into a domain.

7. Acknowledgements

The authors would like to thank Eric Vyncke, Nalini Elkins, Srihari Raghavan, Ranganathan T S, Karthik Babu Harichandra Babu, Akshaya Nadahalli, Stefano Previdi, Hemant Singh, Erik Nordmark, LJ Wobker, and Andrew Yourtchenko for the comments and advice.

8. References

8.1. Normative References

[ETYPES] "IANA Ethernet Numbers",
<<https://www.iana.org/assignments/ethernet-numbers/ethernet-numbers.xhtml>>.

[I-D.ietf-ippm-ioam-data]

Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., remy@barefootnetworks.com, r., daniel.bernier@bell.ca, d., and J. Lemon, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-08 (work in progress), October 2019.

[I-D.ietf-lisp-gpe]

Maino, F., Lemon, J., Agarwal, P., Lewis, D., and M. Smith, "LISP Generic Protocol Extension", draft-ietf-lisp-gpe-09 (work in progress), October 2019.

[I-D.ietf-nvo3-vxlan-gpe]

Maino, F., Kreeger, L., and U. Elzur, "Generic Protocol Extension for VXLAN", draft-ietf-nvo3-vxlan-gpe-08 (work in progress), October 2019.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <<https://www.rfc-editor.org/info/rfc2784>>.

[RFC3232] Reynolds, J., Ed., "Assigned Numbers: RFC 1700 is Replaced by an On-line Database", RFC 3232, DOI 10.17487/RFC3232, January 2002, <<https://www.rfc-editor.org/info/rfc3232>>.

[RFC7605] Touch, J., "Recommendations on Using Assigned Transport Port Numbers", BCP 165, RFC 7605, DOI 10.17487/RFC7605, August 2015, <<https://www.rfc-editor.org/info/rfc7605>>.

8.2. Informative References

[FD.io] "Fast Data Project: FD.io", <<https://fd.io/>>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<https://www.rfc-editor.org/info/rfc7665>>.

Authors' Addresses

Frank Brockners
Cisco Systems, Inc.
Hansaallee 249, 3rd Floor
DUESSELDORF, NORDRHEIN-WESTFALEN 40549
Germany

Email: fbrockne@cisco.com

Shwetha Bhandari
Cisco Systems, Inc.
Cessna Business Park, Sarjapura Marathalli Outer Ring Road
Bangalore, KARNATAKA 560 087
India

Email: shwethab@cisco.com

Vengada Prasad Govindan
Cisco Systems, Inc.

Email: venggovi@cisco.com

Carlos Pignataro
Cisco Systems, Inc.
7200-11 Kit Creek Road
Research Triangle Park, NC 27709
United States

Email: cpignata@cisco.com

Hannes Gredler
RtBrick Inc.

Email: hannes@rtbrick.com

John Leddy

Email: john@leddy.net

Stephen Youell
JP Morgan Chase
25 Bank Street
London E14 5JP
United Kingdom

Email: stephen.youell@jpmorgan.com

Tal Mizrahi
Huawei Network.IO Innovation Lab
Israel

Email: tal.mizrahi.phd@gmail.com

Aviv Kfir
Mellanox Technologies, Inc.
350 Oakmead Parkway, Suite 100
Sunnyvale, CA 94085
U.S.A.

Email: avivk@mellanox.com

Barak Gafni
Mellanox Technologies, Inc.
350 Oakmead Parkway, Suite 100
Sunnyvale, CA 94085
U.S.A.

Email: gbarak@mellanox.com

Petr Lapukhov
Facebook
1 Hacker Way
Menlo Park, CA 94025
US

Email: petr@fb.com

Mickey Spiegel
Barefoot Networks
2185 Park Boulevard
Palo Alto, CA 94306
US

Email: mspiegel@barefootnetworks.com

INTERNET-DRAFT

N. Elkins
Inside Products
G. Fioccola
Telecom Italia
M. Ackermann
BCBS Michigan
R. Hamilton

Intended Status: Proposed Standard
Expires: April 24, 2018

Chemical Abstract Services
October 21, 2018

IPv6 Marking and Performance and Diagnostic Metrics (MPDM)
draft-fear-ippm-mpdm-02

Abstract

To assess performance problems, this document describes optional headers embedded in each packet that provide marking, sequence numbers and timing information as a basis for measurements. Such measurements may be interpreted in real-time or after the fact. This document specifies the IPv6 Marking and Performance and Diagnostic Metrics (M-PDM) Hop-byHop and Destination Options extension headers.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

Copyright and License Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

IETF Trust Legal Provisions of 28-dec-2009, Section 6.b(i), paragraph 3: This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1	Background	4
1.1	Terminology	4
1.2	Rationale for defined solution	4
1.2.1	Alternate Marking Method Operation	5
1.2.1.1	Single Mark Measurement	5
1.2.2.2	Double Mark Measurement	5
1.3	IPv6 Transition Technologies	6
2	Measurement Information Derived from PDM	6
3	Marking and Performance and Diagnostic Metrics (M-PDM)	
	Destination	7
3.1	Destination Options Header	7
3.2.1	M-PDM Layout	7
3.2.2	Base Unit for Time Measurement	9
3.3	Header Placement	9
3.4	Header Placement Using IPSec ESP Mode	9
3.4.1	Using ESP Transport Mode	10
3.4.2	Using ESP Tunnel Mode	10
3.5	Implementation Considerations	10
3.5.1	M-PDM Activation	10
3.5.2	M-PDM Timestamps	10
3.6	Dynamic Configuration Options	11
3.7	Information Access and Storage	11
4	M-PDM HBH Option	12
4.1	HBH Timestamps In and Out	15
5	Security Considerations	15
5.1	Resource Consumption and Resource Consumption Attacks	15
5.2	Pervasive monitoring	15
5.3	M-PDM as a Covert Channel	16
5.4	Timing Attacks	16
6	IANA Considerations	17
7	References	17
7.1	Normative References	17
7.2	Informative References	18
	Acknowledgments	18
	Authors' Addresses	19

1 Background

To assess performance problems, measurements based on marking, sequence numbers and timing may be embedded in each packet. Such measurements may be interpreted in real-time or after the fact.

As defined in RFC8200 [RFC8200], destination options are carried by the IPv6 Destination Options extension header. Destination options include information that need be examined only by the IPv6 node given as the destination address in the IPv6 header, not by routers or "middle boxes".

RFC8200 [RFC8200] additionally defines the IPv6 Hop-by-Hop (HBH) Options extension header. This header may be processed and examined by end nodes, routers and "middle boxes".

This document specifies both the Marking Performance and Diagnostic Metrics (M-PDM) destination option as well as the M-PDM HBH Option.

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1.2 Rationale for defined solution

The M-PDM Destination Option and M-PDM Hop-By-Hop Option are a combining of PDM [RFC8250] with Marking [RFC8321] to obtain path and middle box information.

The Marking Field is designed as:

The 2 currently used bits from the 8 bit Marking field are designated as Mark Field (MF).

```
+--+--+--+--+--+--+--+
| reserved | MF |
+--+--+--+--+--+--+--+
```

Mark Field (MF) is:

```
+--+--+--+
| S | D |
+--+--+--+
```

1.2.1 Alternate Marking Method Operation

[RFC8321] describes in detail the methodology, that we briefly illustrate also here.

1.2.1.1 Single Mark Measurement

As explained in the [RFC8321], marking can be applied to delineate blocks of packets based either on equal number of packets in a block or based on equal time interval. The latter method offers better control as it allows better account for capabilities of downstream nodes to report statistics related to batches of packets and, at the same time, time resolution that affects defect detection interval.

If the Single Mark measurement used, then the D flag MUST be set to zero on transmit and ignored by monitoring point.

The S flag is used to create alternate flows to measure the packet loss by switching value of the S flag. Delay metrics MAY be calculated with the alternate flow using any of the following methods:

- First/Last Batch Packet Delay calculation: timestamps are collected based on order of arrival so this method is sensitive to packet loss and re-ordering.
- Average Packet Delay calculation: an average delay is calculated by considering the average arrival time of the packets within a single block. This method only provides single metric for the duration of the block and it doesn't give information about the delay distribution.

1.2.2.2 Double Mark Measurement

Double Mark method allows more detailed measurement of delays for the monitored flow but it requires more nodal and network resources. If the Double Mark method used, then the S flag MUST be used to create the alternate flow. The D flag MUST be used to mark single packets to measure delay jitter.

The first marking (S flag alternation) is needed for packet loss and also for average delay measurement. The second marking (D flag is put to one) creates a new set of marked packets that are fully identified and dedicated for delay. This method is useful to have not only the average delay but also to know more about the statistic distribution of delay values.

1.3 IPv6 Transition Technologies

In the path to full implementation of IPv6, transition technologies such as translation or tunneling may be employed. It is possible that an IPv6 packet containing M-PDM may be dropped if using IPv6 transition technologies. For example, an implementation using a translation technique (IPv6 to IPv4) which does not support or recognize the IPv6 Destination Options extension header or a new HBH option may simply drop the packet rather than translating it without the extension header.

It is also possible that some devices in the network may not correctly handle multiple IPv6 Extension Headers, including the IPv6 Destination Option. For example, adding the PDM header to a packet may push the layer 4 information to a point in the packet where it is not visible to filtering logic, and may be dropped. This kind of situation is expected to become rare over time.

2 Measurement Information Derived from PDM

Each packet contains information about the sender and receiver. In IP protocol, the identifying information is called a "5-tuple".

The 5-tuple consists of:

SADDR : IP address of the sender
SPORT : Port for sender
DADDR : IP address of the destination
DPORT : Port for destination
PROTC : Protocol for upper layer (ex. TCP, UDP, ICMP, etc.)

The PDM contains the following base fields:

PSNTP : Packet Sequence Number This Packet
PSNLR : Packet Sequence Number Last Received
DELTATLR : Delta Time Last Received
DELTATLS : Delta Time Last Sent

This information, combined with the 5-tuple, allows the measurement of the following metrics:

1. Round-trip delay
2. Server delay

These are further described in RFC8250 [RFC8250].

Performance measurements described in [RFC8321] are allowed.

3 Marking and Performance and Diagnostic Metrics (M-PDM) Destination Option Layout

3.1 Destination Options Header

The IPv6 Destination Options Header is used to carry information that needs to be examined only by a packet's destination node(s). The Destination Options Header is identified by a Next Header value of 60 in the immediately preceding header and is defined in RFC8200 [RFC8200]. The IPv6 Marking and Performance and Diagnostic Metrics Destination Option (M-PDM) is implemented as an IPv6 Option carried in the Destination Options Header. M-PDM does not require time synchronization.

3.2 Marking and Performance and Diagnostic Metrics (M-PDM) Destination Option

3.2.1 M-PDM Layout

The IPv6 Marking and Performance and Diagnostic Metrics Destination Option (M-PDM) contains the following fields:

```

PSNTP      : Packet Sequence Number This Packet
PSNLR      : Packet Sequence Number Last Received
DELTATLR   : Delta Time Last Received
DELTATLS   : Delta Time Last Sent

```

PDM has alignment requirements. Following the convention in IPv6, these options are aligned in a packet so that multi-octet values within the Option Data field of each option fall on natural boundaries (i.e., fields of width n octets are placed at an integer multiple of n octets from the start of the header, for n = 1, 2, 4, or 8) [RFC8200].

The M-PDM destination option is encoded in type-length-value (TLV) format 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	3	4	5	6	7	8	9										
Option Type										Option Length										Vrsn										RSVD										Marking									
PSN This Packet										PSN Last Received																																							
Delta Time Last Sent										Delta Time Last Received																																							

Option Type

TBD = 0xXX (TBD) [To be assigned by IANA] [RFC2780]

In keeping with RFC8200 [RFC8200], the two high-order bits of the Option Type field are encoded to indicate specific processing of the option; for the PDM destination option, these two bits MUST be set to 00.

The third high-order bit of the Option Type field specifies whether or not the Option Data of that option can change en route to the packet's final destination.

In M-PDM, the value of the third high-order bit MUST be 0.

Option Length

8-bit unsigned integer. Length of the option, in octets, excluding the Option Type and Option Length fields. This field MUST be set to 10.

Version

4-bit unsigned integer.

Reserved

4-bit unsigned integer.

Marking

8-bit unsigned integer. (2 currently used - 6 reserved)

Packet Sequence Number This Packet (PSNTP)

16-bit unsigned integer. This field will wrap. It is intended for use while analyzing packet traces.

This field is initialized at a random number and incremented monotonically for each packet of the session flow of the IP stack. The random-number initialization is intended to make it harder to spoof and insert such packets.

Packet Sequence Number Last Received (PSNLR)

16-bit unsigned integer. This is the PSNTP of the packet last received by the IP stack.

This field is initialized to 0.

Delta Time Last Sent (DELTATLS)

16-bit unsigned integer.

Delta Time Last Sent = (receive time packet n - send time packet (n - 1))

Delta Time Last Received (DELTATLR)

16-bit unsigned integer.

Delta Time Last Received = (send time packet n - receive time packet (n - 1))

3.2.2 Base Unit for Time Measurement

Fixed base. TBD. [More information needs to be added here.]

3.3 Header Placement

The M-PDM Destination Option is placed as defined in RFC8200 [RFC8200]. There may be a choice of where to place the Destination Options header. If using ESP mode, please see section 3.4 of this document for placement of the M-PDM Destination Options header.

For each IPv6 packet header, the M-PDM MUST NOT appear more than once. However, an encapsulated packet MAY contain a separate M-PDM associated with each encapsulated IPv6 header.

3.4 Header Placement Using IPsec ESP Mode

IPsec Encapsulating Security Payload (ESP) is defined in [RFC4303] and is widely used. Section 3.1.1 of [RFC4303] discusses placement of Destination Options Headers.

The placement of M-PDM is different depending on if ESP is used in tunnel or transport mode.

In ESP case, no 5-tuple is available, as there are no port numbers. ESP flow should be identified only by using SADDR, DADDR and PROTOC. The SPI numbers SHOULD be ignored when considering the flow over which M-PDM information is measured.

3.4.1 Using ESP Transport Mode

Note that Destination Options may be placed before or after ESP or both. If using M-PDM in ESP transport mode, M-PDM MUST be placed after the ESP header so as not to leak information.

3.4.2 Using ESP Tunnel Mode

Note that Destination Options may be placed before or after ESP or both in both the outer set of IP headers and the inner set of IP headers. A tunnel endpoint that creates a new packet may decide to use M-PDM independent of the use of M-PDM of the original packet to enable delay measurements between the two tunnel endpoints.

3.5 Implementation Considerations

3.5.1 M-PDM Activation

An implementation should provide an interface to enable or disable the use of M-PDM. This specification recommends having M-PDM off by default.

M-PDM MUST NOT be turned on merely if a packet is received with an M-PDM header. The received packet could be spoofed by another device.

3.5.2 M-PDM Timestamps

The M-PDM timestamps are intended to isolate wire time from server or host time, but may necessarily attribute some host processing time to network latency.

RFC2330 [RFC2330] "Framework for IP Performance Metrics" describes two notions of wire time in section 10.2. These notions are only defined in terms of an Internet host H observing an Internet link L at a particular location:

+ For a given IP packet P, the 'wire arrival time' of P at H on L is the first time T at which any bit of P has appeared at H's observational position on L.

+ For a given IP packet P, the 'wire exit time' of P at H on L is the first time T at which all the bits of P have appeared at H's observational position on L.

This specification does not define the exact H's observing position on L. That is left for the deployment setups to define. However, the position where PDM timestamps are taken SHOULD be as close to the physical network interface as possible. Not all implementations will be able to achieve the ideal level of measurement.

3.6 Dynamic Configuration Options

If the M-PDM destination options extension header is used, then it MAY be turned on for all packets flowing through the host, applied to an upper-layer protocol (TCP, UDP, SCTP, etc), a local port, or IP address only. These are at the discretion of the implementation.

3.7 Information Access and Storage

Measurement information provided by M-PDM may be made accessible for higher layers or the user itself. Similar to activating the use of M-PDM, the implementation may also provide an interface to indicate if received.

M-PDM information may be stored, if desired. If a packet with M-PDM information is received and the information should be stored, the upper layers may be notified. Furthermore, the implementation should define a configurable maximum lifetime after which the information can be removed as well as a configurable maximum amount of memory that should be allocated for PDM information.

4 M-PDM HBH Option

The M-PDM Hop-by-Hop option is encoded in type-length-value (TLV) format. It has an alignment requirement of $4n + 2$. (See [IPv6, Section 4.2] for discussion of option alignment.) The option has 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																																						
Option Type										Option Length										Version										LastM										M										MType																			
Middlebox Identifier																				Timestamp In																																																	
Timestamp Out																				PSN This Packet																																																	
Middlebox Identifier																				Timestamp In																																																	
Timestamp Out																				PSN This Packet																																																	
Middlebox Identifier																				Timestamp In																																																	
Timestamp Out																				PSN This Packet																																																	
Middlebox Identifier																				Timestamp In																																																	
Timestamp Out																				PSN This Packet																																																	
Middlebox Identifier																				Timestamp In																																																	
Timestamp Out																				PSN This Packet																																																	
Middlebox Identifier																				Timestamp In																																																	
Timestamp Out																				PSN This Packet																																																	

Option Type

TBD = 0xXX (TBD) [To be assigned by IANA] [RFC2780]

In keeping with RFC 8200 [RFC8200], the two high-order bits of the Option Type field are encoded to indicate specific processing of the option; for the M-PDM HBH option, these two bits MUST be set to 00.

The third high-order bit of the Option Type field specifies whether or not the Option Data of that option can change en route to the packet's final destination.

In M-PDM, the value of the third high-order bit MUST be 0.

Option Length

8-bit unsigned integer. Length of the option, in octets, excluding the Option Type and Option Length fields. This field **MUST** be set to 10.

Version

4-bit unsigned integer.

Last Middlebox

4-bit unsigned integer. Indicates which middlebox number was last done. For example, 3 would indicate that this is the third middlebox. This field could be used to quickly find which set of data to fill. If there have been more than 5 middleboxes, then wrapping will happen and fields will get overwritten.

Marking

2-bit unsigned integer.

Marking Type (M-Type)

4-bit unsigned integer. This indicates the type of marking method being used for the timestamp.

If marking is not used, then the timestamp will be when the packet left the IP interface on this middlebox.

If marking method is used, then this field will contain:

1 - the timestamp of the first packet of a marked batch
2 - the average timestamp of the packets of a batch
3 - a double-marked packet

RSVD

2-bit unsigned integer

Middle Box Identifier

16-bit unsigned integer.

This field MUST be zero if not used. The zeros are intended to make it harder to leak data via the HBH header.

This could be some portion of the IPv4 or IPv6 address or the router ID. [Note to readers: any suggestions for this field are most welcome!]

Timestamp In

16-bit unsigned integer. This can be the timestamp of the packet received by the IP interface on this middlebox. If marking method is used, it can identify the timestamp of the first packet of a marked batch or the average timestamp of the packets of a batch or a double-marked packet, depending on which method is used to perform delay measurements.

This field is initialized to 0.

This timestamp is the delta in nanoseconds from the initial starting timestamp of January 1, 2019 00:00:00.0000000000.

See the section on HBH Timestamps for more on this measurement.

Timestamp Out

16-bit unsigned integer. This can be the timestamp of the packet left the IP interface on this middlebox. If marking method is used, it can identify the timestamp of the first packet of a marked batch or the average timestamp of the packets of a batch or a double-marked packet, depending on which method is used to perform delay measurements.

This field is initialized to 0.

This timestamp is the delta in nanoseconds from the initial starting timestamp of January 1, 2019 00:00:00.0000000000.

See the section on HBH Timestamps for more on this measurement.

Packet Sequence Number This Packet (PSNTP)

16-bit unsigned integer. This field will wrap. It is intended for use while analyzing packet traces.

This field is initialized at a random number and incremented monotonically for each packet of the session flow of the IP stack.

The random-number initialization is intended to make it harder to spoof and insert such packets.

4.1 HBH Timestamps In and Out

The timestamp fields will contain the 16 high-order or most significant bits of the delta between a fixed starting value of January 1, 2019 00:00:00.0000000000 and the current time at the middlebox.

For more on truncation of timestamp values, please see [TCPM].

5 Security Considerations

M-PDM may introduce some new security weaknesses.

5.1 Resource Consumption and Resource Consumption Attacks

M-PDM needs to calculate the deltas for time and keep track of the sequence numbers. This means that control blocks which reside in memory may be kept at the end hosts per 5-tuple.

A limit on how much memory is being used SHOULD be implemented. Without a memory limit, any time a control block is kept in memory, an attacker can try to misuse the control blocks to cause excessive resource consumption. This could be used to compromise the end host.

M-PDM as a Destination is used at the end hosts and memory is used only at the end host M-PDM as an HBH header is used at routers or middle boxes.

5.2 Pervasive monitoring

Since M-PDM passes in the clear, a concern arises as to whether the data can be used to fingerprint the system or somehow obtain information about the contents of the payload.

Let us discuss fingerprinting of the end host first. It is possible that seeing the pattern of deltas or the absolute values could give some information as to the speed of the end host - that is, if it is a very fast system or an older, slow device. This may be useful to the attacker. However, if the attacker has access to PDM, the attacker also has access to the entire packet and could make such a deduction based merely on the time frames elapsed between packets WITHOUT PDM.

As far as deducing the content of the payload, in terms of the

application level information such as web page, user name, user password and so on, it appears to us that PDM is quite unhelpful in this regard. Having said that, the ability to separate wire-time from processing time may potentially provide an attacker with additional information. It is conceivable that an attacker could attempt to deduce the type of application in use by noting the server time and payload length. Some encryption algorithms attempt to obfuscate the packet length to avoid just such vulnerabilities. In the future, encryption algorithms may wish to obfuscate the server time as well.

5.3 M-PDM as a Covert Channel

PDM provides a set of fields in the packet which could be used to leak data. But, there is no real reason to suspect that PDM would be chosen rather than another part of the payload or another Extension Header.

A firewall or another device could sanity check the fields within the PDM but randomly assigned sequence numbers and delta times might be expected to vary widely. The biggest problem though is how an attacker would get access to PDM in the first place to leak data. The attacker would have to either compromise the end host or have Man in the Middle (MitM). It is possible that either one could change the fields. But, then the other end host would get sequence numbers and deltas that don't make any sense.

It is conceivable that someone could compromise an end host and make it start sending packets with PDM without the knowledge of the host. But, again, the bigger problem is the compromise of the end host. Once that is done, the attacker probably has better ways to leak data.

Having said that, if a PDM aware middle box or an implementation (destination host) detects some number of "nonsensical" sequence numbers or timing information, it could take action to block, discard, or alert on this traffic.

5.4 Timing Attacks

The fact that PDM can help in the separation of node processing time from network latency brings value to performance monitoring. Yet, it is this very characteristic of PDM which may be misused to make certain new type of timing attacks against protocols and implementations possible.

Depending on the nature of the cryptographic protocol used, it may be possible to leak the credentials of the device. For example, if an

attacker can see that PDM is being used, then the attacker might use PDM to launch a timing attack against the keying material used by the cryptographic protocol.

An implementation may want to be sure that PDM is enabled only for certain ip addresses, or only for some ports. Additionally, the implementation SHOULD require an explicit restart of monitoring after a certain time period (for example for 1 hour), to make sure that PDM is not accidentally left on after debugging has been done etc.

Even so, if using PDM, a user "Consent to be Measured" SHOULD be a pre-requisite for using PDM. Consent is common in enterprises and with some subscription services. The actual content of "Consent to be Measured" will differ by site but it SHOULD make clear that the traffic is being measured for quality of service and to assist in diagnostics as well as to make clear that there may be potential risks of certain vulnerabilities if the traffic is captured during a diagnostic session.

6 IANA Considerations

This draft requests an Destination Option Type assignment with the act bits set to 00 and the chg bit set to 0 from the Destination Options and Hop-by-Hop Options sub-registry of Internet Protocol Version 6 (IPv6) Parameters [ref to RFCs and URL below.

<http://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml#ipv6-parameters-2>

Hex Value	Binary Value act chg rest	Description	Reference
TBD	TBD	Performance and Diagnostic Metrics (M-PDM)	[This draft]

7 References

7.1 Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2780] Bradner, S. and V. Paxson, "IANA Allocation Guidelines For Values In the Internet Protocol and Related Headers", BCP 37, RFC 2780, March 2000.

[RFC4303] Kent, S, "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.

[RFC8250] Elkins, N., Ackermann, M. and Hamilton, R. "IPv6 Performance and Diagnostic Metrics (PDM) Destination Option", RFC 8250, September 2017.

7.2 Informative References

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

[RFC8321] Fioccola, G. et al, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, January 2018.

[TCPM] Scheffenegger, R., Kuehlewind, M., and B. Trammell, "Encoding of Time Intervals for the TCP Timestamp Option", Work in Progress, draft-trammell-tcpm-timestamp-interval-01, July 2013

Acknowledgments

The authors would like to C.M. Heard for his review.

Authors' Addresses

Nalini Elkins
Inside Products, Inc.
36A Upper Circle
Carmel Valley, CA 93924
United States
Phone: +1 831 659 8360
Email: nalini.elkins@insidethestack.com
<http://www.insidethestack.com>

Giuseppe Fioccola
Telecom Italia
Via Reiss Romoli, 274
Torino 10148
Italy
Email: giuseppe.fioccola@telecomitalia.it

Michael S. Ackermann
Blue Cross Blue Shield of Michigan
P.O. Box 2888
Detroit, Michigan 48231
United States
Phone: +1 310 460 4080
Email: mackermann@bcbsm.com

Robert M. Hamilton
Chemical Abstracts Service
A Division of the American Chemical Society
2540 Olentangy River Road
Columbus, Ohio 43202
United States of America
Phone: +1 614 447 3600 x2517
Email: rhamilton@cas.org

IPPM Working Group
Internet-Draft
Intended status: Experimental
Expires: December 31, 2018

G. Fioccola, Ed.
M. Cociglio
Telecom Italia
A. Sapio
R. Sisto
Politecnico di Torino
June 29, 2018

Multipoint Alternate Marking method for passive and hybrid performance
monitoring
draft-fioccola-ippm-multipoint-alt-mark-04

Abstract

The Alternate Marking method, as presented in RFC 8321 [RFC8321], can be applied only to point-to-point flows because it assumes that all the packets of the flow measured on one node are measured again by a single second node. This document aims to generalize and expand this methodology to measure any kind of unicast flows, whose packets can follow several different paths in the network, in wider terms a multipoint-to-multipoint network. For this reason the technique here described is called Multipoint Alternate Marking. Some definitions here introduced extend the scope of RFC 5644 [RFC5644] in the context of alternate marking schema.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 31, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Correlation with RFC5644	4
3. Flow classification	4
4. Multipoint Performance Measurement	6
4.1. Monitoring Network	7
5. Multipoint Packet Loss	8
6. Network Clustering	9
6.1. Algorithm for Cluster partition	10
7. Timing Aspects	12
8. Multipoint Delay and Delay Variation	14
8.1. Delay measurements on multipoint paths basis	14
8.1.1. Single Marking measurement	14
8.2. Delay measurements on single packets basis	14
8.2.1. Single and Double Marking measurement	14
8.2.2. Hashing selection method	15
9. An SDN enabled Performance Management	17
10. Examples of application	17
11. Security Considerations	18
12. Acknowledgements	18
13. IANA Considerations	18
14. References	18
14.1. Normative References	18
14.2. Informative References	18
Authors' Addresses	19

1. Introduction

The alternate marking method, as presented until now, is applicable to a point-to-point path; so the extension proposed in this document explains the most general case of multipoint-to-multipoint path and

enables flexible and adaptive performance measurements in a managed network.

The Alternate Marking methodology described in RFC 8321 [RFC8321] has the property to synchronize measurements in different points maintaining the coherence of the counters. So it is possible to show what is happening in every marking period for each monitored flow. The monitoring parameters are the packet counter and timestamps of a flow for each marking period.

There are some applications of the alternate marking method where there are a lot of monitored flows and nodes. Multipoint Alternate Marking aims to reduce these values and makes the performance monitoring more flexible in case a detailed analysis is not needed. For instance, by considering n measurement points and m monitored flows, the order of magnitude of the packet counters for each time interval is $n*m*2$ (1 per color). If both n and m are high values the packet counters increase a lot and Multipoint Alternate Marking offers a tool to control these parameters.

The approach presented in this document is applied only to unicast flows and not to multicast. BUM (Broadcast Unknown Unicast Multicast) traffic is not considered here, because traffic replication is not covered by the Multipoint Alternate Marking method.

Alternate Marking method works by definition for multipoint to multipoint paths but the network clustering approach presented in this document is the formalization of how to implement this property and it allows a flexible and optimized performance measurement support.

Without network clustering, it is possible to apply alternate marking only for all the network or per single flow. Instead, with network clustering, it is possible to use the network clusters partition at different levels to perform the needed degree of detail. In some circumstances it is possible to monitor a Multipoint Network by analyzing the Network Clustering, without examining in depth. In case of problems (packet loss is measured or the delay is too high) the filtering criteria could be specified more in order to perform a detailed analysis by using a different combination of clusters up to a per-flow measurement as described in RFC 8321 [RFC8321].

An application could be the Software Defined Network (SDN) paradigm where the SDN Controllers are the brains of the network and can manage flow control to the switches and routers and, in the same way, can calibrate the performance measurements depending on the necessity. An SDN Controller Application can orchestrate how deep the network performance monitoring is setup.

2. Correlation with RFC5644

RFC 5644 [RFC5644] is limited to active measurements using a single source packet or stream, and observations of corresponding packets along the path (spatial), at one or more destinations (one-to-group), or both. Instead, the scope of this memo is to define multiparty metrics for passive and hybrid measurements in a group-to-group topology with multiple sources and destinations.

RFC 5644 [RFC5644] introduces metric names that can be reused also here but have to be extended and rephrased to be applied to the alternate marking schema:

- a. the multiparty metrics are not only one-to-group metrics but can be also group-to-group metrics;
- b. the spatial metrics, used for measuring the performance of segments of a source to destination path, are applied here to group-to-group segments (called Clusters).

3. Flow classification

An unicast flow is identified by all the packets having a set of common characteristics. This definition is inspired by RFC 7011 [RFC7011].

As an example, by considering a flow as all the packets sharing the same source IP address or the same destination IP address, it is easy to understand that the resulting pattern will not be a point-to-point connection, but a point-to-multipoint or multipoint-to-point connection.

In general a flow can be defined by a set of selection rules used to match a subset of the packets processed by the network device. These rules specify a set of headers fields (Identification Fields) and the relative values that must be found in matching packets.

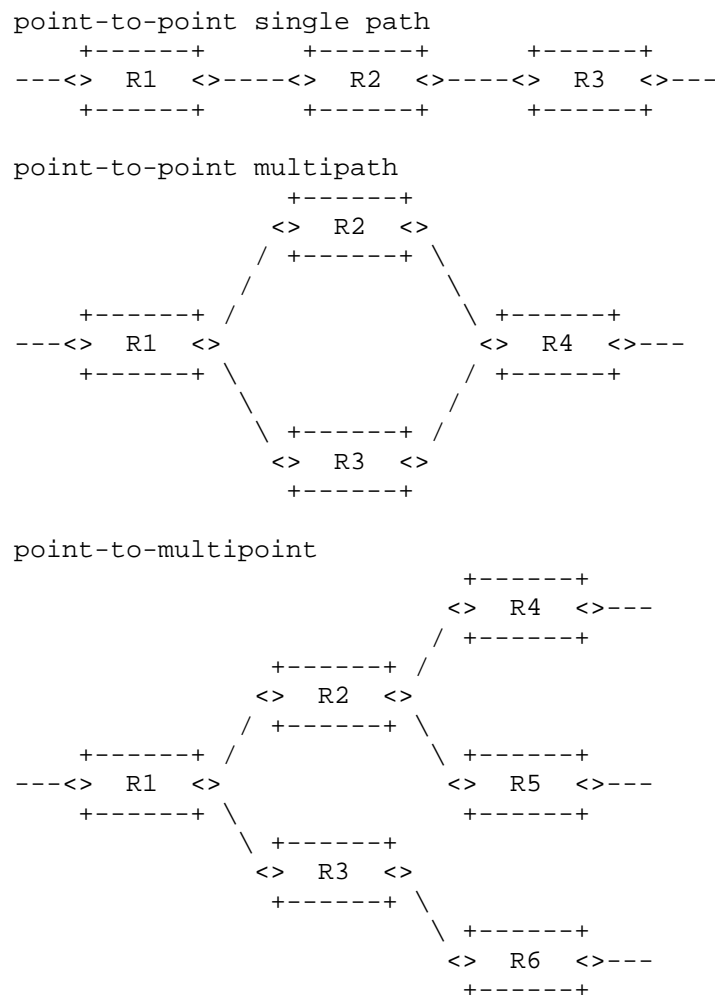
The choice of the identification fields directly affects the type of paths that the flow would follow in the network. In fact, it is possible to relate a set of identification fields with the pattern of the resulting graphs, as listed in Figure 1.

A TCP 5-tuple usually identifies flows following either a single path or a point-to-point multipath (in case of load balancing). On the contrary, a single source address selects flows following a point-to-multipoint, while a multipoint-to-point can be the result of a matching on a single destination address. In case a selection rule and its reverse are used for bidirectional measurements, they can

correspond to a point-to-multipoint in one direction and a multipoint-to-point in the opposite direction.

In this way the flows to be monitored are selected into the monitoring points using packet selection rules, that can also change the pattern of the monitored network.

The alternate marking method is applicable only to a single path (and partially to a one-to-one multipath), so the extension proposed in this document is suitable also for the most general case of multipoint-to-multipoint, which embraces all the other patterns of Figure 1.



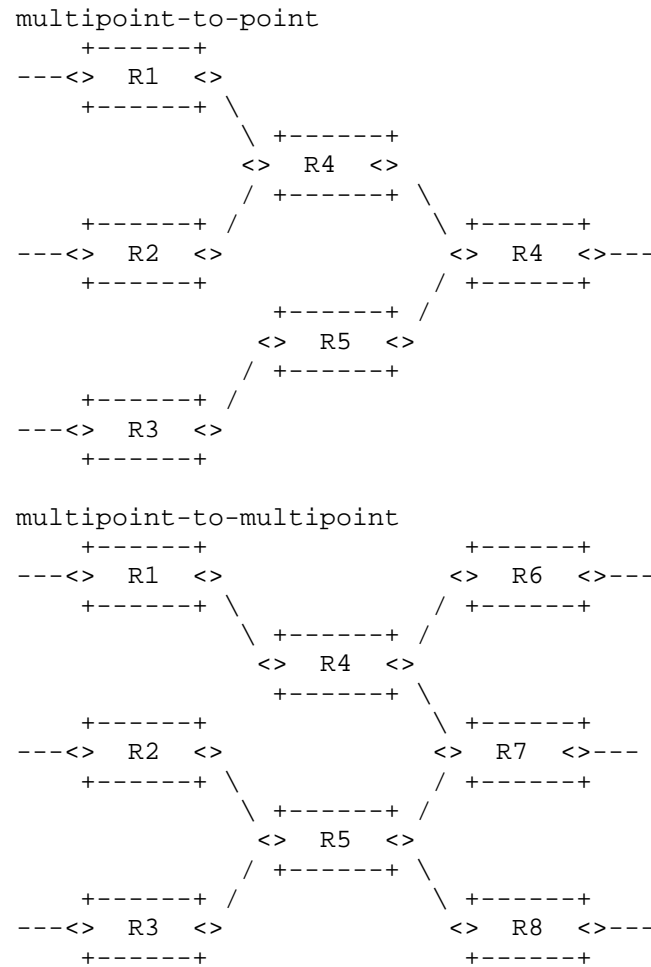


Figure 1: Flow classification

4. Multipoint Performance Measurement

By Using the "traditional" alternate marking method only point-to-point paths can be monitored. To have an IP (TCP/UDP) flow that follows a point-to-point path we have to define, with a specific value, 5 identification fields (IP Source, IP Destination, Transport Protocol, Source Port, Destination Port).

Multipoint Alternate Marking enables the performance measurement for multipoint flows selected by identification fields without any

constraints (even the entire network production traffic). It is also possible to use multiple marking points for the same monitored flow.

4.1. Monitoring Network

The Monitoring Network is deduced from the Production Network, by identifying the nodes of the graph that are the measurement points, and the links that are the connections between measurement points.

There are some techniques that can help with the building of the monitoring network (as an example it is possible to mention [I-D.amf-ippm-route]). In general there are different options: the monitoring network can be obtained by considering all the possible paths for the traffic or also by checking the traffic sometimes and update the graph consequently.

So a graph model of the monitoring network can be built according to the alternate marking method: the monitored interfaces and links are identified. Only the measurement points and links where the traffic has flowed have to be represented in the graph.

The following figure shows a simple example of a Monitoring Network graph:

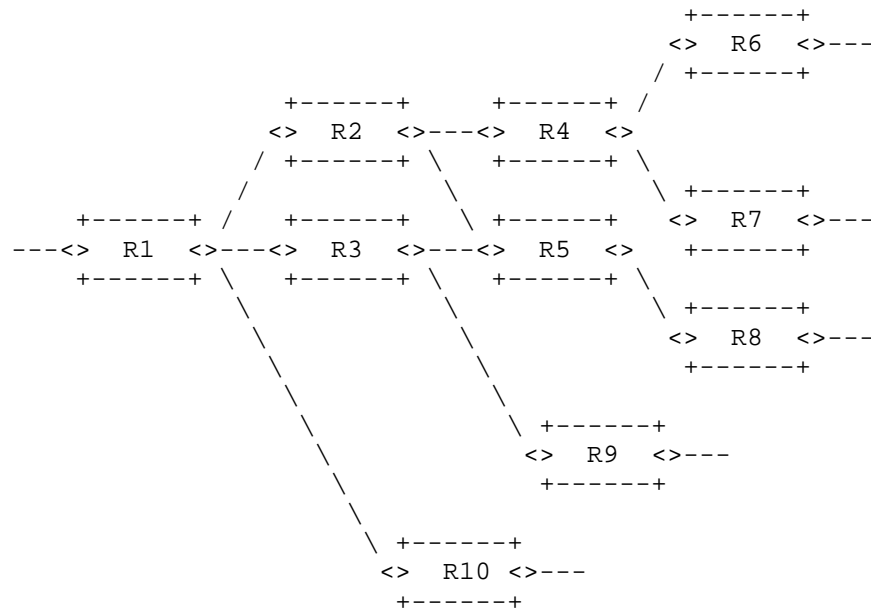


Figure 2: Monitoring Network Graph

Each monitoring point is characterized by the packet counter that refers only to a marking period of the monitored flow.

The same is applicable also for the delay but it will be described in the following sections.

5. Multipoint Packet Loss

Since all the packets of the considered flow leaving the network have previously entered the network, the number of packets counted by all the input nodes is always greater or equal than the number of packets counted by all the output nodes.

And in case of no packet loss occurring in the marking period, if all the input and output points of the network domain to be monitored are measurement points, the sum of the number of packets on all the ingress interfaces and on all the egress interfaces is the same. In this circumstance, if no packet loss occurs, the intermediate measurement points have only the task to split the measurement.

It is possible to define the Network Packet Loss (for 1 flow, for 1 period): <<In a packet network, the number of lost packets is the

number of packets counted by the input nodes minus the number of packets counted by the output nodes>>. This is true for every packet flow in each marking period.

The Monitored Network Packet Loss with n input nodes and m output nodes is given by:

$$PL = (PI_1 + PI_2 + \dots + PI_n) - (PO_1 + PO_2 + \dots + PO_m)$$

where:

PL is the Network Packet Loss (number of lost packets)

PI_i is the Number of packets flowed through the i -th Input node in this period

PO_j is the Number of packets flowed through the j -th Output node in this period

The equation is applied on a per-time-interval basis.

6. Network Clustering

The previous Equation can determine the number of packets lost globally in the monitored network, exploiting only the data provided by the counters in the input and output nodes.

In addition it is also possible to leverage the data provided by the other counters in the network to converge on the smallest identifiable subnetworks where the losses occur. These subnetworks are named Clusters.

A Cluster graph is a subnetwork of the entire Monitoring Network graph that still satisfies the packet loss equation where PL in this case is the number of packets lost in the Cluster.

For this reason a Cluster should contain all the arcs emanating from its input nodes and all the arcs terminating at its output nodes. This ensures that we can count all the packets (and only those) exiting an input node again at the output node, whatever path they follow.

In a completely monitored network (a network where every network interface is monitored), each network device corresponds to a Cluster and each physical link corresponds to two Clusters (one for each direction).

Clusters can have different sizes depending on flow filtering criteria adopted.

Moreover, sometimes Clusters can be optionally simplified. For example when two monitored interfaces are divided by a single router (one is the input interface and the other is the output interface and the router has only these two interfaces), instead of counting exactly twice, upon entering and leaving, it is possible to consider a single measurement point (in this case we do not care of the internal packet loss of the router).

6.1. Algorithm for Cluster partition

A simple algorithm can be applied in order to split our monitoring network into Clusters. It is a two-step algorithm:

- o Group the links where there is the same starting node;
- o Join the grouped links with at least one ending node in common.

In our monitoring network graph example it is possible to identify the Clusters partition by applying this two-step algorithm.

The first step identifies the following groups:

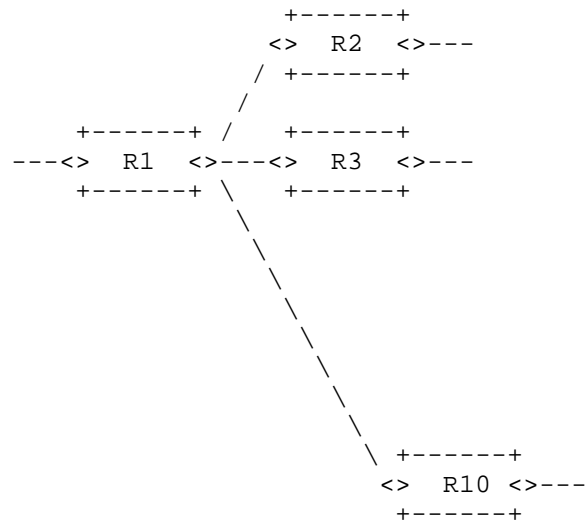
1. Group 1: (R1-R2), (R1-R3), (R1-R10)
2. Group 2: (R2-R4), (R2-R5)
3. Group 3: (R3-R5), (R3-R9)
4. Group 4: (R4-R6), (R4-R7)
5. Group 5: (R5-R8)

And then, the second step builds the Clusters partition (in particular we can underline that Group 2 and Group 3 connect together, since R5 is in common):

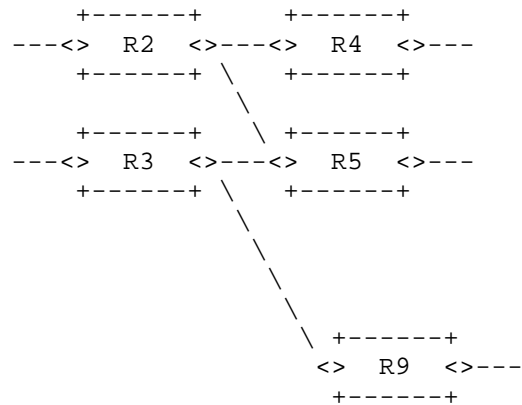
1. Cluster 1: (R1-R2), (R1-R3), (R1-R10)
2. Cluster 2: (R2-R4), (R2-R5), (R3-R5), (R3-R9)
3. Cluster 3: (R4-R6), (R4-R7)
4. Cluster 4: (R5-R8)

In the end the following 4 Clusters are obtained:

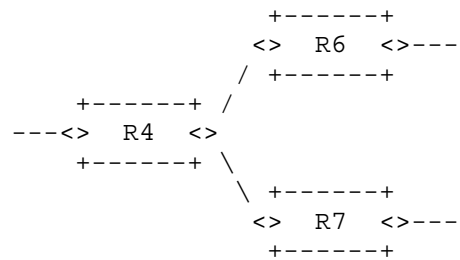
Cluster 1



Cluster 2



Cluster 3



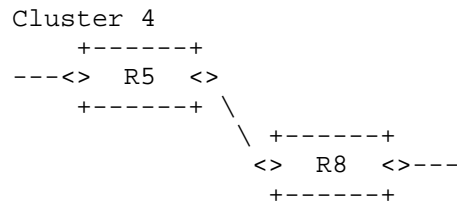


Figure 3: Clusters example

There are Clusters with more than 2 nodes and two-nodes Clusters. In the two-nodes Clusters the loss is on the link (Cluster 4). In more-than-2-nodes Clusters the loss is on the Cluster but we cannot know in which link (Cluster 1, 2, 3).

In this way the calculation of packet loss can be made on Cluster basis. Note that CIR(Committed Information Rate) and EIR(Excess Information Rate) can also be deduced on Cluster basis.

Obviously, by combining some Clusters in a new connected subnetwork (called Super Cluster) the Packet Loss Rule is still true.

In this way in a very large network there is no need to configure detailed filter criteria to inspect the traffic. You can check multipoint network and only in case of problems you can go deep with a step-by-step cluster analysis, but only for the cluster or combination of clusters where the problem happens.

7. Timing Aspects

The mark switching approach based on a fixed timer is considered in this document.

So, if we analyze a multipoint-to-multipoint path with more than one marking node, it is important to recognize the reference measurement interval. In general the measurement interval for describing the results is the interval of the marking node that is more aligned with the start of the measurement, as reported in the following figure.

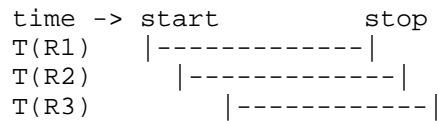


Figure 4: Measurement Interval

T(R1) is the measurement interval and this is essential in order to be compatible and make comparison with other active/passive/hybrid Packet Loss metrics.

That is why, when we expand to multipoint-to-multipoint flows, we have to consider that all source nodes mark the traffic.

Regarding the timing aspects of the methodology, RFC 8321 [RFC8321] already describes two contributions that are taken into account: the clock error between network devices and the network delay between measurement points.

But we should now consider an additional contribution. Since all source nodes mark the traffic, the source measurement intervals can be of different lengths and with different offsets and this mismatch m can be added to d , as shown in figure.

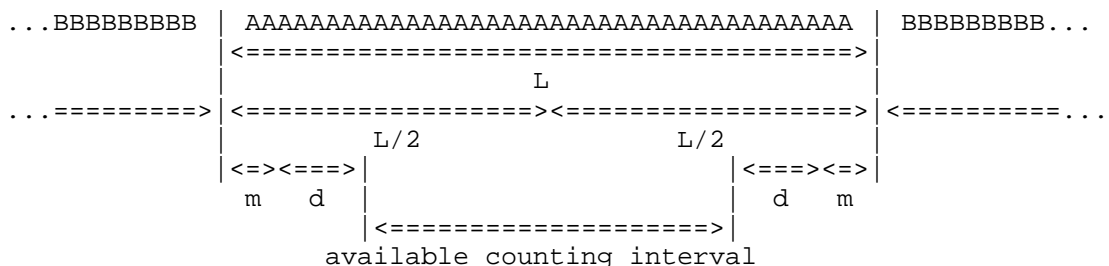


Figure 5: Timing Aspects for Multipoint paths

So the misalignment between the marking source routers gives an additional constraint and the value of m is added to d (that already includes clock error and network delay).

In the end, the condition that must be satisfied to enable the method to function properly is that the available counting interval must be > 0 , and that means: $L - 2m - 2d > 0$ for each measurement point on the multipoint path. Therefore, the mismatch between measurement intervals must satisfy this condition.

8. Multipoint Delay and Delay Variation

The same line of reasoning can be applied to Delay and Delay Variation. It is important to highlight that both delay and delay variation measurements make sense in a multipoint path. The Delay Variation is calculated by considering the same packets selected for measuring the Delay.

In general, it is possible to perform delay and delay variation measurements on multipoint paths basis or on single packets basis:

- o Delay measurements on multipoint paths basis means that the delay value is representative of an entire multipoint path (e.g. whole multipoint network, a cluster or a combination of clusters).
- o Delay measurements on single packets basis means that you can use multipoint path just to easily couple packets between inputs and output nodes of a multipoint path, as it is described in the following sections.

8.1. Delay measurements on multipoint paths basis

8.1.1. Single Marking measurement

Mean delay and mean delay variation measurements can also be generalized to the case of multipoint flows. It is possible to compute the average one-way delay of packets, in one block, in a cluster or in the entire monitored network.

The average latency can be measured as the difference between the weighted averages of the mean timestamps of the sets of output and input nodes.

8.2. Delay measurements on single packets basis

8.2.1. Single and Double Marking measurement

Delay and delay variation measurements relative to only one picked packet per period (both single and double marked) can be performed in the Multipoint scenario with some limitations:

Single marking based on the first/last packet of the interval would not work, because it would not be possible to agree on the first packet of the interval.

Double marking or multiplexed marking would work, but each measurement would only give information about the delay of a single path. However, by repeating the measurement multiple

times, it is possible to get information about all the paths in the multipoint flow. This can be done in case of point-to-multipoint path but it is more difficult to achieve in case of multipoint-to-multipoint path because of the multiple source routers.

if we would perform a delay measurement for more than one picked packet in the same marking period and, especially, if we want to get delay measurements on multipoint-to-multipoint basis, both single and double marking method are not useful in the Multipoint scenario, since they would not be representative of the entire flow. The packets can follow different paths with various delays and in general it can be very difficult to recognize marked packets in a multipoint-to-multipoint path especially in case they are more than one per period.

A desirable option is to monitor simultaneously all the paths of a multipoint path in the same marking period and, for this purpose, hashing can be used as reported in the next Section.

8.2.2. Hashing selection method

RFC 5474 [RFC5474] and RFC 5475 [RFC5475] introduce sampling and filtering techniques for IP Packet Selection.

The hash-based selection methodologies for delay measurement can work in a multipoint-to-multipoint path and can be used both coupled to mean delay or stand alone.

[I-D.mizrahi-ippm-compact-alternate-marking] introduces how to use the Hash method combined with alternate marking method for point-to-point flows. It is also called Mixed Hashed Marking: the coupling of marking method and hashing technique is very useful because the marking batches anchor the samples selected with hashing and this simplifies the correlation of the hashing packets along the path.

It is possible to use a basic hash or a dynamic hash method. One of the challenges of the basic approach is that the frequency of the sampled packets may vary considerably. For this reason the dynamic approach has been introduced for point-to-point flow in order to have the desired and almost fixed number of samples for each measurement period. In the hash-based sampling, alternate marking is used to create periods, so that hash-based samples are divided into batches, allowing to anchor the selected samples to their period. Moreover in the dynamic hash-based sampling, by dynamically adapting the length of the hash value, the number of samples is bounded in each marking period. This can be realized by choosing the maximum number of samples (NMAX) to be caught in a marking period. The algorithm

starts with only few hash bits, that permit to select a greater percentage of packets (e.g. with 0 bit of hash all the packets are sampled, with 1 bit of hash half of the packets are sampled, and so on). When the number of selected packets reaches NMAX, a hashing bit is added. As a consequence, the sampling proceeds at half of the original rate and also the packets already selected that don't match the new hash are discarded. This step can be repeated iteratively. It is assumed that each sample includes the timestamp (used for delay measurement) and the hash value, allowing the management system to match the samples received from the two measurement points. The dynamic process statistically converges at the end of a marking period and the final number of selected samples is between $NMAX/2$ and NMAX. Therefore, the dynamic approach paces the sampling rate, allowing to bound the number of sampled packets per sampling period.

In a multipoint environment the behaviour is similar to point-to-point flow. In particular, in the context of multipoint-to-multipoint flow, the dynamic hash could be the solution to perform delay measurements on specific packets and to overcome the single and double marking limitations.

The management system receives the samples including the timestamps and the hash value from all the MPs, and this happens both for point-to-point and for multipoint-to-multipoint flow. Then the longest hash used by MPs is deduced and it is applied to couple timestamps of same packets of 2 MPs of a point-to-point path or of input and output MPs of a Cluster (or a Super Cluster or the entire network). But some considerations are needed: if there isn't packet loss the set of input samples is always equal to the set of output samples. In case of packet loss the set of output samples can be a subset of input samples but the method still works because, at the end, it is easy to couple the input and output timestamps of each caught packet using the hash (in particular the "unused part of the hash" that should be different for each packet).

In summary, the basic hash is logically similar to the double marking method, and in case of point-to-point path double marking and basic hash selection are equivalent. The dynamic approach scales the number of measurements per interval, and it would seem that double marking would also work well if we reduced the interval length, but this can be done only for point-to-point path and not for multipoint path, where we cannot couple the picked packets in a multipoint paths. So, in general, if we want to get delay measurements on multipoint-to-multipoint path basis and want to select more than one packet per period, double marking cannot be used because we could not be able to couple the picked packets between input and output nodes. On the other hand we can do that by using hashing selection.

9. An SDN enabled Performance Management

The Multipoint Alternate Marking framework that is introduced in this document adds flexibility to PM because it can reduce the order of magnitude of the packet counters. This allows an SDN Orchestrator to supervise, control and manage PM in large networks.

The monitoring network can be considered as a whole or can be split in Clusters, that are the smallest subnetworks (group-to-group segments), maintaining the packet loss property for each subnetwork. They can also be combined in new connected subnetworks at different levels depending on the detail we want to achieve.

An SDN Controller can calibrate Performance Measurements. It can start without examining in depth. In case of necessity (packet loss is measured or the delay is too high), the filtering criteria could be immediately specified more in order to perform a partition of the network by using Clusters and/or different combinations of Clusters. In this way the problem can be localized in a specific Cluster or in a single combination of Clusters and a more detailed analysis can be performed step-by-step by successive approximation up to a point-to-point flow detailed analysis.

In addition an SDN Controller could also collect the measurement history.

10. Examples of application

There are three application fields where it may be useful to take into consideration the Multipoint Alternate Marking:

- o VPN: The IP traffic is selected on IP source basis in both directions. At the end point WAN interface all the output traffic is counted in a single flow. The input traffic is composed by all the other flows aggregated for source address. So, by considering n end-points, the monitored flows are n (each flow with 1 ingress point and $(n-1)$ egress points) instead of $n*(n-1)$ flows (each flow, with 1 ingress point and 1 egress point);
- o Mobile Backhaul: LTE traffic is selected, in the Up direction, by the ENodeB source address and, in Down direction, by the ENodeB destination address because the packets are sent from the Mobile Packet Core to the ENodeB. So the monitored flow is only one per ENodeB in both directions;
- o OTT(Over The Top) services: The traffic is selected, in the Down direction by the source addresses of the packets sent by OTT Servers. In the opposite direction (Up) by the destination IP

addresses of the same Servers. So the monitoring is based on a single flow per OTT Servers in both directions.

11. Security Considerations

This document specifies a method to perform measurements that does not directly affect Internet security nor applications that run on the Internet. However, implementation of this method must be mindful of security and privacy concerns, as explained in RFC 8321 [RFC8321].

12. Acknowledgements

The authors would like to thank Al Morton, Tal Mizrahi, Rachel Huang for the precious contribution.

13. IANA Considerations

tbc

14. References

14.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5644] Stephan, E., Liang, L., and A. Morton, "IP Performance Metrics (IPPM): Spatial and Multicast", RFC 5644, DOI 10.17487/RFC5644, October 2009, <<https://www.rfc-editor.org/info/rfc5644>>.
- [RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321, January 2018, <<https://www.rfc-editor.org/info/rfc8321>>.

14.2. Informative References

- [I-D.amf-ippm-route] Alvarez-Hamelin, J., Morton, A., and J. Fabini, "Advanced Unidirectional Route Assessment", draft-amf-ippm-route-01 (work in progress), October 2017.

- [I-D.mizrahi-ippm-compact-alternate-marking]
Mizrahi, T., Arad, C., Fioccola, G., Cociglio, M., Chen, M., Zheng, L., and G. Mirsky, "Compact Alternate Marking Methods for Passive and Hybrid Performance Monitoring", draft-mizrahi-ippm-compact-alternate-marking-01 (work in progress), March 2018.
- [RFC5474] Duffield, N., Ed., Chiou, D., Claise, B., Greenberg, A., Grossglauser, M., and J. Rexford, "A Framework for Packet Selection and Reporting", RFC 5474, DOI 10.17487/RFC5474, March 2009, <<https://www.rfc-editor.org/info/rfc5474>>.
- [RFC5475] Zseby, T., Molina, M., Duffield, N., Niccolini, S., and F. Raspall, "Sampling and Filtering Techniques for IP Packet Selection", RFC 5475, DOI 10.17487/RFC5475, March 2009, <<https://www.rfc-editor.org/info/rfc5475>>.
- [RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <<https://www.rfc-editor.org/info/rfc7011>>.

Authors' Addresses

Giuseppe Fioccola (editor)
Telecom Italia
Via Reiss Romoli, 274
Torino 10148
Italy

Email: giuseppe.fioccola@telecomitalia.it

Mauro Cociglio
Telecom Italia
Via Reiss Romoli, 274
Torino 10148
Italy

Email: mauro.cociglio@telecomitalia.it

Amedeo Sapia
Politecnico di Torino
Corso Duca degli Abruzzi, 24
Torino 10129
Italy

Email: amedeo.sapia@polito.it

Riccardo Sisto
Politecnico di Torino
Corso Duca degli Abruzzi, 24
Torino 10129
Italy

Email: riccardo.sisto@polito.it

SPRING Working Group
Internet-Draft
Intended Status: Standards Track
Expires: March 18, 2019

R. Gandhi, Ed.
C. Filsfils
Cisco Systems, Inc.
D. Voyer
Bell Canada
S. Salsano
Universita di Roma "Tor Vergata"
P. L. Ventre
CNIT
M. Chen
Huawei
September 14, 2018

UDP Path for In-band
Performance Measurement for Segment Routing Networks
draft-gandhi-spring-udp-pm-02

Abstract

Segment Routing (SR) is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes. This document specifies procedures for using UDP path for sending and processing in-band probe query and response messages for Performance Measurement. The procedure uses the RFC 6374 defined mechanisms for Delay and Loss performance measurement. The procedure specified is applicable to SR-MPLS and SRv6 data planes for both links and end-to-end measurement for SR Policies. This document also defines mechanisms for handling Equal Cost Multipaths (ECMPs) for SR Policies. In addition, this document defines Return Path Segment List TLV for two-way performance measurement and Block Number TLV for loss measurement.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Conventions Used in This Document	4
2.1. Requirements Language	4
2.2. Abbreviations	4
2.3. Reference Topology	5
3. Probe Messages	6
3.1. Probe Query Message	6
3.1.1. Delay Measurement Probe Query Message	6
3.1.2. Loss Measurement Probe Query Message	7
3.1.2.1. Block Number TLV	8
3.1.3. In-band Probe Query for SR Links	8
3.1.4. In-band Probe Query for End-to-end Measurement of SR Policy	8
3.1.4.1. In-band Probe Query Message for SR-MPLS Policy	8
3.1.4.2. In-band Probe Query Message for SRv6 Policy	9
3.2. Probe Response Message	9
3.2.1. One-way Measurement for SR Link and end-to-end SR Policy	10
3.2.1.1. Probe Response Message to Controller	11
3.2.2. Two-way Measurement for SR Links	11
3.2.3. Two-way End-to-end Measurement of SR Policy	11
3.2.3.1. Return Path Segment List TLV	11
3.2.3.2. In-band Probe Response Message for SR-MPLS Policy	13
3.2.3.3. In-band Probe Response Message for SRv6 Policy	13
4. Performance Measurement for P2MP SR Policies	14
5. ECMP Support	14
6. Sequence Number TLV	14
7. Security Considerations	15
8. IANA Considerations	15

9. References	16
9.1. Normative References	16
9.2. Informative References	16
Acknowledgments	19
Contributors	19
Authors' Addresses	19

1. Introduction

Segment Routing (SR) technology greatly simplifies network operations for Software Defined Networks (SDNs). SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes. SR takes advantage of the Equal-Cost Multipaths (ECMPs) between source, transit and destination nodes. SR Policies as defined in [I-D.spring-segment-routing-policy] are used to steer traffic through a specific, user-defined path using a stack of Segments. Built-in SR Performance Measurement (PM) is one of the essential requirements to provide Service Level Agreements (SLAs).

The One-Way Active Measurement Protocol (OWAMP) defined in [RFC4656] and Two-Way Active Measurement Protocol (TWAMP) defined in [RFC5357] provide capabilities for the measurement of various performance metrics in IP networks. These protocols rely on control channel signaling to establish a test channel over an UDP path. These protocols lack support for IEEE 1588 timestamp [IEEE1588] format and direct-mode Loss Measurement (LM), which are required in SR networks [RFC6374]. The Simple Two-way Active Measurement Protocol (STAMP) [I-D.ippm-stamp] alleviates the control channel signaling by using configuration data model to provision test channels. In addition, the STAMP supports IEEE 1588 timestamp format for Delay Measurement (DM). The TWAMP Light from broadband forum [BBF.TR-390] provides simplified mechanisms for active performance measurement in Customer Edge IP networks.

[RFC6374] specifies protocol mechanisms to enable the efficient and accurate measurement of performance metrics and can be used in SR networks with MPLS data plane [I-D.spring-sr-mpls-pm]. [RFC6374] addresses the limitations of the IP based performance measurement protocols as specified in Section 1 of [RFC6374]. The [RFC6374] requires data plane to support MPLS Generic Associated Channel Label (GAL) and Generic Associated Channel (G-Ach), which may not be supported on all nodes in the network.

[RFC7876] specifies the procedures to be used when sending and processing out-of-band performance measurement probe response messages over an UDP return path for RFC 6374 based probe queries.

[RFC7876] can be used to send out-of-band PM probe responses in both SR-MPLS and SRv6 networks for one-way performance measurement.

For SR Policies, there are ECMPs between the source and transit nodes, between transit nodes and between transit and destination nodes. Existing PM protocols (e.g. RFC 6374) do not define handling for ECMP forwarding paths in SR networks.

For two-way measurements for SR Policies, there is a need to specify a return path in the form of a Segment List in PM probe query messages without requiring any SR Policy state on the destination node. Existing protocols do not have such mechanisms to specify return path in the PM probe query messages.

This document specifies a procedure for using UDP path for sending and processing in-band probe query and response messages for Performance Measurement that does not require to bootstrap PM sessions. The procedure uses RFC 6374 defined mechanisms for Delay and Loss PM and unless otherwise specified, the procedures from RFC 6374 are not modified. The procedure specified is applicable to both SR-MPLS and SRv6 data planes. The procedure does not require to bootstrap PM sessions and can be used for both SR links and end-to-end performance measurement for SR Policies. This document also defines mechanisms for handling Equal Cost Multipaths (ECMPs) for SR Policies. In addition, this document defines Return Path Segment List (RPSL) TLV for two-way performance measurement and Block Number TLV for loss measurement.

2. Conventions Used in This Document

2.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.2. Abbreviations

ACH: Associated Channel Header.

BSID: Binding Segment ID.

DFlag: Data Format Flag.

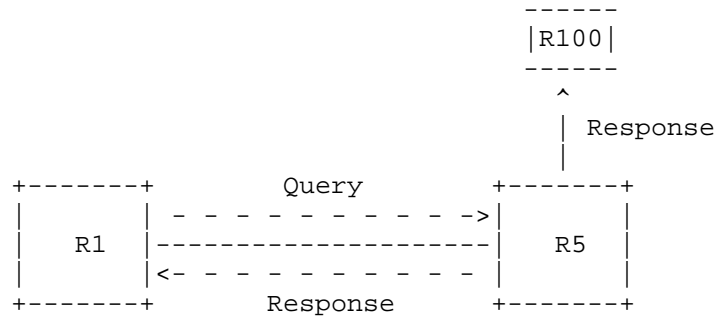
DM: Delay Measurement.

ECMP: Equal Cost Multi-Path.
G-ACh: Generic Associated Channel (G-ACh).
GAL: Generic Associated Channel (G-ACh) Label.
LM: Loss Measurement.
MPLS: Multiprotocol Label Switching.
NTP: Network Time Protocol.
OWAMP: One-Way Active Measurement Protocol.
PM: Performance Measurement.
PTP: Precision Time Protocol.
RPSL: Return Path Segment List.
SID: Segment ID.
SL: Segment List.
SR: Segment Routing.
SR-MPLS: Segment Routing with MPLS data plane.
SRv6: Segment Routing with IPv6 data plane.
STAMP: Simple Two-way Active Measurement Protocol.
TC: Traffic Class.
TWAMP: Two-Way Active Measurement Protocol.
URO: UDP Return Object.

2.3. Reference Topology

In the reference topology, the querier node R1 initiates a probe query for performance measurement and the responder node R5 sends a probe response for the query message received. The probe response may be sent to the querier node R1 or to a controller node R100. The nodes R1 and R5 may be directly connected via a link enabled with Segment Routing or there exists a Point-to-Point (P2P) SR Policy [I-D.spring-segment-routing-policy] on node R1 with destination to

node R5. In case of Point-to-Multipoint (P2MP), SR Policy originating from source node R1 may terminate on multiple destination leaf nodes [I-D.spring-sr-p2mp-policy].



Reference Topology

Both Delay and Loss performance measurement is performed in-band for the traffic traversing between node R1 and node R5. One-way delay and two-way delay measurements are defined in Section 2.4 of [RFC6374]. Transmit and Receive packet loss measurements are defined in Section 2.2 and Section 2.6 of [RFC6374]. One-way loss measurement provides receive packet loss whereas two-way loss measurement provides both transmit and receive packet loss.

3. Probe Messages

3.1. Probe Query Message

In this document, UDP path is defined for sending and processing PM probe query messages for Delay and Loss measurements for SR links and end-to-end SR Policies as described in the following Sections. As well-known UDP port is used for identifying PM probe packets, bootstrapping of the PM session [RFC5357] is not required. The TTL / Hop Limit field of the IP header MUST be set to 1.

3.1.1. Delay Measurement Probe Query Message

The message content for Delay Measurement probe query message using UDP header [RFC768] is shown in Figure 1. As shown, the DM probe query message is sent with Destination UDP port number TBA1 defined in this document. The Source UDP port may optionally be set to TBA1 for two-way delay measurement. The DM probe query message contains the payload for delay measurement defined in Section 3.2 of [RFC6374].

```

+-----+
| IP Header                                     |
.   Source IP Address = Querier IPv4 or IPv6 Address   .
.   Destination IP Address = Responder IPv4 or IPv6 Address .
.   Protocol = UDP                                     .
.   IP TTL = 1                                         .
.   Router Alert Option Not Set                       .
.                                                     .
+-----+
| UDP Header                                     |
.   Source Port = As chosen by Querier                 .
.   Destination Port = TBA1 by IANA for Delay Measurement .
.                                                     .
+-----+
| Payload = Message as specified in Section 3.2 of RFC 6374 |
.                                                     .
+-----+

```

Figure 1: DM Probe Query Message

3.1.2. Loss Measurement Probe Query Message

The message content for Loss measurement probe query message using UDP header [RFC768] is shown in Figure 2. As shown, the LM probe query message is sent with Destination UDP port number TBA2 defined in this document. The Source UDP port may optionally be set to TBA2 for two-way loss measurement. The LM probe query message contains the payload for loss measurement defined in Section 3.1 of [RFC6374].

```

+-----+
| IP Header                                     |
.   Source IP Address = Querier IPv4 or IPv6 Address   .
.   Destination IP Address = Responder IPv4 or IPv6 Address .
.   Protocol = UDP                                     .
.   IP TTL = 1                                         .
.   Router Alert Option Not Set                       .
.                                                     .
+-----+
| UDP Header                                     |
.   Source Port = As chosen by Querier                 .
.   Destination Port = TBA2 by IANA for Loss Measurement .
.                                                     .
+-----+
| Payload = Message as specified in Section 3.1 of RFC 6374 |
.                                                     .
+-----+

```

Figure 2: LM Probe Query Message

The path segment identifier [I-D.spring-mpls-path-segment]
[I-D.pce-sr-path-segment] of the SR Policy is required for accounting
received traffic on the egress node for loss measurement.

3.1.2.1. Block Number TLV

The Loss Measurement using Alternate-Marking method defined in [RFC8321] requires to identify the Block Number (color) of the traffic counters carried by the probe query and response messages. Probe query and response messages specified in [RFC6374] for Loss Measurement do not define any means to carry the Block Number.

[RFC6374] defines probe query and response messages that can include one or more optional TLVs. New TLV Type (value TBA8) is defined in this document to carry Block Number (32-bit) for the traffic counters in the probe query and response messages for loss measurement. The format of the Block Number TLV is shown in Figure 11:

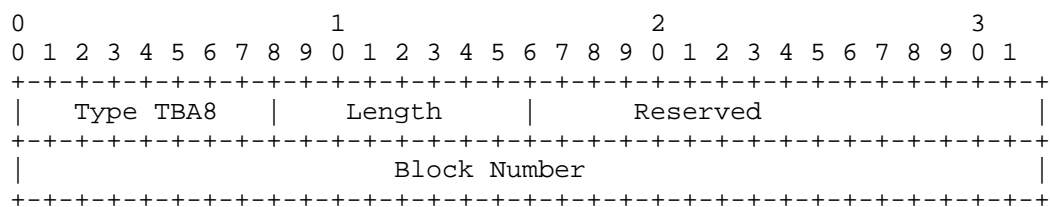


Figure 11: Block Number TLV

The Block Number TLV is optional. The PM querier node SHOULD only insert one Block Number TLV in the probe query message and the responder node in the probe response message SHOULD return the first Block Number TLV from the probe query messages and ignore other Block Number TLVs if present. In both probe query and response messages, the counters MUST belong to the same Block Number.

3.1.3. In-band Probe Query for SR Links

The probe query message as defined in Figure 1 is sent in-band for Delay measurement. The probe query message as defined in Figure 2 is sent in-band for Loss measurement.

3.1.4. In-band Probe Query for End-to-end Measurement of SR Policy

3.1.4.1. In-band Probe Query Message for SR-MPLS Policy

The message content for in-band probe query message using UDP header

for end-to-end performance measurement of SR-MPLS Policy is shown in Figure 3.

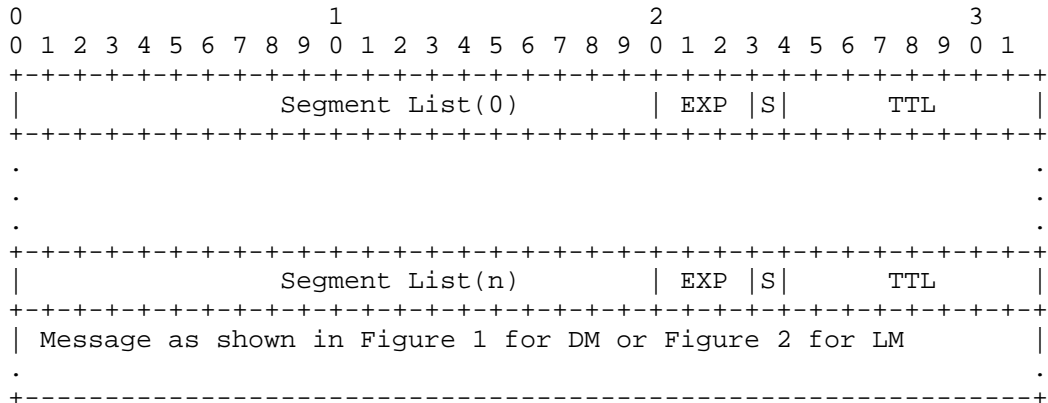


Figure 3: In-band Probe Query Message for SR-MPLS Policy

The Segment List (SL) can be empty to indicate Implicit NULL label case.

3.1.4.2. In-band Probe Query Message for SRv6 Policy

The in-band probe query messages using UDP header for end-to-end performance measurement of an SRv6 Policy is sent using SRv6 Segment Routing Header (SRH) and Segment List of the SRv6 Policy as defined in [I-D.6man-segment-routing-header] and is shown in Figure 4.

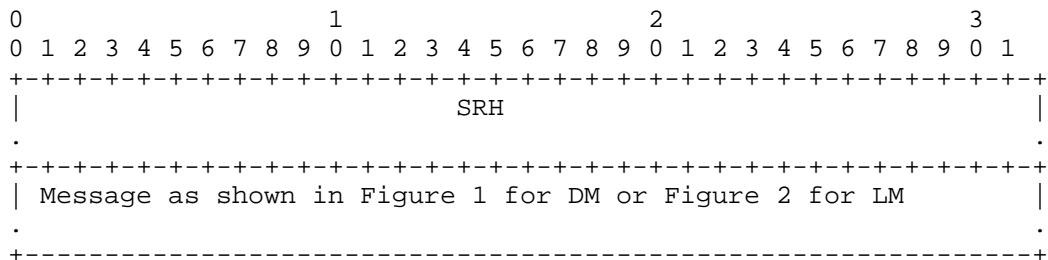


Figure 4: In-band Probe Query Message for SRv6 Policy

3.2. Probe Response Message

When the received probe query message does not contain any UDP Return Object (URO) TLV [RFC7876], the probe response message is sent using the IP/UDP information from the probe query message. The content of

the probe response message is shown in Figure 5.

```

+-----+
| IP Header                                     |
.   Source IP Address = Responder IPv4 or IPv6 Address   .
.   Destination IP Address = Source IP Address from Query .
.   Protocol = UDP                                       .
.   Router Alert Option Not Set                         .
.                                                       .
+-----+
| UDP Header                                     |
.   Source Port = As chosen by Responder                 .
.   Destination Port = Source Port from Query            .
.                                                       .
+-----+
| Message as specified in Section 3.2 of RFC 6374 for DM, or |
. Message as specified in Section 3.1 of RFC 6374 for LM   .
.                                                       .
+-----+

```

Figure 5: Probe Response Message

When the received probe query message contains UDP Return Object (URO) TLV [RFC7876], the probe response message the message uses the IP/UDP information from the URO in the probe query message. The content of the probe response message is shown in Figure 6.

```

+-----+
| IP Header                                     |
.   Source IP Address = Responder IPv4 or IPv6 Address   .
.   Destination IP Address = URO.Address                 .
.   Protocol = UDP                                       .
.   Router Alert Option Not Set                         .
.                                                       .
+-----+
| UDP Header                                     |
.   Source Port = As chosen by Responder                 .
.   Destination Port = URO.UDP-Destination-Port          .
.                                                       .
+-----+
| Message as specified in Section 3.2 of RFC 6374 for DM, or |
. Message as specified in Section 3.1 of RFC 6374 for LM   .
.                                                       .
+-----+

```

Figure 6: Probe Response Message Using URO from Probe Query Message

3.2.1. One-way Measurement for SR Link and end-to-end SR Policy

For one-way performance measurement, the probe response message as defined in Figure 5 or Figure 6 is sent out-of-band for both SR links and SR Policies.

The PM querier node can receive probe response message back by properly setting its own IP address as Source Address of the header or by adding URO TLV in the probe query message and setting its own IP address in the IP Address in the URO TLV (Type=131) [RFC7876]. In addition, the "control code" in the probe query message is set to "out-of-band response requested". The "Source Address" TLV (Type 130), and "Return Address" TLV (Type 1), if present in the probe query message, are not used to send probe response message.

3.2.1.1. Probe Response Message to Controller

As shown in the Reference Topology, if the querier node requires the probe response message to be sent to the controller R100, it adds URO TLV in the probe query message and sets the IP address of R100 in the IP Address field and UDP port TBA1 for DM and TBA2 for LM in the UDP-Destination-Port field of the URO TLV (Type=131) [RFC7876].

3.2.2. Two-way Measurement for SR Links

For two-way performance measurement, when using a bidirectional channel, the probe response message as defined in Figure 5 or Figure 6 is sent back in-band to the querier node for SR links. In this case, the "control code" in the probe query message is set to "in-band response requested" [RFC6374].

3.2.3. Two-way End-to-end Measurement of SR Policy

For two-way performance measurement, when using a bidirectional channel, the probe response message is sent back in-band to the querier node for end-to-end measurement of SR Policies. In this case, the "control code" in the probe query message is set to "in-band response requested" [RFC6374].

The path segment identifier [I-D.spring-mpls-path-segment] [I-D.pce-sr-path-segment] of the forward SR Policy can be used to find the reverse SR Policy to send the probe response message in the absence of RPSL TLV defined in the following Section.

3.2.3.1. Return Path Segment List TLV

For two-way performance measurement, the responder node needs to send the probe response message in-band on a specific reverse SR path. This way the destination node does not require any additional SR Policy state. The querier node can request in the probe query

message to the responder node to send a response back on a given reverse path (typically co-routed path for two-way measurement).

[RFC6374] defines DM and LM probe query messages that can include one or more optional TLVs. New TLV Types are defined in this document for Return Path Segment List (RPSL) to carry reverse SR path for probe response messages. The format of the RPSL TLV is shown in Figure 7:

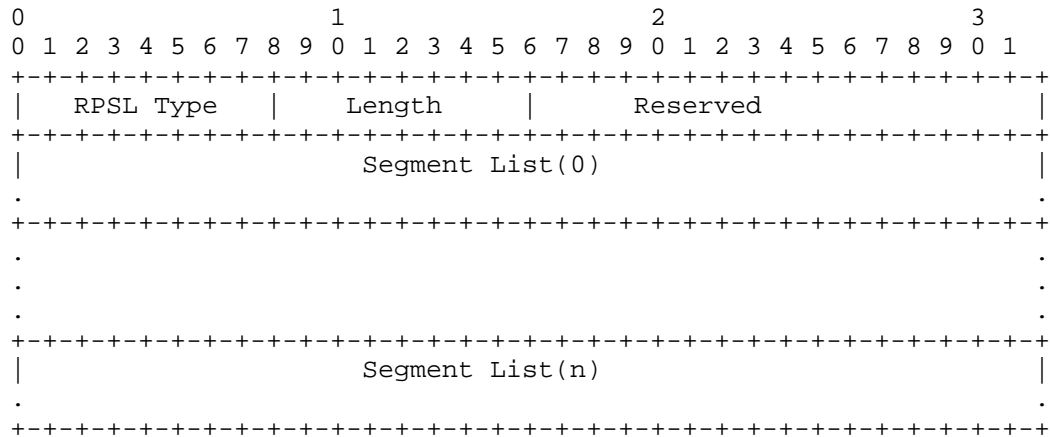


Figure 7: Return Path Segment List TLV

The RPSL can be one of following Types:

- o RPSL Type (value TBA3): SR-MPLS Label Stack of the Reverse SR Policy
- o RPSL Type (value TBA4): SRv6 Segment List of the Reverse SR Policy
- o RPSL Type (value TBA5): SR-MPLS Binding SID [I-D.pce-binding-label-sid] of the Reverse SR Policy
- o RPSL Type (value TBA6): SRv6 Binding SID [I-D.pce-binding-label-sid] of the Reverse SR Policy

The Segment List(0) can be used by the responder node to compute the next-hop IP address and outgoing interface to send the probe response messages.

The RPSL TLV is optional. The PM querier node MUST only insert one RPSL TLV in the probe query message and the responder node MUST only process the first RPSL TLV in the probe query message and ignore

other RPSL TLVs if present. The responder node MUST send probe response message back on the reverse path specified in the RPSL TLV and MUST NOT add RPSL TLV in the probe response message.

3.2.3.2. In-band Probe Response Message for SR-MPLS Policy

The message content for sending probe response message in-band using UDP header for two-way end-to-end performance measurement of an SR-MPLS Policy is shown in Figure 8. The SR-MPLS label stack in the packet header is built using the Segment List received in the RPSL TLV in the probe query message.

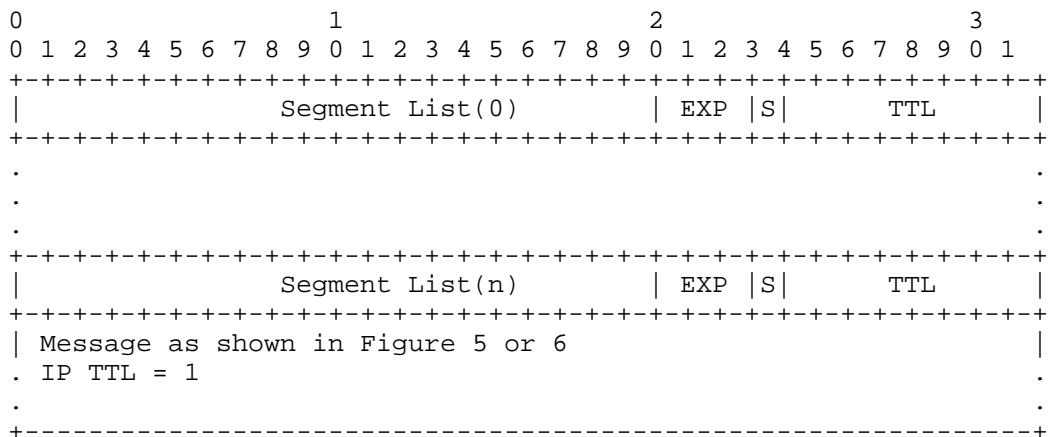


Figure 8: In-band Probe Response Message for SR-MPLS Policy

3.2.3.3. In-band Probe Response Message for SRv6 Policy

The message content for sending probe response message in-band using UDP header for two-way end-to-end performance measurement of an SRv6 Policy is shown in Figure 9. For SRv6 Policy, the SRv6 SID list in the SRH of the probe response message is built using the SRv6 Segment List received in the RPSL TLV in the probe query message.

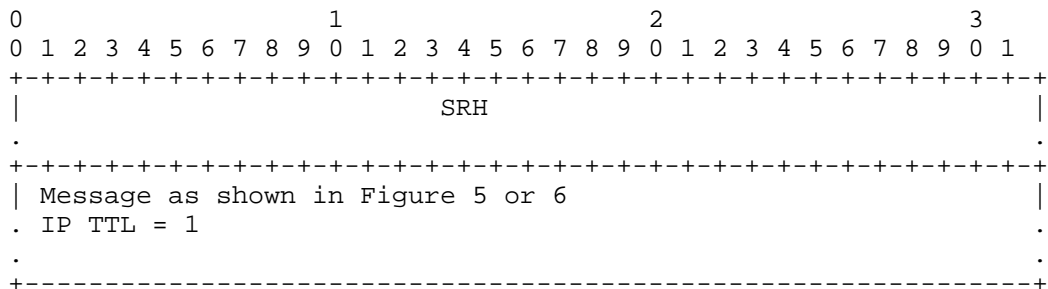


Figure 9: In-band Probe Response Message for SRv6 Policy

4. Performance Measurement for P2MP SR Policies

The procedures for delay and loss measurement described in this document for Point-to-Point (P2P) SR-MPLS Policies are also equally applicable to the Point-to-Multipoint (P2MP) SR Policies.

5. ECMP Support

An SR Policy can have ECMPs between the source and transit nodes, between transit nodes and between transit and destination nodes. The PM probe messages can be sent to traverse different ECMP paths to measure performance of an SR Policy.

Forwarding plane has various hashing functions available to forward packets on specific ECMP paths. Following mechanisms can be used in PM probe messages to take advantage of the hashing function in forwarding plane to influence the path taken by them.

- o The mechanisms described in [RFC8029] [RFC5884] for handling ECMPs are also applicable to the performance measurement. In the IP/UDP header of the PM probe messages, Destination Addresses in 127/8 range for IPv4 or 0:0:0:0:0:FFFF:7F00/104 range for IPv6 can be used to exercise a particular ECMP path. In addition, different Source Addresses or different Source UDP ports can be used for this purpose. As specified in [RFC6437], 3-tuple of Flow Label, Source Address and Destination Address fields in the IPv6 header can also be used.
- o For SR-MPLS, entropy label [RFC6790] in the PM probe messages can be used.
- o For SRv6, Flow Label in SRH [I-D.6man-segment-routing-header] of the PM probe messages can be used.

6. Sequence Number TLV

The message formats for DM and LM [RFC6374] do not contain sequence number for probe query packets. Sequence numbers can be useful when some probe query messages are lost or they arrive out of order.

[RFC6374] defines DM and LM probe query and response messages that can include one or more optional TLVs. New TLV Type (value TBA7) is defined in this document to carry sequence number for probe query and response messages for delay and loss measurement. The format of the

Sequence Number TLV is shown in Figure 10:

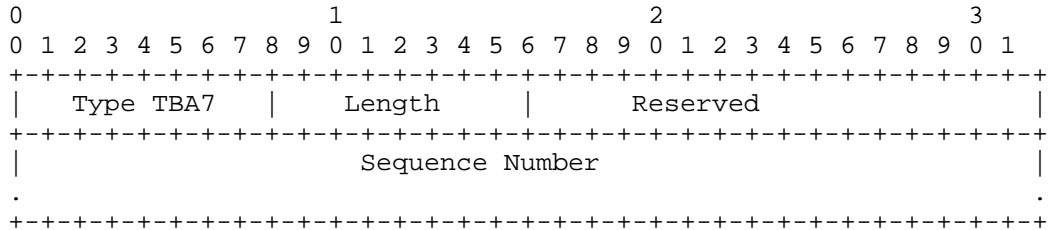


Figure 10: Sequence Number TLV

The sequence numbers start with 0 and are incremented by one for each subsequent probe query packet. The sequence number can be of any length determined by the querier node. The Sequence Number TLV is optional. The PM querier node SHOULD only insert one Sequence Number TLV in the probe query message and the responder node in the probe response message SHOULD return the first Sequence Number TLV from the probe query message and ignore other Sequence Number TLVs if present.

7. Security Considerations

The performance measurement is intended for deployment in well-managed private and service provider networks. The security considerations described in Section 8 of [RFC6374] are applicable to this specification, and particular attention should be paid to the last two paragraphs. Cryptographic measures may be enhanced by the correct configuration of access-control lists and firewalls.

8. IANA Considerations

IANA is requested to allocate following UDP ports for performance measurements:

- o UDP Port TBA1: Delay Performance Measurement
- o UDP Port TBA2: Loss Performance Measurement

IANA is also requested to allocate values for the following Return Path Segment List TLV Types for RFC 6374 to be carried in PM probe query messages:

- o Type TBA3: SR-MPLS Label Stack of the Reverse SR Policy

- o Type TBA4: SRv6 Segment List of the Reverse SR Policy
- o Type TBA5: SR-MPLS Binding SID of the Reverse SR Policy
- o Type TBA6: SRv6 Binding SID of the Reverse SR Policy

IANA is also requested to allocate a value for the following Sequence Number TLV Type for RFC 6374 to be carried in the PM probe query and response messages for delay and loss measurement:

- o Type TBA7: Sequence Number TLV

IANA is also requested to allocate a value for the following Block Number TLV Type for RFC 6374 to be carried in the PM probe query and response messages for loss measurement:

- o Type TBA8: Block Number TLV

9. References

9.1. Normative References

- [RFC768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.
- [RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS networks", RFC 6374, September 2011.
- [RFC7876] Bryant, S., Sivabalan, S., and Soni, S., "UDP Return Path for Packet Loss and Delay Measurement for MPLS Networks", RFC 7876, July 2016.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", RFC 8174, May 2017.

9.2. Informative References

- [IEEE1588] IEEE, "1588-2008 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", March 2008.

- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.
- [RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, DOI 10.17487/RFC5884, June 2010.
- [RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, November 2011.
- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, November 2012.
- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Kumar, N., Aldrin, S. and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, March 2017.
- [RFC8321] Fioccola, G. Ed., "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, January 2018.
- [I-D.spring-segment-routing-policy] Filsfils, C., et al., "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy, work in progress.
- [I-D.spring-sr-p2mp-policy] Voyer, D. Ed., et al., "SR Replication Policy for P2MP Service Delivery", draft-voyer-spring-sr-p2mp-policy, work in progress.
- [I-D.6man-segment-routing-header] Filsfils, C., et al., "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header, work in progress.
- [I-D.spring-sr-mpls-pm] Filsfils, C., Gandhi, R. Ed., et al. "Performance Measurement in Segment Routing Networks with MPLS Data Plane", draft-gandhi-spring-sr-mpls-pm, work in progress.
- [I-D.pce-binding-label-sid] Filsfils, C., et al., "Carrying Binding

Label Segment-ID in PCE-based Networks",
draft-sivabalan-pce-binding-label-sid, work in progress.

[I-D.spring-mpls-path-segment] Cheng, W., et al., "Path Segment in
MPLS Based Segment Routing Network",
draft-cheng-spring-mpls-path-segment, work in progress.

[I-D.pce-sr-path-segment] Li, C., et al., "Path Computation Element
Communication Protocol (PCEP) Extension for Path
Identification in Segment Routing (SR)",
draft-li-pce-sr-path-segment, work in progress.

[I-D.ippm-stamp] Mirsky, G. et al. "Simple Two-way Active
Measurement Protocol", draft-ietf-ippm-stamp, work in
progress.

[BBF.TR-390] "Performance Measurement from IP Edge to Customer
Equipment using TWAMP Light", BBF TR-390, May 2017.

Acknowledgments

The authors would like to thank Nagendra Kumar and Carlos Pignataro for the discussion on SRv6 Performance Measurement.

Contributors

Sagar Soni
Cisco Systems, Inc.
Email: sagsoni@cisco.com

Patrick Khordoc
Cisco Systems, Inc.
Email: pkhordoc@cisco.com

Zafar Ali
Cisco Systems, Inc.
Email: zali@cisco.com

Daniel Bernier
Bell Canada
Email: daniel.bernier@bell.ca

Dirk Steinberg
Steinberg Consulting
Germany
Email: dws@dirksteinberg.de

Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada
Email: rgandhi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
Email: cfilsfil@cisco.com

Daniel Voyer

Bell Canada
Email: daniel.voyer@bell.ca

Stefano Salsano
Universita di Roma "Tor Vergata"
Italy
Email: stefano.salsano@uniroma2.it

Pier Luigi Ventre
CNIT
Italy
Email: pierluigi.ventre@cnit.it

Mach(Guoyi) Chen
Huawei
Email: mach.chen@huawei.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2020

A. Morton
AT&T Labs
M. Bagnulo
UC3M
P. Eardley
BT
K. D'Souza
AT&T Labs
March 9, 2020

Initial Performance Metrics Registry Entries
draft-ietf-ippm-initial-registry-16

Abstract

This memo defines the set of Initial Entries for the IANA Performance Metrics Registry. The set includes: UDP Round-trip Latency and Loss, Packet Delay Variation, DNS Response Latency and Loss, UDP Poisson One-way Delay and Loss, UDP Periodic One-way Delay and Loss, ICMP Round-trip Latency and Loss, and TCP round-trip Latency and Loss.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14[RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 10, 2020.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	6
2. Scope	7
3. Registry Categories and Columns	7
4. UDP Round-trip Latency and Loss Registry Entries	8
4.1. Summary	9
4.1.1. ID (Identifier)	9
4.1.2. Name	9
4.1.3. URI	9
4.1.4. Description	9
4.1.5. Change Controller	9
4.1.6. Version (of Registry Format)	9
4.2. Metric Definition	10
4.2.1. Reference Definition	10
4.2.2. Fixed Parameters	10
4.3. Method of Measurement	11
4.3.1. Reference Method	11
4.3.2. Packet Stream Generation	12
4.3.3. Traffic Filtering (observation) Details	13
4.3.4. Sampling Distribution	13
4.3.5. Run-time Parameters and Data Format	13
4.3.6. Roles	14
4.4. Output	14
4.4.1. Type	14
4.4.2. Reference Definition	14
4.4.3. Metric Units	15
4.4.4. Calibration	15
4.5. Administrative items	16
4.5.1. Status	16
4.5.2. Requester	16
4.5.3. Revision	16
4.5.4. Revision Date	16

4.6.	Comments and Remarks	16
5.	Packet Delay Variation Registry Entry	16
5.1.	Summary	16
5.1.1.	ID (Identifier)	16
5.1.2.	Name	16
5.1.3.	URI	17
5.1.4.	Description	17
5.1.5.	Change Controller	17
5.1.6.	Version (of Registry Format)	17
5.2.	Metric Definition	17
5.2.1.	Reference Definition	17
5.2.2.	Fixed Parameters	18
5.3.	Method of Measurement	19
5.3.1.	Reference Method	19
5.3.2.	Packet Stream Generation	19
5.3.3.	Traffic Filtering (observation) Details	20
5.3.4.	Sampling Distribution	20
5.3.5.	Run-time Parameters and Data Format	20
5.3.6.	Roles	21
5.4.	Output	21
5.4.1.	Type	21
5.4.2.	Reference Definition	21
5.4.3.	Metric Units	22
5.4.4.	Calibration	22
5.5.	Administrative items	23
5.5.1.	Status	23
5.5.2.	Requester	23
5.5.3.	Revision	23
5.5.4.	Revision Date	23
5.6.	Comments and Remarks	23
6.	DNS Response Latency and Loss Registry Entries	23
6.1.	Summary	23
6.1.1.	ID (Identifier)	24
6.1.2.	Name	24
6.1.3.	URI	24
6.1.4.	Description	24
6.1.5.	Change Controller	24
6.1.6.	Version (of Registry Format)	24
6.2.	Metric Definition	24
6.2.1.	Reference Definition	24
6.2.2.	Fixed Parameters	25
6.3.	Method of Measurement	27
6.3.1.	Reference Method	27
6.3.2.	Packet Stream Generation	28
6.3.3.	Traffic Filtering (observation) Details	29
6.3.4.	Sampling Distribution	29
6.3.5.	Run-time Parameters and Data Format	29
6.3.6.	Roles	30

6.4.	Output	30
6.4.1.	Type	30
6.4.2.	Reference Definition	31
6.4.3.	Metric Units	31
6.4.4.	Calibration	31
6.5.	Administrative items	32
6.5.1.	Status	32
6.5.2.	Requester	32
6.5.3.	Revision	32
6.5.4.	Revision Date	32
6.6.	Comments and Remarks	32
7.	UDP Poisson One-way Delay and Loss Registry Entries	32
7.1.	Summary	32
7.1.1.	ID (Identifier)	33
7.1.2.	Name	33
7.1.3.	URI	33
7.1.4.	Description	33
7.2.	Metric Definition	34
7.2.1.	Reference Definition	34
7.2.2.	Fixed Parameters	35
7.3.	Method of Measurement	36
7.3.1.	Reference Method	36
7.3.2.	Packet Stream Generation	36
7.3.3.	Traffic Filtering (observation) Details	37
7.3.4.	Sampling Distribution	37
7.3.5.	Run-time Parameters and Data Format	37
7.3.6.	Roles	38
7.4.	Output	38
7.4.1.	Type	38
7.4.2.	Reference Definition	38
7.4.3.	Metric Units	41
7.4.4.	Calibration	41
7.5.	Administrative items	42
7.5.1.	Status	42
7.5.2.	Requester	42
7.5.3.	Revision	42
7.5.4.	Revision Date	43
7.6.	Comments and Remarks	43
8.	UDP Periodic One-way Delay and Loss Registry Entries	43
8.1.	Summary	43
8.1.1.	ID (Identifier)	43
8.1.2.	Name	43
8.1.3.	URI	44
8.1.4.	Description	44
8.2.	Metric Definition	44
8.2.1.	Reference Definition	44
8.2.2.	Fixed Parameters	45
8.3.	Method of Measurement	46

8.3.1.	Reference Method	46
8.3.2.	Packet Stream Generation	47
8.3.3.	Traffic Filtering (observation) Details	48
8.3.4.	Sampling Distribution	48
8.3.5.	Run-time Parameters and Data Format	48
8.3.6.	Roles	48
8.4.	Output	49
8.4.1.	Type	49
8.4.2.	Reference Definition	49
8.4.3.	Metric Units	52
8.4.4.	Calibration	52
8.5.	Administrative items	53
8.5.1.	Status	53
8.5.2.	Requester	53
8.5.3.	Revision	53
8.5.4.	Revision Date	53
8.6.	Comments and Remarks	54
9.	ICMP Round-trip Latency and Loss Registry Entries	54
9.1.	Summary	54
9.1.1.	ID (Identifier)	54
9.1.2.	Name	54
9.1.3.	URI	54
9.1.4.	Description	55
9.1.5.	Change Controller	55
9.1.6.	Version (of Registry Format)	55
9.2.	Metric Definition	55
9.2.1.	Reference Definition	55
9.2.2.	Fixed Parameters	56
9.3.	Method of Measurement	57
9.3.1.	Reference Method	57
9.3.2.	Packet Stream Generation	58
9.3.3.	Traffic Filtering (observation) Details	59
9.3.4.	Sampling Distribution	59
9.3.5.	Run-time Parameters and Data Format	59
9.3.6.	Roles	59
9.4.	Output	60
9.4.1.	Type	60
9.4.2.	Reference Definition	60
9.4.3.	Metric Units	62
9.4.4.	Calibration	62
9.5.	Administrative items	62
9.5.1.	Status	62
9.5.2.	Requester	63
9.5.3.	Revision	63
9.5.4.	Revision Date	63
9.6.	Comments and Remarks	63
10.	TCP Round-Trip Delay and Loss Registry Entries	63
10.1.	Summary	63

10.1.1.	ID (Identifier)	63
10.1.2.	Name	63
10.1.3.	URI	64
10.1.4.	Description	64
10.1.5.	Change Controller	64
10.1.6.	Version (of Registry Format)	64
10.2.	Metric Definition	65
10.2.1.	Reference Definitions	65
10.2.2.	Fixed Parameters	67
10.3.	Method of Measurement	68
10.3.1.	Reference Methods	68
10.3.2.	Packet Stream Generation	70
10.3.3.	Traffic Filtering (observation) Details	70
10.3.4.	Sampling Distribution	70
10.3.5.	Run-time Parameters and Data Format	70
10.3.6.	Roles	71
10.4.	Output	71
10.4.1.	Type	71
10.4.2.	Reference Definition	71
10.4.3.	Metric Units	73
10.4.4.	Calibration	73
10.5.	Administrative items	73
10.5.1.	Status	73
10.5.2.	Requester	73
10.5.3.	Revision	74
10.5.4.	Revision Date	74
10.6.	Comments and Remarks	74
11.	Security Considerations	74
12.	IANA Considerations	74
13.	Acknowledgements	74
14.	References	75
14.1.	Normative References	75
14.2.	Informative References	77
	Authors' Addresses	78

1. Introduction

This memo proposes an initial set of entries for the Performance Metrics Registry. It uses terms and definitions from the IPPM literature, primarily [RFC2330].

Although there are several standard templates for organizing specifications of performance metrics (see [RFC7679] for an example of the traditional IPPM template, based to large extent on the Benchmarking Methodology Working Group's traditional template in [RFC1242], and see [RFC6390] for a similar template), none of these templates were intended to become the basis for the columns of an IETF-wide registry of metrics. While examining aspects of metric

specifications which need to be registered, it became clear that none of the existing metric templates fully satisfies the particular needs of a registry.

Therefore, [I-D.ietf-ippm-metric-registry] defines the overall format for a Performance Metrics Registry. Section 5 of [I-D.ietf-ippm-metric-registry] also gives guidelines for those requesting registration of a Metric, that is the creation of entry(s) in the Performance Metrics Registry: "In essence, there needs to be evidence that a candidate Registered Performance Metric has significant industry interest, or has seen deployment, and there is agreement that the candidate Registered Performance Metric serves its intended purpose." The process in [I-D.ietf-ippm-metric-registry] also requires that new entries are administered by IANA through Specification Required policy, which will ensure that the metrics are tightly defined.

2. Scope

This document defines a set of initial Performance Metrics Registry entries. Most are Active Performance Metrics, which are based on RFCs prepared in the IPPM working group of the IETF, according to their framework [RFC2330] and its updates.

3. Registry Categories and Columns

This memo uses the terminology defined in [I-D.ietf-ippm-metric-registry].

This section provides the categories and columns of the registry, for easy reference. An entry (row) therefore gives a complete description of a Registered Metric.

Legend:

Registry Categories and Columns, shown as

Category	
Column	Column

Summary

Identifier	Name	URI	Desc.	Reference	Change Controller	Ver
------------	------	-----	-------	-----------	-------------------	-----

Metric Definition

Reference Definition	Fixed Parameters
----------------------	------------------

Method of Measurement

Reference Method	Packet Stream Generation	Traffic Filter	Sampling Distribution	Run-time Parameters	Role
------------------	--------------------------	----------------	-----------------------	---------------------	------

Output

Type	Reference Definition	Units	Calibration
------	----------------------	-------	-------------

Administrative Information

Status	Requester	Rev	Rev.Date
--------	-----------	-----	----------

Comments and Remarks

4. UDP Round-trip Latency and Loss Registry Entries

This section specifies an initial registry entry for the UDP Round-trip Latency, and another entry for UDP Round-trip Loss Ratio.

Note: Each Registry entry only produces a "raw" output or a statistical summary. To describe both "raw" and one or more statistics efficiently, the Identifier, Name, and Output Categories can be split and a single section can specify two or more closely-related metrics. For example, this section specifies two Registry entries with many common columns. See Section 7 for an example specifying multiple Registry entries with many common columns.

All column entries beside the ID, Name, Description, and Output Reference Method categories are the same, thus this section proposes

two closely-related registry entries. As a result, IANA is also asked to assign a corresponding URL to each Named Metric.

4.1. Summary

This category includes multiple indexes to the registry entry: the element ID and metric name.

4.1.1. ID (Identifier)

IANA is asked to assign different numeric identifiers to each of the two Named Metrics.

4.1.2. Name

RTDelay_Active_IP-UDP-Periodic_RFCXXXXsec4_Seconds_95Percentile

RTLoss_Active_IP-UDP-Periodic_RFCXXXXsec4_Percent_LossRatio

4.1.3. URI

URL: <https://www.iana.org/> ... <name>

4.1.4. Description

RTDelay: This metric assesses the delay of a stream of packets exchanged between two hosts (which are the two measurement points), and the Output is the Round-trip delay for all successfully exchanged packets expressed as the 95th percentile of their conditional delay distribution.

RTLoss: This metric assesses the loss ratio of a stream of packets exchanged between two hosts (which are the two measurement points), and the Output is the Round-trip loss ratio for all successfully exchanged packets expressed as a percentage.

4.1.5. Change Controller

IETF

4.1.6. Version (of Registry Format)

1.0

4.2. Metric Definition

This category includes columns to prompt the entry of all necessary details related to the metric definition, including the RFC reference and values of input factors, called fixed parameters.

4.2.1. Reference Definition

Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, September 1999.

[RFC2681]

Section 2.4 of [RFC2681] provides the reference definition of the singleton (single value) Round-trip delay metric. Section 3.4 of [RFC2681] provides the reference definition expanded to cover a multi-singleton sample. Note that terms such as singleton and sample are defined in Section 11 of [RFC2330].

Note that although the [RFC2681] definition of "Round-trip-Delay between Src and Dst" is directionally ambiguous in the text, this metric tightens the definition further to recognize that the host in the "Src" role will send the first packet to "Dst", and ultimately receive the corresponding return packet from "Dst" (when neither are lost).

Finally, note that the variable "dT" is used in [RFC2681] to refer to the value of Round-trip delay in metric definitions and methods. The variable "dT" has been re-used in other IPPM literature to refer to different quantities, and cannot be used as a global variable name.

Morton, A., "Round-trip Packet Loss Metrics", RFC 6673, August 2012.

[RFC6673]

Both delay and loss metrics employ a maximum waiting time for received packets, so the count of lost packets to total packets sent is the basis for the loss ratio calculation as per Section 6.1 of [RFC6673].

4.2.2. Fixed Parameters

Type-P as defined in Section 13 of [RFC2330]:

- o IPv4 header values:

- * DSCP: set to 0

- * TTL: set to 255
 - * Protocol: set to 17 (UDP)
 - o IPv6 header values:
 - * DSCP: set to 0
 - * Hop Count: set to 255
 - * Next Header: set to 17 (UDP)
 - * Flow Label: set to zero
 - * Extension Headers: none
 - o UDP header values:
 - * Checksum: the checksum MUST be calculated and the non-zero checksum included in the header
 - o UDP Payload
 - * total of 100 bytes
- Other measurement parameters:
- o Tmax: a loss threshold waiting time
 - * 3.0, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms), with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

4.3. Method of Measurement

This category includes columns for references to relevant sections of the RFC(s) and any supplemental information needed to ensure an unambiguous methods for implementations.

4.3.1. Reference Method

The methodology for this metric is defined as Type-P-Round-trip-Delay-Poisson-Stream in section 2.6 of RFC 2681 [RFC2681] and section 3.6 of RFC 2681 [RFC2681] using the Type-P and Tmax defined under Fixed Parameters. However, the Periodic stream will be generated according to [RFC3432].

The reference method distinguishes between long-delayed packets and lost packets by implementing a maximum waiting time for packet arrival. Tmax is the waiting time used as the threshold to declare a packet lost. Lost packets SHALL be designated as having undefined delay, and counted for the RTLoss metric.

The calculations on the delay (RTT) SHALL be performed on the conditional distribution, conditioned on successful packet arrival within Tmax. Also, when all packet delays are stored, the process which calculates the RTT value MUST enforce the Tmax threshold on stored values before calculations. See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

The reference method requires some way to distinguish between different packets in a stream to establish correspondence between sending times and receiving times for each successfully-arriving packet. Sequence numbers or other send-order identification MUST be retained at the Src or included with each packet to disambiguate packet reordering if it occurs.

If a standard measurement protocol is employed, then the measurement process will determine the sequence numbers or timestamps applied to test packets after the Fixed and Runtime parameters are passed to that process. The chosen measurement protocol will dictate the format of sequence numbers and time-stamps, if they are conveyed in the packet payload.

Refer to Section 4.4 of [RFC6673] for expanded discussion of the instruction to "send a Type-P packet back to the Src as quickly as possible" in Section 2.6 of RFC 2681 [RFC2681]. Section 8 of [RFC6673] presents additional requirements which MUST be included in the method of measurement for this metric.

4.3.2. Packet Stream Generation

This section gives the details of the packet traffic which is the basis for measurement. In IPPM metrics, this is called the Stream, and can easily be described by providing the list of stream parameters.

Section 3 of [RFC3432] prescribes the method for generating Periodic streams using associated parameters.

incT the nominal duration of inter-packet interval, first bit to first bit, with value 0.0200, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see

section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms).

dT the duration of the interval for allowed sample start times, with value 1.0, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms).

NOTE: an initiation process with a number of control exchanges resulting in unpredictable start times (within a time interval) may be sufficient to avoid synchronization of periodic streams, and therefore a valid replacement for selecting a start time at random from a fixed interval.

The T0 parameter will be reported as a measured parameter. Parameters incT and dT are Fixed Parameters.

4.3.3. Traffic Filtering (observation) Details

NA

4.3.4. Sampling Distribution

NA

4.3.5. Run-time Parameters and Data Format

Run-time Parameters are input factors that must be determined, configured into the measurement system, and reported with the results for the context to be complete.

Src the IP address of the host in the Src Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see Section 4 of [RFC6991])

Dst the IP address of the host in the Dst Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see section 4 of [RFC6991])

T0 a time, the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a start time is unspecified and Tf is to be interpreted as the Duration of the measurement interval. The start time is controlled through other means.

Tf a time, the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of

[RFC6991])). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a end time date is ignored and Tf is interpreted as the Duration of the measurement interval.

4.3.6. Roles

Src launches each packet and waits for return transmissions from Dst.

Dst waits for each packet from Src and sends a return packet to Src.

4.4. Output

This category specifies all details of the Output of measurements using the metric.

4.4.1. Type

Percentile -- for the conditional distribution of all packets with a valid value of Round-trip delay (undefined delays are excluded), a single value corresponding to the 95th percentile, as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

The percentile = 95, meaning that the reported delay, "95Percentile", is the smallest value of Round-trip delay for which the Empirical Distribution Function (EDF), $F(95\text{Percentile}) \geq 95\%$ of the singleton Round-trip delay values in the conditional distribution. See section 11.3 of [RFC2330] for the definition of the percentile statistic using the EDF.

LossRatio -- the count of lost packets to total packets sent is the basis for the loss ratio calculation as per Section 6.1 of [RFC6673].

4.4.2. Reference Definition

For all outputs ---

T0 the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991])). The UTC Time Zone is required by Section 6.1 of [RFC2330].

Tf the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of

[RFC6991])). The UTC Time Zone is required by Section 6.1 of [RFC2330].

TotalPkts the count of packets sent by the Src to Dst during the measurement interval.

For

RTDelay_Active_IP-UDP-Periodic_RFCXXXXsec4_Seconds_95Percentile:

95Percentile The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.0000000001 seconds (1.0 ns).

For

RTLoss_Active_IP-UDP-Periodic_RFCXXXXsec4_Percent_LossRatio:

Percentile The numeric value of the result is expressed in units of lost packets to total packets times 100%, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.0000000001.

4.4.3. Metric Units

The 95th Percentile of Round-trip Delay is expressed in seconds.

The Round-trip Loss Ratio is expressed as a percentage of lost packets to total packets sent.

4.4.4. Calibration

Section 3.7.3 of [RFC7679] provides a means to quantify the systematic and random errors of a time measurement. In-situ calibration could be enabled with an internal loopback at the Source host that includes as much of the measurement system as possible, performs address manipulation as needed, and provides some form of isolation (e.g., deterministic delay) to avoid send-receive interface contention. Some portion of the random and systematic error can be characterized this way.

When a measurement controller requests a calibration measurement, the loopback is applied and the result is output in the same format as a normal measurement with additional indication that it is a calibration result.

Both internal loopback calibration and clock synchronization can be used to estimate the available accuracy of the Output Metric Units. For example, repeated loopback delay measurements will reveal the portion of the Output result resolution which is the result of system noise, and thus inaccurate.

4.5. Administrative items

4.5.1. Status

Current

4.5.2. Requester

This RFC number

4.5.3. Revision

1.0

4.5.4. Revision Date

YYYY-MM-DD

4.6. Comments and Remarks

None.

5. Packet Delay Variation Registry Entry

This section gives an initial registry entry for a Packet Delay Variation metric.

5.1. Summary

This category includes multiple indexes to the registry entries, the element ID and metric name.

5.1.1. ID (Identifier)

<insert numeric identifier, an integer>

5.1.2. Name

OWPDV_Active_IP-UDP-Periodic_RFCXXXXsec5_Seconds_95Percentile

5.1.3. URI

URL: <https://www.iana.org/> ... <name>

5.1.4. Description

An assessment of packet delay variation with respect to the minimum delay observed on the periodic stream, and the Output is expressed as the 95th percentile of the packet delay variation distribution.

5.1.5. Change Controller

IETF

5.1.6. Version (of Registry Format)

1.0

5.2. Metric Definition

This category includes columns to prompt the entry of all necessary details related to the metric definition, including the RFC reference and values of input factors, called fixed parameters.

5.2.1. Reference Definition

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

Demichelis, C. and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, November 2002. [RFC3393]

Morton, A. and B. Claise, "Packet Delay Variation Applicability Statement", RFC 5481, March 2009. [RFC5481]

Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010. [RFC5905]

See sections 2.4 and 3.4 of [RFC3393]. Singleton delay differences measured are referred to by the variable name "ddT" (applicable to all forms of delay variation). However, this metric entry specifies the PDV form defined in section 4.2 of [RFC5481], where the singleton PDV for packet *i* is referred to by the variable name "PDV(*i*)".

5.2.2. Fixed Parameters

- o IPv4 header values:
 - * DSCP: set to 0
 - * TTL: set to 255
 - * Protocol: set to 17 (UDP)
- o IPv6 header values:
 - * DSCP: set to 0
 - * Hop Count: set to 255
 - * Next Header: set to 17 (UDP)
 - * Flow Label: set to zero
 - * Extension Headers: none
- o UDP header values:
 - * Checksum: the checksum MUST be calculated and the non-zero checksum included in the header
- o UDP Payload
 - * total of 200 bytes

Other measurement parameters:

Tmax: a loss threshold waiting time with value 3.0, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms), with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

F a selection function unambiguously defining the packets from the stream selected for the metric. See section 4.2 of [RFC5481] for the PDV form.

See the Packet Stream generation category for two additional Fixed Parameters.

5.3. Method of Measurement

This category includes columns for references to relevant sections of the RFC(s) and any supplemental information needed to ensure an unambiguous methods for implementations.

5.3.1. Reference Method

See section 2.6 and 3.6 of [RFC3393] for general singleton element calculations. This metric entry requires implementation of the PDV form defined in section 4.2 of [RFC5481]. Also see measurement considerations in section 8 of [RFC5481].

The reference method distinguishes between long-delayed packets and lost packets by implementing a maximum waiting time for packet arrival. Tmax is the waiting time used as the threshold to declare a packet lost. Lost packets SHALL be designated as having undefined delay.

The calculations on the one-way delay SHALL be performed on the conditional distribution, conditioned on successful packet arrival within Tmax. Also, when all packet delays are stored, the process which calculates the one-way delay value MUST enforce the Tmax threshold on stored values before calculations. See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

The reference method requires some way to distinguish between different packets in a stream to establish correspondence between sending times and receiving times for each successfully-arriving packet. Sequence numbers or other send-order identification MUST be retained at the Src or included with each packet to disambiguate packet reordering if it occurs.

If a standard measurement protocol is employed, then the measurement process will determine the sequence numbers or timestamps applied to test packets after the Fixed and Runtime parameters are passed to that process. The chosen measurement protocol will dictate the format of sequence numbers and time-stamps, if they are conveyed in the packet payload.

5.3.2. Packet Stream Generation

This section gives the details of the packet traffic which is the basis for measurement. In IPPM metrics, this is called the Stream, and can easily be described by providing the list of stream parameters.

Section 3 of [RFC3432] prescribes the method for generating Periodic streams using associated parameters.

incT the nominal duration of inter-packet interval, first bit to first bit, with value 0.0200, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms).

dT the duration of the interval for allowed sample start times, with value 1.0, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms).

NOTE: an initiation process with a number of control exchanges resulting in unpredictable start times (within a time interval) may be sufficient to avoid synchronization of periodic streams, and therefore a valid replacement for selecting a start time at random from a fixed interval.

The T0 parameter will be reported as a measured parameter. Parameters incT and dT are Fixed Parameters.

5.3.3. Traffic Filtering (observation) Details

NA

5.3.4. Sampling Distribution

NA

5.3.5. Run-time Parameters and Data Format

Src the IP address of the host in the Src Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see Section 4 of [RFC6991])

Dst the IP address of the host in the Dst Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see section 4 of [RFC6991])

T0 a time, the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a start time is unspecified and Tf is to be interpreted as the Duration of the measurement interval. The start time is controlled through other means.

Tf a time, the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a end time date is ignored and Tf is interpreted as the Duration of the measurement interval.

5.3.6. Roles

Src launches each packet and waits for return transmissions from Dst.

Dst waits for each packet from Src and sends a return packet to Src.

5.4. Output

This category specifies all details of the Output of measurements using the metric.

5.4.1. Type

Percentile -- for the conditional distribution of all packets with a valid value of one-way delay (undefined delays are excluded), a single value corresponding to the 95th percentile, as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

The percentile = 95, meaning that the reported delay, "95Percentile", is the smallest value of one-way PDV for which the Empirical Distribution Function (EDF), $F(95\text{Percentile}) \geq 95\%$ of the singleton one-way PDV values in the conditional distribution. See section 11.3 of [RFC2330] for the definition of the percentile statistic using the EDF.

5.4.2. Reference Definition

T0 the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

Tf the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

95Percentile The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

5.4.3. Metric Units

The 95th Percentile of one-way PDV is expressed in seconds.

5.4.4. Calibration

Section 3.7.3 of [RFC7679] provides a means to quantify the systematic and random errors of a time measurement. In-situ calibration could be enabled with an internal loopback that includes as much of the measurement system as possible, performs address manipulation as needed, and provides some form of isolation (e.g., deterministic delay) to avoid send-receive interface contention. Some portion of the random and systematic error can be characterized this way.

For one-way delay measurements, the error calibration must include an assessment of the internal clock synchronization with its external reference (this internal clock is supplying timestamps for measurement). In practice, the time offsets [RFC5905] of clocks at both the source and destination are needed to estimate the systematic error due to imperfect clock synchronization (the time offsets are smoothed, thus the random variation is not usually represented in the results).

time_offset The time value of the result is expressed in units of seconds, as a signed value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

When a measurement controller requests a calibration measurement, the loopback is applied and the result is output in the same format as a normal measurement with additional indication that it is a calibration result. In any measurement, the measurement function SHOULD report its current estimate of time offset [RFC5905] as an indicator of the degree of synchronization.

Both internal loopback calibration and clock synchronization can be used to estimate the available accuracy of the Output Metric Units. For example, repeated loopback delay measurements will reveal the portion of the Output result resolution which is the result of system noise, and thus inaccurate.

5.5. Administrative items

5.5.1. Status

Current

5.5.2. Requester

This RFC number

5.5.3. Revision

1.0

5.5.4. Revision Date

YYYY-MM-DD

5.6. Comments and Remarks

Lost packets represent a challenge for delay variation metrics. See section 4.1 of [RFC3393] and the delay variation applicability statement [RFC5481] for extensive analysis and comparison of PDV and an alternate metric, IPDV.

6. DNS Response Latency and Loss Registry Entries

This section gives initial registry entries for DNS Response Latency and Loss from a network user's perspective, for a specific named resource. The metric can be measured repeatedly using different names. RFC 2681 [RFC2681] defines a Round-trip delay metric. We build on that metric by specifying several of the input parameters to precisely define two metrics for measuring DNS latency and loss.

Note to IANA: Each Registry "Name" below specifies a single registry entry, whose output format varies in accordance with the name.

All column entries beside the ID, Name, Description, and Output Reference Method categories are the same, thus this section proposes two closely-related registry entries. As a result, IANA is also asked to assign corresponding URLs to each Named Metric.

6.1. Summary

This category includes multiple indexes to the registry entries, the element ID and metric name.

6.1.1. ID (Identifier)

<insert numeric identifier, an integer>

IANA is asked to assign different numeric identifiers to each of the two Named Metrics.

6.1.2. Name

RTDNS_Active_IP-UDP-Poisson_RFCXXXXsec6_Seconds_Raw

RLDNS_Active_IP-UDP-Poisson_RFCXXXXsec6_Logical_Raw

6.1.3. URI

URL: <https://www.iana.org/> ... <name>

6.1.4. Description

This is a metric for DNS Response performance from a network user's perspective, for a specific named resource. The metric can be measured repeatedly using different resource names.

RTDNS: This metric assesses the response time, the interval from the query transmission to the response.

RLDNS: This metric indicates that the response was deemed lost. In other words, the response time exceeded the maximum waiting time.

6.1.5. Change Controller

IETF

6.1.6. Version (of Registry Format)

1.0

6.2. Metric Definition

This category includes columns to prompt the entry of all necessary details related to the metric definition, including the RFC reference and values of input factors, called fixed parameters.

6.2.1. Reference Definition

Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987. (and updates)

[RFC1035]

Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, September 1999.

[RFC2681]

Section 2.4 of [RFC2681] provides the reference definition of the singleton (single value) Round-trip delay metric. Section 3.4 of [RFC2681] provides the reference definition expanded to cover a multi-singleton sample. Note that terms such as singleton and sample are defined in Section 11 of [RFC2330].

For DNS Response Latency, the entities in [RFC1035] must be mapped to [RFC2681]. The Local Host with its User Program and Resolver take the role of "Src", and the Foreign Name Server takes the role of "Dst".

Note that although the [RFC2681] definition of "Round-trip-Delay between Src and Dst at T" is directionally ambiguous in the text, this metric tightens the definition further to recognize that the host in the "Src" role will send the first packet to "Dst", and ultimately receive the corresponding return packet from "Dst" (when neither are lost).

Morton, A., "Round-trip Packet Loss Metrics", RFC 6673, August 2012.

[RFC6673]

Both response time and loss metrics employ a maximum waiting time for received responses, so the count of lost packets to total packets sent is the basis for the loss determination as per Section 4.3 of [RFC6673].

6.2.2. Fixed Parameters

Type-P as defined in Section 13 of [RFC2330]:

- o IPv4 header values:

- * DSCP: set to 0
- * TTL set to 255
- * Protocol: set to 17 (UDP)

- o IPv6 header values:

- * DSCP: set to 0
- * Hop Count: set to 255
- * Next Header: set to 17 (UDP)
- * Flow Label: set to zero
- * Extension Headers: none
- o UDP header values:
 - * Source port: 53
 - * Destination port: 53
 - * Checksum: the checksum must be calculated and the non-zero checksum included in the header
- o Payload: The payload contains a DNS message as defined in RFC 1035 [RFC1035] with the following values:
 - * The DNS header section contains:
 - + Identification (see the Run-time column)
 - + QR: set to 0 (Query)
 - + OPCODE: set to 0 (standard query)
 - + AA: not set
 - + TC: not set
 - + RD: set to one (recursion desired)
 - + RA: not set
 - + RCODE: not set
 - + QDCOUNT: set to one (only one entry)
 - + ANCOUNT: not set
 - + NSCOUNT: not set
 - + ARCOUNT: not set

- * The Question section contains:
 - + QNAME: the Fully Qualified Domain Name (FQDN) provided as input for the test, see the Run-time column
 - + QTYPE: the query type provided as input for the test, see the Run-time column
 - + QCLASS: set to 1 for IN
- * The other sections do not contain any Resource Records.

Other measurement parameters:

- o Tmax: a loss threshold waiting time (and to help disambiguate queries)
 - * 5.0, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms), with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

Observation: reply packets will contain a DNS response and may contain RRs.

6.3. Method of Measurement

This category includes columns for references to relevant sections of the RFC(s) and any supplemental information needed to ensure an unambiguous methods for implementations.

6.3.1. Reference Method

The methodology for this metric is defined as Type-P-Round-trip-Delay-Poisson-Stream in section 2.6 of RFC 2681 [RFC2681] and section 3.6 of RFC 2681 [RFC2681] using the Type-P and Timeout defined under Fixed Parameters.

The reference method distinguishes between long-delayed packets and lost packets by implementing a maximum waiting time for packet arrival. Tmax is the waiting time used as the threshold to declare a response packet lost. Lost packets SHALL be designated as having undefined delay and counted for the RLDNS metric.

The calculations on the delay (RTT) SHALL be performed on the conditional distribution, conditioned on successful packet arrival within Tmax. Also, when all packet delays are stored, the process

which calculates the RTT value MUST enforce the Tmax threshold on stored values before calculations. See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

The reference method requires some way to distinguish between different packets in a stream to establish correspondence between sending times and receiving times for each successfully-arriving reply.

DNS Messages bearing Queries provide for random ID Numbers in the Identification header field, so more than one query may be launched while a previous request is outstanding when the ID Number is used. Therefore, the ID Number MUST be retained at the Src and included with each response packet to disambiguate packet reordering if it occurs.

IF a DNS response does not arrive within Tmax, the response time RTDNS is undefined, and RLDNS = 1. The Message ID SHALL be used to disambiguate the successive queries that are otherwise identical.

Since the ID Number field is only 16 bits in length, it places a limit on the number of simultaneous outstanding DNS queries during a stress test from a single Src address.

Refer to Section 4.4 of [RFC6673] for expanded discussion of the instruction to "send a Type-P packet back to the Src as quickly as possible" in Section 2.6 of RFC 2681 [RFC2681]. However, the DNS Server is expected to perform all required functions to prepare and send a response, so the response time will include processing time and network delay. Section 8 of [RFC6673] presents additional requirements which SHALL be included in the method of measurement for this metric.

In addition to operations described in [RFC2681], the Src MUST parse the DNS headers of the reply and prepare the query response information for subsequent reporting as a measured result, along with the Round-Trip Delay.

6.3.2. Packet Stream Generation

This section gives the details of the packet traffic which is the basis for measurement. In IPPM metrics, this is called the Stream, and can easily be described by providing the list of stream parameters.

Section 11.1.3 of RFC 2681 [RFC2330] provides three methods to generate Poisson sampling intervals. The reciprocal of lambda is the average packet spacing, thus the Run-time Parameter is $\text{Reciprocal_lambda} = 1/\text{lambda}$, in seconds.

Method 3 is used, where given a start time (Run-time Parameter), the subsequent send times are all computed prior to measurement by computing the pseudo-random distribution of inter-packet send times, (truncating the distribution as specified in the Run-time Parameters), and the Src sends each packet at the computed times.

Note that Trunc is the upper limit on inter-packet times in the Poisson distribution. A random value greater than Trunc is set equal to Trunc instead.

6.3.3. Traffic Filtering (observation) Details

NA

6.3.4. Sampling Distribution

NA

6.3.5. Run-time Parameters and Data Format

Run-time Parameters are input factors that must be determined, configured into the measurement system, and reported with the results for the context to be complete.

Src the IP address of the host in the Src Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see Section 4 of [RFC6991])

Dst the IP address of the host in the Dst Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see section 4 of [RFC6991])

T0 a time, the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a start time is unspecified and Tf is to be interpreted as the Duration of the measurement interval. The start time is controlled through other means.

Tf a time, the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of

[RFC2330]. When T0 is "all-zeros", a end time date is ignored and Tf is interpreted as the Duration of the measurement interval.

Reciprocal_lambda average packet interval for Poisson Streams expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) with resolution of 0.0001 seconds (0.1 ms), and with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

Trunc Upper limit on Poisson distribution expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) with resolution of 0.0001 seconds (0.1 ms), and with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905] (values above this limit will be clipped and set to the limit value).

ID The 16-bit identifier assigned by the program that generates the query, and which must vary in successive queries (a list of IDs is needed), see Section 4.1.1 of [RFC1035]. This identifier is copied into the corresponding reply and can be used by the requester (Src) to match-up replies to outstanding queries.

QNAME The domain name of the Query, formatted as specified in section 4 of [RFC6991].

QTYPE The Query Type, which will correspond to the IP address family of the query (decimal 1 for IPv4 or 28 for IPv6, formatted as a uint16, as per section 9.2 of [RFC6020]).

6.3.6. Roles

Src launches each packet and waits for return transmissions from Dst.

Dst waits for each packet from Src and sends a return packet to Src.

6.4. Output

This category specifies all details of the Output of measurements using the metric.

6.4.1. Type

Raw -- for each DNS Query packet sent, sets of values as defined in the next column, including the status of the response, only assigning delay values to successful query-response pairs.

6.4.2. Reference Definition

For all outputs:

T the time the DNS Query was sent during the measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

dT The time value of the round-trip delay to receive the DNS response, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]. This value is undefined when the response packet is not received at Src within waiting time Tmax seconds.

Rcode The value of the Rcode field in the DNS response header, expressed as a uint64 as specified in section 9.2 of [RFC6020]. Non-zero values convey errors in the response, and such replies must be analyzed separately from successful requests.

6.4.3. Metric Units

RTDNS: Round-trip Delay, dT, is expressed in seconds.

RTLDNS: the Logical value, where 1 = Lost and 0 = Received.

6.4.4. Calibration

Section 3.7.3 of [RFC7679] provides a means to quantify the systematic and random errors of a time measurement. In-situ calibration could be enabled with an internal loopback at the Source host that includes as much of the measurement system as possible, performs address and payload manipulation as needed, and provides some form of isolation (e.g., deterministic delay) to avoid send-receive interface contention. Some portion of the random and systematic error can be characterized this way.

When a measurement controller requests a calibration measurement, the loopback is applied and the result is output in the same format as a normal measurement with additional indication that it is a calibration result.

Both internal loopback calibration and clock synchronization can be used to estimate the available accuracy of the Output Metric Units. For example, repeated loopback delay measurements will reveal the

portion of the Output result resolution which is the result of system noise, and thus inaccurate.

6.5. Administrative items

6.5.1. Status

Current

6.5.2. Requester

This RFC number

6.5.3. Revision

1.0

6.5.4. Revision Date

YYYY-MM-DD

6.6. Comments and Remarks

None

7. UDP Poisson One-way Delay and Loss Registry Entries

This section specifies five initial registry entries for the UDP Poisson One-way Delay, and one for UDP Poisson One-way Loss.

IANA Note: Registry "Name" below specifies multiple registry entries, whose output format varies according to the <statistic> element of the name that specifies one form of statistical summary. There is an additional metric name for the Loss metric.

All column entries beside the ID, Name, Description, and Output Reference Method categories are the same, thus this section proposes six closely-related registry entries. As a result, IANA is also asked to assign corresponding URLs to each Named Metric.

7.1. Summary

This category includes multiple indexes to the registry entries, the element ID and metric name.

7.1.1. ID (Identifier)

IANA is asked to assign different numeric identifiers to each of the six Metrics.

7.1.2. Name

OWDelay_Active_IP-UDP-Poisson-
Payload250B_RFCXXXXsec7_Seconds_<statistic>

where <statistic> is one of:

- o 95Percentile
- o Mean
- o Min
- o Max
- o StdDev

OWLoss_Active_IP-UDP-Poisson-
Payload250B_RFCXXXXsec7_Percent_LossRatio

7.1.3. URI

URL: <https://www.iana.org/> ... <name>

7.1.4. Description

OWDelay: This metric assesses the delay of a stream of packets exchanged between two hosts (or measurement points), and reports the <statistic> One-way delay for all successfully exchanged packets based on their conditional delay distribution.

where <statistic> is one of:

- o 95Percentile
- o Mean
- o Min
- o Max
- o StdDev

OWLoss: This metric assesses the loss ratio of a stream of packets exchanged between two hosts (which are the two measurement points), and the Output is the One-way loss ratio for all successfully received packets expressed as a percentage.

7.2. Metric Definition

This category includes columns to prompt the entry of all necessary details related to the metric definition, including the RFC reference and values of input factors, called fixed parameters.

7.2.1. Reference Definition

For Delay:

Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <<http://www.rfc-editor.org/info/rfc7679>>.

[RFC7679]

Morton, A., and Stephan, E., "Spatial Composition of Metrics", RFC 6049, January 2011.

[RFC6049]

Section 3.4 of [RFC7679] provides the reference definition of the singleton (single value) One-way delay metric. Section 4.4 of [RFC7679] provides the reference definition expanded to cover a multi-value sample. Note that terms such as singleton and sample are defined in Section 11 of [RFC2330].

Only successful packet transfers with finite delay are included in the sample, as prescribed in section 4.1.2 of [RFC6049].

For loss:

Almes, G., Kalidini, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Loss Metric for IP Performance Metrics (IPPM)", RFC 7680, DOI 10.17487/RFC7680, January 2016, <<http://www.rfc-editor.org/info/rfc7680>>.

Section 2.4 of [RFC7680] provides the reference definition of the singleton (single value) one-way loss metric. Section 3.4 of [RFC7680] provides the reference definition expanded to cover a multi-singleton sample. Note that terms such as singleton and sample are defined in Section 11 of [RFC2330].

7.2.2. Fixed Parameters

Type-P:

- o IPv4 header values:

- * DSCP: set to 0
- * TTL: set to 255
- * Protocol: Set to 17 (UDP)

- o IPv6 header values:

- * DSCP: set to 0
- * Hop Count: set to 255
- * Next Header: set to 17 (UDP)
- * Flow Label: set to zero
- * Extension Headers: none

- o UDP header values:

- * Checksum: the checksum MUST be calculated and the non-zero checksum included in the header

- o UDP Payload: TWAMP Test Packet Formats, Section 4.1.2 of [RFC5357]

- * Security features in use influence the number of Padding octets.
- * 250 octets total, including the TWAMP format type, which MUST be reported.

Other measurement parameters:

Tmax: a loss threshold waiting time with value 3.0, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms), with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

See the Packet Stream generation category for two additional Fixed Parameters.

7.3. Method of Measurement

This category includes columns for references to relevant sections of the RFC(s) and any supplemental information needed to ensure an unambiguous methods for implementations.

7.3.1. Reference Method

The methodology for this metric is defined as Type-P-One-way-Delay-Poisson-Stream in section 3.6 of [RFC7679] and section 4.6 of [RFC7679] using the Type-P and Tmax defined under Fixed Parameters.

The reference method distinguishes between long-delayed packets and lost packets by implementing a maximum waiting time for packet arrival. Tmax is the waiting time used as the threshold to declare a packet lost. Lost packets SHALL be designated as having undefined delay, and counted for the OWLoss metric.

The calculations on the one-way delay SHALL be performed on the conditional distribution, conditioned on successful packet arrival within Tmax. Also, when all packet delays are stored, the process which calculates the one-way delay value MUST enforce the Tmax threshold on stored values before calculations. See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

The reference method requires some way to distinguish between different packets in a stream to establish correspondence between sending times and receiving times for each successfully-arriving packet.

Since a standard measurement protocol is employed [RFC5357], then the measurement process will determine the sequence numbers or timestamps applied to test packets after the Fixed and Runtime parameters are passed to that process. The measurement protocol dictates the format of sequence numbers and time-stamps conveyed in the TWAMP-Test packet payload.

7.3.2. Packet Stream Generation

This section gives the details of the packet traffic which is the basis for measurement. In IPPM metrics, this is called the Stream, and can easily be described by providing the list of stream parameters.

Section 11.1.3 of RFC 2681 [RFC2330] provides three methods to generate Poisson sampling intervals. The reciprocal of lambda is the

average packet spacing, thus the Run-time Parameter is $\text{Reciprocal_lambda} = 1/\text{lambda}$, in seconds.

Method 3 SHALL be used, where given a start time (Run-time Parameter), the subsequent send times are all computed prior to measurement by computing the pseudo-random distribution of inter-packet send times, (truncating the distribution as specified in the Parameter Trunc), and the Src sends each packet at the computed times.

Note that Trunc is the upper limit on inter-packet times in the Poisson distribution. A random value greater than Trunc is set equal to Trunc instead.

Reciprocal_lambda average packet interval for Poisson Streams expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) with resolution of 0.0001 seconds (0.1 ms), and with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905]. $\text{Reciprocal_lambda} = 1$ second.

Trunc Upper limit on Poisson distribution expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) with resolution of 0.0001 seconds (0.1 ms), and with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905] (values above this limit will be clipped and set to the limit value). $\text{Trunc} = 30.0000$ seconds.

7.3.3. Traffic Filtering (observation) Details

NA

7.3.4. Sampling Distribution

NA

7.3.5. Run-time Parameters and Data Format

Run-time Parameters are input factors that must be determined, configured into the measurement system, and reported with the results for the context to be complete.

Src the IP address of the host in the Src Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see Section 4 of [RFC6991])

Dst the IP address of the host in the Dst Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see section 4 of [RFC6991])

T0 a time, the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a start time is unspecified and Tf is to be interpreted as the Duration of the measurement interval. The start time is controlled through other means.

Tf a time, the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a end time date is ignored and Tf is interpreted as the Duration of the measurement interval.

7.3.6. Roles

Src launches each packet and waits for return transmissions from Dst. This is the TWAMP Session-Sender.

Dst waits for each packet from Src and sends a return packet to Src. This is the TWAMP Session-Reflector.

7.4. Output

This category specifies all details of the Output of measurements using the metric.

7.4.1. Type

See subsection titles below for Types.

7.4.2. Reference Definition

For all output types ---

T0 the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

Tf the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

For LossRatio -- the count of lost packets to total packets sent is the basis for the loss ratio calculation as per Section 4.1 of [RFC7680].

For each <statistic>, one of the following sub-sections apply:

7.4.2.1. Percentile95

The 95th percentile SHALL be calculated using the conditional distribution of all packets with a finite value of One-way delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.3 of [RFC3393] for details on the percentile statistic (where Round-trip delay should be substituted for "ipdv").

The percentile = 95, meaning that the reported delay, "95Percentile", is the smallest value of one-way delay for which the Empirical Distribution Function (EDF), $F(95\text{Percentile}) \geq 95\%$ of the singleton one-way delay values in the conditional distribution. See section 11.3 of [RFC2330] for the definition of the percentile statistic using the EDF.

95Percentile The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

7.4.2.2. Mean

The mean SHALL be calculated using the conditional distribution of all packets with a finite value of One-way delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.2.2 of [RFC6049] for details on calculating this statistic, and 4.2.3 of [RFC6049].

Mean The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001

seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

7.4.2.3. Min

The minimum SHALL be calculated using the conditional distribution of all packets with a finite value of One-way delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.3.2 of [RFC6049] for details on calculating this statistic, and 4.3.3 of [RFC6049].

Min The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

7.4.2.4. Max

The maximum SHALL be calculated using the conditional distribution of all packets with a finite value of One-way delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.3.2 of [RFC6049] for a closely related method for calculating this statistic, and 4.3.3 of [RFC6049]. The formula is as follows:

$$\text{Max} = (\text{FiniteDelay}[j])$$

such that for some index, j , where $1 \leq j \leq N$
 $\text{FiniteDelay}[j] \geq \text{FiniteDelay}[n]$ for all n

Max The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

7.4.2.5. Std_Dev

The Std_Dev SHALL be calculated using the conditional distribution of all packets with a finite value of One-way delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 6.1.4 of [RFC6049] for a closely related method for calculating this statistic. The formula is the classic calculation for standard deviation of a population.

Define Population Std_Dev_Delay as follows:

(where all packets $n = 1$ through N have a value for Delay[n], and MeanDelay calculated as in 7.4.2.2), and SQRT[] is the Square Root function:

$$\text{Std_Dev} = \text{SQRT} \left[\frac{1}{N} \sum_{n=1}^N (\text{Delay}[n] - \text{MeanDelay})^2 \right]$$

Std_Dev The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

7.4.3. Metric Units

The <statistic> of One-way Delay is expressed in seconds.

The One-way Loss Ratio is expressed as a percentage of lost packets to total packets sent.

7.4.4. Calibration

Section 3.7.3 of [RFC7679] provides a means to quantify the systematic and random errors of a time measurement. In-situ calibration could be enabled with an internal loopback that includes as much of the measurement system as possible, performs address manipulation as needed, and provides some form of isolation (e.g.,

deterministic delay) to avoid send-receive interface contention. Some portion of the random and systematic error can be characterized this way.

For one-way delay measurements, the error calibration must include an assessment of the internal clock synchronization with its external reference (this internal clock is supplying timestamps for measurement). In practice, the time offsets [RFC5905] of clocks at both the source and destination are needed to estimate the systematic error due to imperfect clock synchronization (the time offsets [RFC5905] are smoothed, thus the random variation is not usually represented in the results).

`time_offset` The time value of the result is expressed in units of seconds, as a signed value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

When a measurement controller requests a calibration measurement, the loopback is applied and the result is output in the same format as a normal measurement with additional indication that it is a calibration result. In any measurement, the measurement function SHOULD report its current estimate of time offset [RFC5905] as an indicator of the degree of synchronization.

Both internal loopback calibration and clock synchronization can be used to estimate the available accuracy of the Output Metric Units. For example, repeated loopback delay measurements will reveal the portion of the Output result resolution which is the result of system noise, and thus inaccurate.

7.5. Administrative items

7.5.1. Status

Current

7.5.2. Requester

This RFC number

7.5.3. Revision

1.0

7.5.4. Revision Date

YYYY-MM-DD

7.6. Comments and Remarks

None

8. UDP Periodic One-way Delay and Loss Registry Entries

This section specifies five initial registry entries for the UDP Periodic One-way Delay, and one for UDP Periodic One-way Loss.

IANA Note: Registry "Name" below specifies multiple registry entries, whose output format varies according to the <statistic> element of the name that specifies one form of statistical summary. There is an additional metric name for the Loss metric.

All column entries beside the ID, Name, Description, and Output Reference Method categories are the same, thus this section proposes six closely-related registry entries. As a result, IANA is also asked to assign corresponding URLs to each Named Metric.

8.1. Summary

This category includes multiple indexes to the registry entries, the element ID and metric name.

8.1.1. ID (Identifier)

IANA is asked to assign a different numeric identifiers to each of the six Metrics.

8.1.2. Name

OWDelay_Active_IP-UDP-Periodic20m-
Payload142B_RFCXXXXsec8_Seconds_<statistic>

where <statistic> is one of:

- o 95Percentile
- o Mean
- o Min
- o Max

- o StdDev

OWLoss_Active_IP-UDP-Periodic-
Payload142B_RFCXXXXsec8_Percent_LossRatio

8.1.3. URI

URL: <https://www.iana.org/> ... <name>

8.1.4. Description

OWDelay: This metric assesses the delay of a stream of packets exchanged between two hosts (or measurement points), and reports the <statistic> One-way delay for all successfully exchanged packets based on their conditional delay distribution.

where <statistic> is one of:

- o 95Percentile
- o Mean
- o Min
- o Max
- o StdDev

OWLoss: This metric assesses the loss ratio of a stream of packets exchanged between two hosts (which are the two measurement points), and the Output is the One-way loss ratio for all successfully received packets expressed as a percentage.

8.2. Metric Definition

This category includes columns to prompt the entry of all necessary details related to the metric definition, including the RFC reference and values of input factors, called fixed parameters.

8.2.1. Reference Definition

For Delay:

Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <<http://www.rfc-editor.org/info/rfc7679>>.

[RFC7679]

Morton, A., and Stephan, E., "Spatial Composition of Metrics", RFC 6049, January 2011.

[RFC6049]

Section 3.4 of [RFC7679] provides the reference definition of the singleton (single value) One-way delay metric. Section 4.4 of [RFC7679] provides the reference definition expanded to cover a multi-value sample. Note that terms such as singleton and sample are defined in Section 11 of [RFC2330].

Only successful packet transfers with finite delay are included in the sample, as prescribed in section 4.1.2 of [RFC6049].

For loss:

Almes, G., Kalidini, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Loss Metric for IP Performance Metrics (IPPM)", RFC 7680, DOI 10.17487/RFC7680, January 2016, <<http://www.rfc-editor.org/info/rfc7680>>.

Section 2.4 of [RFC7680] provides the reference definition of the singleton (single value) one-way loss metric. Section 3.4 of [RFC7680] provides the reference definition expanded to cover a multi-singleton sample. Note that terms such as singleton and sample are defined in Section 11 of [RFC2330].

8.2.2. Fixed Parameters

Type-P:

- o IPv4 header values:

- * DSCP: set to 0
- * TTL: set to 255
- * Protocol: Set to 17 (UDP)

- o IPv6 header values:

- * DSCP: set to 0
- * Hop Count: set to 255
- * Next Header: set to 17 (UDP)

- * Flow Label: set to zero
- * Extension Headers: none
- o UDP header values:
 - * Checksum: the checksum MUST be calculated and the non-zero checksum included in the header
- o UDP Payload: TWAMP Test Packet Formats, Section 4.1.2 of [RFC5357]
 - * Security features in use influence the number of Padding octets.
 - * 142 octets total, including the TWAMP format (and format type MUST be reported, if used)

Other measurement parameters:

Tmax: a loss threshold waiting time with value 3.0, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms), with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

See the Packet Stream generation category for two additional Fixed Parameters.

8.3. Method of Measurement

This category includes columns for references to relevant sections of the RFC(s) and any supplemental information needed to ensure an unambiguous methods for implementations.

8.3.1. Reference Method

The methodology for this metric is defined as Type-P-One-way-Delay-Poisson-Stream in section 3.6 of [RFC7679] and section 4.6 of [RFC7679] using the Type-P and Tmax defined under Fixed Parameters. However, a Periodic stream is used, as defined in [RFC3432].

The reference method distinguishes between long-delayed packets and lost packets by implementing a maximum waiting time for packet arrival. Tmax is the waiting time used as the threshold to declare a packet lost. Lost packets SHALL be designated as having undefined delay, and counted for the OWLoss metric.

The calculations on the one-way delay SHALL be performed on the conditional distribution, conditioned on successful packet arrival within Tmax. Also, when all packet delays are stored, the process which calculates the one-way delay value MUST enforce the Tmax threshold on stored values before calculations. See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

The reference method requires some way to distinguish between different packets in a stream to establish correspondence between sending times and receiving times for each successfully-arriving packet.

Since a standard measurement protocol is employed [RFC5357], then the measurement process will determine the sequence numbers or timestamps applied to test packets after the Fixed and Runtime parameters are passed to that process. The measurement protocol dictates the format of sequence numbers and time-stamps conveyed in the TWAMP-Test packet payload.

8.3.2. Packet Stream Generation

This section gives the details of the packet traffic which is the basis for measurement. In IPPM metrics, this is called the Stream, and can easily be described by providing the list of stream parameters.

Section 3 of [RFC3432] prescribes the method for generating Periodic streams using associated parameters.

incT the nominal duration of inter-packet interval, first bit to first bit, with value 0.0200 expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms), with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

dT the duration of the interval for allowed sample start times, with value 1.0000, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms), with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

T0 the actual start time of the periodic stream, determined from T0 and dT.

NOTE: an initiation process with a number of control exchanges resulting in unpredictable start times (within a time interval) may be sufficient to avoid synchronization of periodic streams, and therefore a valid replacement for selecting a start time at random from a fixed interval.

These stream parameters will be specified as Run-time parameters.

8.3.3. Traffic Filtering (observation) Details

NA

8.3.4. Sampling Distribution

NA

8.3.5. Run-time Parameters and Data Format

Run-time Parameters are input factors that must be determined, configured into the measurement system, and reported with the results for the context to be complete.

Src the IP address of the host in the Src Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see Section 4 of [RFC6991])

Dst the IP address of the host in the Dst Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see section 4 of [RFC6991])

T0 a time, the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a start time is unspecified and Tf is to be interpreted as the Duration of the measurement interval. The start time is controlled through other means.

Tf a time, the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a end time date is ignored and Tf is interpreted as the Duration of the measurement interval.

8.3.6. Roles

Src launches each packet and waits for return transmissions from Dst. This is the TWAMP Session-Sender.

Dst waits for each packet from Src and sends a return packet to Src.
This is the TWAMP Session-Reflector.

8.4. Output

This category specifies all details of the Output of measurements using the metric.

8.4.1. Type

See subsection titles in Reference Definition for Latency Types.

8.4.2. Reference Definition

For all output types ---

T0 the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

Tf the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

For LossRatio -- the count of lost packets to total packets sent is the basis for the loss ratio calculation as per Section 4.1 of [RFC7680].

For each <statistic>, one of the following sub-sections apply:

8.4.2.1. Percentile95

The 95th percentile SHALL be calculated using the conditional distribution of all packets with a finite value of One-way delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.3 of [RFC3393] for details on the percentile statistic (where Round-trip delay should be substituted for "ipdv").

The percentile = 95, meaning that the reported delay, "95Percentile", is the smallest value of one-way delay for which the Empirical Distribution Function (EDF), $F(95\text{Percentile}) \geq 95\%$ of the singleton

one-way delay values in the conditional distribution. See section 11.3 of [RFC2330] for the definition of the percentile statistic using the EDF.

95Percentile The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

8.4.2.2. Mean

The mean SHALL be calculated using the conditional distribution of all packets with a finite value of One-way delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.2.2 of [RFC6049] for details on calculating this statistic, and 4.2.3 of [RFC6049].

Mean The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

8.4.2.3. Min

The minimum SHALL be calculated using the conditional distribution of all packets with a finite value of One-way delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.3.2 of [RFC6049] for details on calculating this statistic, and 4.3.3 of [RFC6049].

Min The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

8.4.2.4. Max

The maximum SHALL be calculated using the conditional distribution of all packets with a finite value of One-way delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.3.2 of [RFC6049] for a closely related method for calculating this statistic, and 4.3.3 of [RFC6049]. The formula is as follows:

$$\text{Max} = (\text{FiniteDelay } [j])$$

such that for some index, j , where $1 \leq j \leq N$
 $\text{FiniteDelay}[j] \geq \text{FiniteDelay}[n]$ for all n

Max The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

8.4.2.5. Std_Dev

The Std_Dev SHALL be calculated using the conditional distribution of all packets with a finite value of One-way delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.3.2 of [RFC6049] for a closely related method for calculating this statistic, and 4.3.3 of [RFC6049]. The formula is the classic calculation for standard deviation of a population.

Define Population Std_Dev_Delay as follows:
 (where all packets $n = 1$ through N have a value for Delay[n],
 and MeanDelay calculated as in 7.4.2.2), and SQRT[] is the
 Square Root function:

$$\text{Std_Dev} = \text{SQRT} \left[\frac{1}{(N)} \sum_{n=1}^N (\text{Delay}[n] - \text{MeanDelay})^2 \right]$$

Std_Dev The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

8.4.3. Metric Units

The <statistic> of One-way Delay is expressed in seconds, where <statistic> is one of:

- o 95Percentile
- o Mean
- o Min
- o Max
- o StdDev

The One-way Loss Ratio is expressed as a percentage of lost packets to total packets sent.

8.4.4. Calibration

Section 3.7.3 of [RFC7679] provides a means to quantify the systematic and random errors of a time measurement. In-situ calibration could be enabled with an internal loopback that includes as much of the measurement system as possible, performs address manipulation as needed, and provides some form of isolation (e.g., deterministic delay) to avoid send-receive interface contention. Some portion of the random and systematic error can be characterized this way.

For one-way delay measurements, the error calibration must include an assessment of the internal clock synchronization with its external reference (this internal clock is supplying timestamps for measurement). In practice, the time offsets [RFC5905] of clocks at both the source and destination are needed to estimate the systematic error due to imperfect clock synchronization (the time offsets [RFC5905] are smoothed, thus the random variation is not usually represented in the results).

`time_offset` The time value of the result is expressed in units of seconds, as a signed value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

When a measurement controller requests a calibration measurement, the loopback is applied and the result is output in the same format as a normal measurement with additional indication that it is a calibration result. In any measurement, the measurement function SHOULD report its current estimate of time offset [RFC5905] as an indicator of the degree of synchronization.

Both internal loopback calibration and clock synchronization can be used to estimate the available accuracy of the Output Metric Units. For example, repeated loopback delay measurements will reveal the portion of the Output result resolution which is the result of system noise, and thus inaccurate.

8.5. Administrative items

8.5.1. Status

Current

8.5.2. Requester

This RFC number

8.5.3. Revision

1.0

8.5.4. Revision Date

YYYY-MM-DD

8.6. Comments and Remarks

None.

9. ICMP Round-trip Latency and Loss Registry Entries

This section specifies three initial registry entries for the ICMP Round-trip Latency, and another entry for ICMP Round-trip Loss Ratio.

IANA Note: Registry "Name" below specifies multiple registry entries, whose output format varies according to the <statistic> element of the name that specifies one form of statistical summary. There is an additional metric name for the Loss metric.

All column entries beside the ID, Name, Description, and Output Reference Method categories are the same, thus this section proposes two closely-related registry entries. As a result, IANA is also asked to assign corresponding URLs to each Named Metric.

9.1. Summary

This category includes multiple indexes to the registry entry: the element ID and metric name.

9.1.1. ID (Identifier)

IANA is asked to assign different numeric identifiers to each of the four Named Metrics.

9.1.2. Name

RTDelay_Active_IP-ICMP-SendOnRcv_RFCXXXXsec9_Seconds_<statistic>

where <statistic> is one of:

- o Mean
- o Min
- o Max

RTLoss_Active_IP-ICMP-SendOnRcv_RFCXXXXsec9_Percent_LossRatio

9.1.3. URI

URL: <https://www.iana.org/> ... <name>

9.1.4. Description

RTDelay: This metric assesses the delay of a stream of ICMP packets exchanged between two hosts (which are the two measurement points), and the Output is the Round-trip delay for all successfully exchanged packets expressed as the <statistic> of their conditional delay distribution, where <statistic> is one of:

- o Mean
- o Min
- o Max

RTLoss: This metric assesses the loss ratio of a stream of ICMP packets exchanged between two hosts (which are the two measurement points), and the Output is the Round-trip loss ratio for all successfully exchanged packets expressed as a percentage.

9.1.5. Change Controller

IETF

9.1.6. Version (of Registry Format)

1.0

9.2. Metric Definition

This category includes columns to prompt the entry of all necessary details related to the metric definition, including the RFC reference and values of input factors, called fixed parameters.

9.2.1. Reference Definition

Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, September 1999.

[RFC2681]

Section 2.4 of [RFC2681] provides the reference definition of the singleton (single value) Round-trip delay metric. Section 3.4 of [RFC2681] provides the reference definition expanded to cover a multi-singleton sample. Note that terms such as singleton and sample are defined in Section 11 of [RFC2330].

Note that although the [RFC2681] definition of "Round-trip-Delay between Src and Dst" is directionally ambiguous in the text, this

metric tightens the definition further to recognize that the host in the "Src" role will send the first packet to "Dst", and ultimately receive the corresponding return packet from "Dst" (when neither are lost).

Finally, note that the variable "dT" is used in [RFC2681] to refer to the value of Round-trip delay in metric definitions and methods. The variable "dT" has been re-used in other IPPM literature to refer to different quantities, and cannot be used as a global variable name.

Morton, A., "Round-trip Packet Loss Metrics", RFC 6673, August 2012.

[RFC6673]

Both delay and loss metrics employ a maximum waiting time for received packets, so the count of lost packets to total packets sent is the basis for the loss ratio calculation as per Section 6.1 of [RFC6673].

9.2.2. Fixed Parameters

Type-P as defined in Section 13 of [RFC2330]:

- o IPv4 header values:

- * DSCP: set to 0
- * TTL: set to 255
- * Protocol: Set to 01 (ICMP)

- o IPv6 header values:

- * DSCP: set to 0
- * Hop Count: set to 255
- * Next Header: set to 128 decimal (ICMP)
- * Flow Label: set to zero
- * Extension Headers: none

- o ICMP header values:

- * Type: 8 (Echo Request)
- * Code: 0

- * Checksum: the checksum MUST be calculated and the non-zero checksum included in the header
- * (Identifier and Sequence Number set at Run-Time)
- o ICMP Payload
 - * total of 32 bytes of random info, constant per test.

Other measurement parameters:

- o Tmax: a loss threshold waiting time
 - * 3.0, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms), with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

9.3. Method of Measurement

This category includes columns for references to relevant sections of the RFC(s) and any supplemental information needed to ensure an unambiguous methods for implementations.

9.3.1. Reference Method

The methodology for this metric is defined as Type-P-Round-trip-Delay-Poisson-Stream in section 2.6 of RFC 2681 [RFC2681] and section 3.6 of RFC 2681 [RFC2681] using the Type-P and Tmax defined under Fixed Parameters.

The reference method distinguishes between long-delayed packets and lost packets by implementing a maximum waiting time for packet arrival. Tmax is the waiting time used as the threshold to declare a packet lost. Lost packets SHALL be designated as having undefined delay, and counted for the RTLoss metric.

The calculations on the delay (RTD) SHALL be performed on the conditional distribution, conditioned on successful packet arrival within Tmax. Also, when all packet delays are stored, the process which calculates the RTD value MUST enforce the Tmax threshold on stored values before calculations. See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

The reference method requires some way to distinguish between different packets in a stream to establish correspondence between sending times and receiving times for each successfully-arriving packet. Sequence numbers or other send-order identification MUST be retained at the Src or included with each packet to disambiguate packet reordering if it occurs.

The measurement process will determine the sequence numbers applied to test packets after the Fixed and Runtime parameters are passed to that process. The ICMP measurement process and protocol will dictate the format of sequence numbers and other identifiers.

Refer to Section 4.4 of [RFC6673] for expanded discussion of the instruction to "send a Type-P packet back to the Src as quickly as possible" in Section 2.6 of RFC 2681 [RFC2681]. Section 8 of [RFC6673] presents additional requirements which MUST be included in the method of measurement for this metric.

9.3.2. Packet Stream Generation

This section gives the details of the packet traffic which is the basis for measurement. In IPPM metrics, this is called the Stream, and can easily be described by providing the list of stream parameters.

The ICMP metrics use a sending discipline called "SendOnRcv" or Send On Receive. This is a modification of Section 3 of [RFC3432], which prescribes the method for generating Periodic streams using associated parameters as defined below for this description:

incT the nominal duration of inter-packet interval, first bit to first bit

dT the duration of the interval for allowed sample start times

The incT stream parameter will be specified as a Run-time parameter, and dT is not used in SendOnRcv.

A SendOnRcv sender behaves exactly like a Periodic stream generator while all reply packets arrive with $RTD < incT$, and the inter-packet interval will be constant.

If a reply packet arrives with $RTD \geq incT$, then the inter-packet interval for the next sending time is nominally RTD.

If a reply packet fails to arrive within Tmax, then the inter-packet interval for the next sending time is nominally Tmax.

If an immediate send on reply arrival is desired, then set incT=0.

9.3.3. Traffic Filtering (observation) Details

NA

9.3.4. Sampling Distribution

NA

9.3.5. Run-time Parameters and Data Format

Run-time Parameters are input factors that must be determined, configured into the measurement system, and reported with the results for the context to be complete.

Src the IP address of the host in the Src Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see Section 4 of [RFC6991])

Dst the IP address of the host in the Dst Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see section 4 of [RFC6991])

incT the nominal duration of inter-packet interval, first bit to first bit, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms).

T0 a time, the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a start time is unspecified and Tf is to be interpreted as the Duration of the measurement interval. The start time is controlled through other means.

Count The total count of ICMP Echo Requests to send, formatted as a uint16, as per section 9.2 of [RFC6020].

(see the Packet Stream Generation section for additional Run-time parameters)

9.3.6. Roles

Src launches each packet and waits for return transmissions from Dst.

Dst waits for each packet from Src and sends a return packet to Src.

9.4. Output

This category specifies all details of the Output of measurements using the metric.

9.4.1. Type

See subsection titles in Reference Definition for Latency Types.

LossRatio -- the count of lost packets to total packets sent is the basis for the loss ratio calculation as per Section 6.1 of [RFC6673].

9.4.2. Reference Definition

For all output types ---

T0 the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

Tf the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

TotalCount the count of packets actually sent by the Src to Dst during the measurement interval.

For LossRatio -- the count of lost packets to total packets sent is the basis for the loss ratio calculation as per Section 4.1 of [RFC7680].

For each <statistic>, one of the following sub-sections apply:

9.4.2.1. Mean

The mean SHALL be calculated using the conditional distribution of all packets with a finite value of Round-trip delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.2.2 of [RFC6049] for details on calculating this statistic, and 4.2.3 of [RFC6049].

Mean The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

9.4.2.2. Min

The minimum SHALL be calculated using the conditional distribution of all packets with a finite value of Round-trip delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.3.2 of [RFC6049] for details on calculating this statistic, and 4.3.3 of [RFC6049].

Min The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

9.4.2.3. Max

The maximum SHALL be calculated using the conditional distribution of all packets with a finite value of Round-trip delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.3.2 of [RFC6049] for a closely related method for calculating this statistic, and 4.3.3 of [RFC6049]. The formula is as follows:

$$\text{Max} = (\text{FiniteDelay } [j])$$

such that for some index, j , where $1 \leq j \leq N$
 $\text{FiniteDelay}[j] \geq \text{FiniteDelay}[n]$ for all n

Max The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001

seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

9.4.3. Metric Units

The <statistic> of Round-trip Delay is expressed in seconds, where <statistic> is one of:

- o Mean
- o Min
- o Max

The Round-trip Loss Ratio is expressed as a percentage of lost packets to total packets sent.

9.4.4. Calibration

Section 3.7.3 of [RFC7679] provides a means to quantify the systematic and random errors of a time measurement. In-situ calibration could be enabled with an internal loopback at the Source host that includes as much of the measurement system as possible, performs address manipulation as needed, and provides some form of isolation (e.g., deterministic delay) to avoid send-receive interface contention. Some portion of the random and systematic error can be characterized this way.

When a measurement controller requests a calibration measurement, the loopback is applied and the result is output in the same format as a normal measurement with additional indication that it is a calibration result.

Both internal loopback calibration and clock synchronization can be used to estimate the available accuracy of the Output Metric Units. For example, repeated loopback delay measurements will reveal the portion of the Output result resolution which is the result of system noise, and thus inaccurate.

9.5. Administrative items

9.5.1. Status

Current

9.5.2. Requester

This RFC number

9.5.3. Revision

1.0

9.5.4. Revision Date

YYYY-MM-DD

9.6. Comments and Remarks

None

10. TCP Round-Trip Delay and Loss Registry Entries

This section specifies three initial registry entries for the Passive assessment of TCP Round-Trip Delay (RTD) and another entry for TCP Round-trip Loss Count.

IANA Note: Registry "Name" below specifies multiple registry entries, whose output format varies according to the <statistic> element of the name that specifies one form of statistical summary. There are two additional metric names for Singleton RT Delay and Packet Count metrics.

All column entries beside the ID, Name, Description, and Output Reference Method categories are the same, thus this section proposes four closely-related registry entries. As a result, IANA is also asked to assign corresponding URLs to each Named Metric.

10.1. Summary

This category includes multiple indexes to the registry entry: the element ID and metric name.

10.1.1. ID (Identifier)

IANA is asked to assign different numeric identifiers to each of the four Named Metrics.

10.1.2. Name

RTDelay_Passive_IP-TCP_RFCXXXXsec10_Seconds_<statistic>

where <statistic> is one of:

- o Mean
- o Min
- o Max

RTDelay_Passive_IP-TCP-HS_RFCXXXXsec10_Seconds_Singleton

Note that a mid-point observer only has the opportunity to compose a single RTDelay on the TCP Hand Shake.

RTLoss_Passive_IP-TCP_RFCXXXXsec10_Packet_Count

10.1.3. URI

URL: <https://www.iana.org/> ... <name>

10.1.4. Description

RTDelay: This metric assesses the round-trip delay of TCP packets constituting a single connection, exchanged between two hosts. We consider the measurement of round-trip delay based on a single Observation Point [RFC7011] somewhere in the network. The Output is the Round-trip delay for all successfully exchanged packets expressed as the <statistic> of their conditional delay distribution, where <statistic> is one of:

- o Mean
- o Min
- o Max

RTLoss: This metric assesses the estimated loss count for TCP packets constituting a single connection, exchanged between two hosts. We consider the measurement of round-trip delay based on a single Observation Point [RFC7011] somewhere in the network. The Output is the estimated Loss Count for the measurement interval.

10.1.5. Change Controller

IETF

10.1.6. Version (of Registry Format)

1.0

10.2. Metric Definition

This category includes columns to prompt the entry of all necessary details related to the metric definition, including the RFC reference and values of input factors, called fixed parameters.

10.2.1. Reference Definitions

Although there is no RFC that describes passive measurement of Round-Trip Delay, the parallel definition for Active measurement is:

Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, September 1999.

[RFC2681]

This metric definition uses the terms singleton and sample as defined in Section 11 of [RFC2330]. (Section 2.4 of [RFC2681] provides the reference definition of the singleton (single value) Round-trip delay metric. Section 3.4 of [RFC2681] provides the reference definition expanded to cover a multi-singleton sample.)

With the Observation Point [RFC7011] (OP) typically located between the hosts participating in the TCP connection, the Round-trip Delay metric requires two individual measurements between the OP and each host, such that the Spatial Composition [RFC6049] of the measurements yields a Round-trip Delay singleton (we are extending the composition of one-way subpath delays to subpath round-trip delay).

Using the direction of TCP SYN transmission to anchor the nomenclature, host A sends the SYN and host B replies with SYN-ACK during connection establishment. The direction of SYN transfer is considered the Forward direction of transmission, from A through OP to B (Reverse is B through OP to A).

Traffic filters reduce the packet stream at the OP to a Qualified bidirectional flow of packets.

In the definitions below, Corresponding Packets are transferred in different directions and convey a common value in a TCP header field that establishes correspondence (to the extent possible). Examples may be found in the TCP timestamp fields.

For a real number, RTD_{fwd} , \gg the Round-trip Delay in the Forward direction from OP to host B at time T' is RTD_{fwd} \ll it is REQUIRED that OP observed a Qualified Packet to host B at wire-time T' , that host B received that packet and sent a Corresponding Packet back to

host A, and OP observed the Corresponding Packet at wire-time $T' + \text{RTD_fwd}$.

For a real number, RTD_rev , \gg the Round-trip Delay in the Reverse direction from OP to host A at time T'' is $\text{RTD_rev} \ll$ it is REQUIRED that OP observed a Qualified Packet to host A at wire-time T'' , that host A received that packet and sent a Corresponding Packet back to host B, and that OP observed the Corresponding Packet at wire-time $T'' + \text{RTD_rev}$.

Ideally, the packet sent from host B to host A in both definitions above SHOULD be the same packet (or, when measuring RTD_rev first, the packet from host A to host B in both definitions should be the same).

The REQUIRED Composition Function for a singleton of Round-trip Delay at time T (where T is the earliest of T' and T'' above) is:

$$\text{RTDelay} = \text{RTD_fwd} + \text{RTD_rev}$$

Note that when OP is located at host A or host B, one of the terms composing RTDelay will be zero or negligible.

When the Qualified and Corresponding Packets are a TCP-SYN and a TCP-SYN-ACK, then $\text{RTD_fwd} == \text{RTD_HS_fwd}$.

When the Qualified and Corresponding Packets are a TCP-SYN-ACK and a TCP-ACK, then $\text{RTD_rev} == \text{RTD_HS_rev}$.

The REQUIRED Composition Function for a singleton of Round-trip Delay for the connection Hand Shake:

$$\text{RTDelay_HS} = \text{RTD_HS_fwd} + \text{RTD_HS_rev}$$

The definition of Round-trip Loss Count uses the nomenclature developed above, based on observation of the TCP header sequence numbers and storing the sequence number gaps observed. Packet Losses can be inferred from:

- o Out-of-order segments: TCP segments are transmitted with monotonically increasing sequence numbers, but these segments may be received out of order. Section 3 of [RFC4737] describes the notion of "next expected" sequence numbers which can be adapted to TCP segments (for the purpose of detecting reordered packets). Observation of out-of-order segments indicates loss on the path prior to the OP, and creates a gap.

- o Duplicate segments: Section 2 of [RFC5560] defines identical packets and is suitable for evaluation of TCP packets to detect duplication. Observation of duplicate segments *without a corresponding gap* indicates loss on the path following the OP (because they overlap part of the delivered sequence numbers already observed at OP).

Each observation of an out-of-order or duplicate infers a singleton of loss, but composition of Round-trip Loss Counts will be conducted over a measurement interval which is synonymous with a single TCP connection.

With the above observations in the Forward direction over a measurement interval, the count of out-of-order and duplicate segments is defined as RTL_fwd. Comparable observations in the Reverse direction are defined as RTL_rev.

For a measurement interval (corresponding to a single TCP connection), T0 to Tf, the REQUIRED Composition Function for a the two single-direction counts of inferred loss is:

$RTL_{Loss} = RTL_{fwd} + RTL_{rev}$

10.2.2. Fixed Parameters

Traffic Filters:

- o IPv4 header values:
 - * DSCP: set to 0
 - * Protocol: Set to 06 (TCP)
- o IPv6 header values:
 - * DSCP: set to 0
 - * Hop Count: set to 255
 - * Next Header: set to 6 (TCP)
 - * Flow Label: set to zero
 - * Extension Headers: none
- o TCP header values:
 - * Flags: ACK, SYN, FIN, set as required

- * Timestamp Option (TSopt): Set

- + Section 3.2 of [RFC7323]

10.3. Method of Measurement

This category includes columns for references to relevant sections of the RFC(s) and any supplemental information needed to ensure an unambiguous methods for implementations.

10.3.1. Reference Methods

The foundation methodology for this metric is defined in Section 4 of [RFC7323] using the Timestamp Option with modifications that allow application at a mid-path Observation Point (OP) [RFC7011]. Further details and applicable heuristics were derived from [Strowes] and [Trammell-14].

The Traffic Filter at the OP is configured to observe a single TCP connection. When the SYN, SYN-ACK, ACK handshake occurs, it offers the first opportunity to measure both RTD_fwd (on the SYN to SYN-ACK pair) and RTD_rev (on the SYN-ACK to ACK pair). Label this singleton of RTDelay as RTDelay_HS (composed using the forward and reverse measurement pair). RTDelay_HS SHALL be treated separately from other RTDelays on data-bearing packets and their ACKs. The RTDelay_HS value MAY be used as a sanity check on other Composed values of RTDelay.

For payload bearing packets, the OP measures the time interval between observation of a packet with Sequence Number *s*, and the corresponding ACK with same Sequence number. When the payload is transferred from host A to host B, the observed interval is RTD_fwd.

Because many data transfers are unidirectional (say, in the Forward direction from host A to host B), it is necessary to use pure ACK packets with Timestamp (TSval) and their Timestamp value echo to perform a RTD_rev measurement. The time interval between observation of the ACK from B to A, and the corresponding packet with Timestamp echo (TSecr) is the RTD_rev.

Delay Measurement Filtering Heuristics:

If Data payloads were transferred in both Forward and Reverse directions, then the Round-Trip Time Measurement Rule in Section 4.1 of [RFC7323] could be applied. This rule essentially excludes any measurement using a packet unless it makes progress in the transfer (advances the left edge of the send window, consistent with [Strowes]).

A different heuristic from [Trammell-14] is to exclude any `RTD_rev` that is larger than previously observed values. This would tend to exclude Reverse measurements taken when the Application has no data ready to send, because considerable time could be added to `RTD_rev` from this source of error.

Note that the above Heuristic assumes that host A is sending data. Host A expecting a download would mean that this heuristic should be applied to `RTD_fwd`.

The statistic calculations to summarize the delay (`RTDelay`) SHALL be performed on the conditional distribution, conditioned on successful Forward and Reverse measurements which follow the Heuristics.

Method for Inferring Loss:

The OP tracks sequence numbers and stores gaps for each direction of transmission, as well as the next-expected sequence number as in [Trammell-14] and [RFC4737]. Loss is inferred from Out-of-order segments and Duplicate segments.

Loss Measurement Filtering Heuristics:

[Trammell-14] adds a window of evaluation based on the `RTDelay`.

Distinguish Re-ordered from OOO due to loss, because sequence number gap is filled during the same `RTDelay` window. Segments detected as re-ordered according to [RFC4737] MUST reduce the Loss Count inferred from Out-of-order segments.

Spurious (unneeded) retransmissions (observed as duplicates) can also be reduced this way, as described in [Trammell-14].

Sources of Error:

The principal source of `RTDelay` error is the host processing time to return a packet that defines the termination of a time interval. The heuristics above intend to mitigate these errors by excluding measurements where host processing time is a significant part of `RTD_fwd` or `RTD_rev`.

A key source of `RTLoss` error is observation loss, described in section 3 of [Trammell-14].

10.3.2. Packet Stream Generation

NA

10.3.3. Traffic Filtering (observation) Details

The Fixed Parameters above give a portion of the Traffic Filter. Other aspects will be supplied as Run-time Parameters (below).

10.3.4. Sampling Distribution

This metric requires a complete sample of all packets that qualify according to the Traffic Filter criteria.

10.3.5. Run-time Parameters and Data Format

Run-time Parameters are input factors that must be determined, configured into the measurement system, and reported with the results for the context to be complete.

Src the IP address of the host in the host A Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see Section 4 of [RFC6991])

Dst the IP address of the host in the host B (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see section 4 of [RFC6991])

T0 a time, the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a start time is unspecified and Td is to be interpreted as the Duration of the measurement interval. The start time is controlled through other means.

Td Optionally, the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]), or the duration (see T0). The UTC Time Zone is required by Section 6.1 of [RFC2330]. Alternatively, the end of the measurement interval MAY be controlled by the measured connection, where the second pair of FIN and ACK packets exchanged between host A and B effectively ends the interval.

TTL or Hop Limit Set at desired value.

10.3.6. Roles

host A launches the SYN packet to open the connection, and synonymous with an IP address.

host B replies with the SYN-ACK packet to open the connection, and synonymous with an IP address.

10.4. Output

This category specifies all details of the Output of measurements using the metric.

10.4.1. Type

See subsection titles in Reference Definition for RTDelay Types.

For RTLoss -- the count of lost packets.

10.4.2. Reference Definition

For all output types ---

T0 the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

Tf the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. The end of the measurement interval MAY be controlled by the measured connection, where the second pair of FIN and ACK packets exchanged between host A and B effectively ends the interval.

... ..

For RTDelay_HS -- the Round trip delay of the Handshake.

For RTLoss -- the count of lost packets.

For each <statistic>, one of the following sub-sections apply:

10.4.2.1. Mean

The mean SHALL be calculated using the conditional distribution of all packets with a finite value of Round-trip delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.2.2 of [RFC6049] for details on calculating this statistic, and 4.2.3 of [RFC6049].

Mean The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

10.4.2.2. Min

The minimum SHALL be calculated using the conditional distribution of all packets with a finite value of Round-trip delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.3.2 of [RFC6049] for details on calculating this statistic, and 4.3.3 of [RFC6049].

Min The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

10.4.2.3. Max

The maximum SHALL be calculated using the conditional distribution of all packets with a finite value of Round-trip delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.3.2 of [RFC6049] for a closely related method for calculating this statistic, and 4.3.3 of [RFC6049]. The formula is as follows:

$$\text{Max} = (\text{FiniteDelay}[j])$$

such that for some index, j , where $1 \leq j \leq N$
 $\text{FiniteDelay}[j] \geq \text{FiniteDelay}[n]$ for all n

Max The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

10.4.3. Metric Units

The <statistic> of Round-trip Delay is expressed in seconds, where <statistic> is one of:

- o Mean
- o Min
- o Max

The Round-trip Delay of the Hand Shake is expressed in seconds.

The Round-trip Loss Count is expressed as a number of packets.

10.4.4. Calibration

Passive measurements at an OP could be calibrated against an active measurement (with loss emulation) at host A or B, where the active measurement represents the ground-truth.

10.5. Administrative items

10.5.1. Status

Current

10.5.2. Requester

This RFC number

10.5.3. Revision

1.0

10.5.4. Revision Date

YYYY-MM-DD

10.6. Comments and Remarks

None.

11. Security Considerations

These registry entries represent no known implications for Internet Security. Each RFC referenced above contains a Security Considerations section. Further, the LMAP Framework [RFC7594] provides both security and privacy considerations for measurements.

There are potential privacy considerations for observed traffic, particularly for passive metrics in section 10. An attacker that knows that its TCP connection is being measured can modify its behavior to skew the measurement results.

12. IANA Considerations

IANA is requested to populate The Performance Metrics Registry defined in [I-D.ietf-ippm-metric-registry] with the values defined in sections 4 through 10.

See the IANA Considerations section of [I-D.ietf-ippm-metric-registry] for additional requests and considerations.

13. Acknowledgements

The authors thank Brian Trammell for suggesting the term "Run-time Parameters", which led to the distinction between run-time and fixed parameters implemented in this memo, for identifying the IPFIX metric with Flow Key as an example, for suggesting the Passive TCP RTD metric and supporting references, and for many other productive suggestions. Thanks to Peter Koch, who provided several useful suggestions for disambiguating successive DNS Queries in the DNS Response time metric.

The authors also acknowledge the constructive reviews and helpful suggestions from Barbara Stark, Juergen Schoenwaelder, Tim Carey, Yaakov Stein, and participants in the LMAP working group. Thanks to

Michelle Cotton for her early IANA reviews, and to Amanda Barber for answering questions related to the presentation of the registry and accessibility of the complete template via URL.

14. References

14.1. Normative References

- [I-D.ietf-ippm-metric-registry]
Bagnulo, M., Claise, B., Eardley, P., and A. Morton,
"Registry for Performance Metrics", Internet Draft (work
in progress) draft-ietf-ippm-metric-registry, 2019.
- [RFC1035] Mockapetris, P., "Domain names - implementation and
specification", STD 13, RFC 1035, DOI 10.17487/RFC1035,
November 1987, <<https://www.rfc-editor.org/info/rfc1035>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,
<<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis,
"Framework for IP Performance Metrics", RFC 2330,
DOI 10.17487/RFC2330, May 1998,
<<https://www.rfc-editor.org/info/rfc2330>>.
- [RFC2681] Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip
Delay Metric for IPPM", RFC 2681, DOI 10.17487/RFC2681,
September 1999, <<https://www.rfc-editor.org/info/rfc2681>>.
- [RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet:
Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002,
<<https://www.rfc-editor.org/info/rfc3339>>.
- [RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation
Metric for IP Performance Metrics (IPPM)", RFC 3393,
DOI 10.17487/RFC3393, November 2002,
<<https://www.rfc-editor.org/info/rfc3393>>.
- [RFC3432] Raisanen, V., Grotefeld, G., and A. Morton, "Network
performance measurement with periodic streams", RFC 3432,
DOI 10.17487/RFC3432, November 2002,
<<https://www.rfc-editor.org/info/rfc3432>>.

- [RFC4737] Morton, A., Ciavattone, L., Ramachandran, G., Shalunov, S., and J. Perser, "Packet Reordering Metrics", RFC 4737, DOI 10.17487/RFC4737, November 2006, <<https://www.rfc-editor.org/info/rfc4737>>.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, DOI 10.17487/RFC5357, October 2008, <<https://www.rfc-editor.org/info/rfc5357>>.
- [RFC5481] Morton, A. and B. Claise, "Packet Delay Variation Applicability Statement", RFC 5481, DOI 10.17487/RFC5481, March 2009, <<https://www.rfc-editor.org/info/rfc5481>>.
- [RFC5560] Uijterwaal, H., "A One-Way Packet Duplication Metric", RFC 5560, DOI 10.17487/RFC5560, May 2009, <<https://www.rfc-editor.org/info/rfc5560>>.
- [RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <<https://www.rfc-editor.org/info/rfc5905>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6049] Morton, A. and E. Stephan, "Spatial Composition of Metrics", RFC 6049, DOI 10.17487/RFC6049, January 2011, <<https://www.rfc-editor.org/info/rfc6049>>.
- [RFC6673] Morton, A., "Round-Trip Packet Loss Metrics", RFC 6673, DOI 10.17487/RFC6673, August 2012, <<https://www.rfc-editor.org/info/rfc6673>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <<https://www.rfc-editor.org/info/rfc7011>>.

- [RFC7323] Borman, D., Braden, B., Jacobson, V., and R. Scheffenegger, Ed., "TCP Extensions for High Performance", RFC 7323, DOI 10.17487/RFC7323, September 2014, <<https://www.rfc-editor.org/info/rfc7323>>.
- [RFC7679] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <<https://www.rfc-editor.org/info/rfc7679>>.
- [RFC7680] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Loss Metric for IP Performance Metrics (IPPM)", STD 82, RFC 7680, DOI 10.17487/RFC7680, January 2016, <<https://www.rfc-editor.org/info/rfc7680>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [Strowes] Strowes, S., "Passively Measuring TCP Round Trip Times, Communications of the ACM, Vol. 56 No. 10, Pages 57-64", September 2013.
- [Trammell-14]
Trammell, B., "Inline Data Integrity Signals for Passive Measurement, In: Dainotti A., Mahanti A., Uhlig S. (eds) Traffic Monitoring and Analysis. TMA 2014. Lecture Notes in Computer Science, vol 8406. Springer, Berlin, Heidelberg https://link.springer.com/chapter/10.1007/978-3-642-54999-1_2", March 2014.

14.2. Informative References

- [RFC1242] Bradner, S., "Benchmarking Terminology for Network Interconnection Devices", RFC 1242, DOI 10.17487/RFC1242, July 1991, <<https://www.rfc-editor.org/info/rfc1242>>.
- [RFC6390] Clark, A. and B. Claise, "Guidelines for Considering New Performance Metric Development", BCP 170, RFC 6390, DOI 10.17487/RFC6390, October 2011, <<https://www.rfc-editor.org/info/rfc6390>>.
- [RFC6703] Morton, A., Ramachandran, G., and G. Maguluri, "Reporting IP Network Performance Metrics: Different Points of View", RFC 6703, DOI 10.17487/RFC6703, August 2012, <<https://www.rfc-editor.org/info/rfc6703>>.

[RFC7594] Eardley, P., Morton, A., Bagnulo, M., Burbridge, T.,
Aitken, P., and A. Akhter, "A Framework for Large-Scale
Measurement of Broadband Performance (LMAP)", RFC 7594,
DOI 10.17487/RFC7594, September 2015,
<<https://www.rfc-editor.org/info/rfc7594>>.

Authors' Addresses

Al Morton
AT&T Labs
200 Laurel Avenue South
Middletown,, NJ 07748
USA

Phone: +1 732 420 1571
Fax: +1 732 368 1192
Email: acmorton@att.com

Marcelo Bagnulo
Universidad Carlos III de
Madrid
Av. Universidad 30
Leganes, Madrid 28911
SPAIN

Phone: 34 91 6249500
Email: marcelo@it.uc3m.es
URI: <http://www.it.uc3m.es>

Philip Eardley
BT
Adastral Park, Martlesham Heath
Ipswich
ENGLAND

Email: philip.eardley@bt.com

Kevin D'Souza
AT&T Labs
200 Laurel Avenue South
Middletown,, NJ 07748
USA

Phone: +1 732 420 xxxx
Email: kld@att.com

ippm
Internet-Draft
Intended status: Standards Track
Expires: June 16, 2022

F. Brockners, Ed.
Cisco
S. Bhandari, Ed.
Thoughtspot
T. Mizrahi, Ed.
Huawei
December 13, 2021

Data Fields for In-situ OAM
draft-ietf-ippm-ioam-data-17

Abstract

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information in the packet while the packet traverses a path in the network. This document discusses the data fields and associated data types for in-situ OAM. In-situ OAM data fields can be encapsulated into a variety of protocols such as NSH, Segment Routing, Geneve, or IPv6. In-situ OAM can be used to complement OAM mechanisms based on, e.g., ICMP or other types of probe packets.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on June 16, 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Contributors	3
3. Conventions	4
4. Scope, Applicability, and Assumptions	5
5. IOAM Data-Fields, Types, Nodes	6
5.1. IOAM Data-Fields and Option-Types	7
5.2. IOAM-Domains and types of IOAM Nodes	7
5.3. IOAM-Namespaces	8
5.4. IOAM Trace Option-Types	11
5.4.1. Pre-allocated and Incremental Trace Option-Types . .	13
5.4.2. IOAM node data fields and associated formats	17
5.4.2.1. Hop_Lim and node_id short format	18
5.4.2.2. ingress_if_id and egress_if_id	19
5.4.2.3. timestamp seconds	19
5.4.2.4. timestamp fraction	20
5.4.2.5. transit delay	20
5.4.2.6. namespace specific data	20
5.4.2.7. queue depth	21
5.4.2.8. Checksum Complement	21
5.4.2.9. Hop_Lim and node_id wide	22
5.4.2.10. ingress_if_id and egress_if_id wide	22
5.4.2.11. namespace specific data wide	22
5.4.2.12. buffer occupancy	23
5.4.2.13. Opaque State Snapshot	23
5.4.3. Examples of IOAM node data	24
5.5. IOAM Proof of Transit Option-Type	26
5.5.1. IOAM Proof of Transit Type 0	28
5.6. IOAM Edge-to-Edge Option-Type	28
6. Timestamp Formats	31
6.1. PTP Truncated Timestamp Format	31
6.2. NTP 64-bit Timestamp Format	32
6.3. POSIX-based Timestamp Format	33
7. IOAM Data Export	34
8. IANA Considerations	35
8.1. IOAM Option-Type Registry	35
8.2. IOAM Trace-Type Registry	36
8.3. IOAM Trace-Flags Registry	37
8.4. IOAM POT-Type Registry	37
8.5. IOAM POT-Flags Registry	38

8.6. IOAM E2E-Type Registry	38
8.7. IOAM Namespace-ID Registry	39
9. Management and Deployment Considerations	40
10. Security Considerations	40
11. Acknowledgements	43
12. References	43
12.1. Normative References	43
12.2. Informative References	44
Contributors' Addresses	45
Authors' Addresses	47

1. Introduction

This document defines data fields for "in-situ" Operations, Administration, and Maintenance (IOAM). In-situ OAM records OAM information within the packet while the packet traverses a particular network domain. The term "in-situ" refers to the fact that the OAM data is added to the data packets rather than being sent within packets specifically dedicated to OAM. IOAM is to complement mechanisms such as Ping or Traceroute. In terms of "active" or "passive" OAM, "in-situ" OAM can be considered a hybrid OAM type. "In-situ" mechanisms do not require extra packets to be sent. IOAM adds information to the already available data packets and therefore cannot be considered passive. In terms of the classification given in [RFC7799], IOAM could be portrayed as Hybrid Type I. IOAM mechanisms can be leveraged where mechanisms using, e.g., ICMP do not apply or do not offer the desired results, such as proving that a certain traffic flow takes a pre-defined path, SLA verification for the data traffic, detailed statistics on traffic distribution paths in networks that distribute traffic across multiple paths, or scenarios in which probe traffic is potentially handled differently from regular data traffic by the network devices.

The term "in situ OAM" was originally motivated by the use of OAM related mechanisms that add information into a packet. This document uses IOAM as a term defining the IOAM technology. IOAM includes "in-situ" mechanisms, but also mechanisms that could trigger the creation of additional packets dedicated to OAM.

2. Contributors

This document was the collective effort of several authors. The text and content were contributed by the editors and the co-authors listed below. The contact information of the co-authors appears at the end of this document.

- o Carlos Pignataro

- o Mickey Spiegel
- o Barak Gafni
- o Jennifer Lemon
- o Hannes Gredler
- o John Leddy
- o Stephen Youell
- o David Mozes
- o Petr Lapukhov
- o Remy Chang
- o Daniel Bernier

3. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Abbreviations and definitions used in this document:

E2E: Edge to Edge

Geneve: Generic Network Virtualization Encapsulation [RFC8926]

IOAM: In-situ Operations, Administration, and Maintenance

MTU: Maximum Transmit Unit

NSH: Network Service Header [RFC8300]

OAM: Operations, Administration, and Maintenance

PMTU: Path MTU

POT: Proof of Transit

Short format: "Short format" refers to an IOAM-Data-Field which comprises 4 octets.

SID: Segment Identifier

SR: Segment Routing

VXLAN-GPE: Virtual eXtensible Local Area Network, Generic Protocol Extension [I-D.ietf-nvo3-vxlan-gpe]

Wide format: "Wide format" refers to an IOAM-Data-Field which comprises 8 octets.

4. Scope, Applicability, and Assumptions

IOAM assumes a set of constraints as well as guiding principles and concepts that go hand in hand with the definition of the IOAM data fields. These constraints, guiding principles, and concepts are described in this section. A discussion of how IOAM data fields and the associated concepts are applied to an IOAM deployment are out of scope for this document. Please refer to [I-D.ietf-ippm-ioam-deployment] for IOAM deployment considerations.

Scope: This document defines the data fields and associated data types for in-situ OAM. The in-situ OAM data fields can be encapsulated in a variety of protocols, including NSH, Segment Routing, Geneve, and IPv6. Specification details for these different protocols are outside the scope of this document. It is expected that each such encapsulation would be specified by an RFC, jointly designed by the working group that develops or maintains the encapsulation protocol and the IETF IPPM working group.

Deployment domain (or scope) of in-situ OAM deployment: IOAM is focused on "limited domains" as defined in [RFC8799]. For IOAM, a limited domain could for example be an enterprise campus using physical connections between devices or an overlay network using virtual connections / tunnels for connectivity between said devices. A limited domain which uses IOAM may constitute one or multiple "IOAM-domains", each disambiguated through separate namespace identifiers. An IOAM-domain is bounded by its perimeter or edge. IOAM-domains may overlap inside the limited domain. Designers of protocol encapsulations for IOAM specify mechanisms to ensure that IOAM data stays within an IOAM-domain. In addition, the operator of such a domain is expected to put provisions in place to ensure that IOAM data does not leak beyond the edge of an IOAM-domain using, for example, packet filtering methods. The operator SHOULD consider the potential operational impact of IOAM to mechanisms such as ECMP processing (e.g., load-balancing schemes based on packet length could be impacted by the increased packet size due to IOAM), path MTU (i.e., ensure that the MTU of all links within a domain is sufficiently large to support the increased packet size due to IOAM)

and ICMP message handling (i.e., in case of IPv6, IOAM support for ICMPv6 Echo Request/Reply is desired which would translate into ICMPv6 extensions to enable IOAM-Data-Fields to be copied from an Echo Request message to an Echo Reply message).

IOAM control points: IOAM-Data-Fields are added to or removed from the user traffic by the devices which form the edge of a domain. Devices which form an IOAM-Domain can add, update or remove IOAM-Data-Fields. Edge devices of an IOAM-Domain can be hosts or network devices.

Traffic-sets that IOAM is applied to: IOAM can be deployed on all or only on subsets of the user traffic. Using IOAM on a selected set of traffic (e.g., per interface, based on an access control list or flow specification defining a specific set of traffic, etc.) could be useful in deployments where the cost of processing IOAM-Data-Fields by encapsulating, transit, or decapsulating node(s) might be a concern from a performance or operational perspective. Thus limiting the amount of traffic IOAM is applied to could be beneficial in some deployments.

Encapsulation independence: The definition of IOAM-Data-Fields is independent from the protocols the IOAM-Data-Fields are encapsulated into. IOAM-Data-Fields can be encapsulated into several encapsulating protocols.

Layering: If several encapsulation protocols (e.g., in case of tunneling) are stacked on top of each other, IOAM-Data-Fields could be present at multiple layers. The behavior follows the ships-in-the-night model, i.e., IOAM-Data-Fields in one layer are independent from IOAM-Data-Fields in another layer. Layering allows operators to instrument the protocol layer they want to measure. The different layers could, but do not have to, share the same IOAM encapsulation mechanisms.

IOAM implementation: The definition of the IOAM-Data-Fields take the specifics of devices with hardware data planes and software data planes into account.

5. IOAM Data-Fields, Types, Nodes

This section details IOAM-related nomenclature and describes data types such as IOAM-Data-Fields, IOAM-Types, IOAM-Namespaces as well as the different types of IOAM nodes.

5.1. IOAM Data-Fields and Option-Types

An IOAM-Data-Field is a set of bits with a defined format and meaning, which can be stored at a certain place in a packet for the purpose of IOAM.

To accommodate the different uses of IOAM, IOAM-Data-Fields fall into different categories. In IOAM, these categories are referred to as IOAM-Option-Types. A common registry is maintained for IOAM-Option-Types, see Section 8.1 for details. Corresponding to these IOAM-Option-Types, different IOAM-Data-Fields are defined.

This document defines four IOAM-Option-Types:

- o Pre-allocated Trace Option-Type
- o Incremental Trace Option-Type
- o Proof of Transit (POT) Option-Type
- o Edge-to-Edge (E2E) Option-Type

Future IOAM-Option-Types can be allocated by IANA, as described in Section 8.1.

5.2. IOAM-Domains and types of IOAM Nodes

Section 4 already mentioned that IOAM is expected to be deployed in a limited domain [RFC8799]. One or more IOAM-Option-Types are added to a packet upon entering an IOAM-Domain and are removed from the packet when exiting the domain. Within the IOAM-Domain, the IOAM-Data-Fields MAY be updated by network nodes that the packet traverses. An IOAM-Domain consists of "IOAM encapsulating nodes", "IOAM decapsulating nodes" and "IOAM transit nodes". The role of a node (i.e., encapsulating, transit, decapsulating) is defined within an IOAM-Namespace (see below). A node can have different roles in different IOAM-Namespace.

A device which adds at least one IOAM-Option-Type to the packet is called an "IOAM encapsulating node", whereas a device which removes an IOAM-Option-Type is referred to as an "IOAM decapsulating node". Nodes within the domain which are aware of IOAM data and read and/or write and/or process IOAM data are called "IOAM transit nodes". IOAM nodes which add or remove the IOAM-Data-Fields can also update the IOAM-Data-Fields at the same time. Or in other words, IOAM encapsulating or decapsulating nodes can also serve as IOAM transit nodes at the same time. Note that not every node in an IOAM-domain needs to be an IOAM transit node. For example, a deployment might

require that packets traverse a set of firewalls which support IOAM. In that case, only the set of firewall nodes would be IOAM transit nodes rather than all nodes.

An "IOAM encapsulating node" incorporates one or more IOAM-Option-Types (from the list of IOAM-Types, see Section 8.1) into packets that IOAM is enabled for. If IOAM is enabled for a selected subset of the traffic, the IOAM encapsulating node is responsible for applying the IOAM functionality to the selected subset.

An "IOAM transit node" reads and/or writes and/or processes one or more of the IOAM-Data-Fields. If both the Pre-allocated and the Incremental Trace Option-Types are present in the packet, each IOAM transit node based on configuration and available implementation of IOAM might populate IOAM trace data in either Pre-allocated or Incremental Trace Option-Type but not both. Note that not populating any of the Trace Option-Types is also valid behavior for an IOAM transit node. A transit node MUST ignore IOAM-Option-Types that it does not understand. A transit node MUST NOT add new IOAM-Option-Types to a packet, MUST NOT remove IOAM-Option-Types from a packet, and MUST NOT change the IOAM-Data-Fields of an IOAM Edge-to-Edge Option-Type.

An "IOAM decapsulating node" removes IOAM-Option-Type(s) from packets.

The role of an IOAM-encapsulating, IOAM-transit or IOAM-decapsulating node is always performed within a specific IOAM-Namespace. This means that an IOAM node which is, e.g., an IOAM-decapsulating node for IOAM-Namespace "A" but not for IOAM-Namespace "B" will only remove the IOAM-Option-Types for IOAM-Namespace "A" from the packet. Note that this applies even for IOAM-Option-Types that the node does not understand, for example an IOAM-Option-Type other than the four described above, that is added in a future revision.

IOAM-Namespace allow for a namespace-specific definition and interpretation of IOAM-Data-Fields. An interface-id could for example point to a physical interface (e.g., to understand which physical interface of an aggregated link is used when receiving or transmitting a packet) whereas in another case it could refer to a logical interface (e.g., in case of tunnels). Please refer to Section 5.3 for details on IOAM-Namespace.

5.3. IOAM-Namespace

IOAM-Namespace add further context to IOAM-Option-Types and associated IOAM-Data-Fields. The IOAM-Option-Types and associated IOAM-Data-Fields are interpreted as defined in this document,

regardless of the value of the IOAM-Namespace. However, IOAM-Namespaces provide a way to group nodes to support different deployment approaches of IOAM (see a few example use-cases below). IOAM-Namespaces also help to resolve potential issues which can occur due to IOAM-Data-Fields not being globally unique (e.g., IOAM node identifiers do not have to be globally unique). IOAM-Data-Fields significance is always within a particular IOAM-Namespace. Given that IOAM-Data-Fields are always interpreted the context of a specific namespace, the namespace-id field always needs to be carried along with the IOAM data-fields themselves.

An IOAM-Namespace is identified by a 16-bit namespace identifier (Namespace-ID). The IOAM-Namespace field is included in all the IOAM-Option-Types defined in this document, and MUST be included in all future IOAM-Option-Types. The Namespace-ID value is divided into two sub-ranges:

- o An operator-assigned range from 0x0001 to 0x7FFF
- o An IANA-assigned range from 0x8000 to 0xFFFF

The IANA-assigned range is intended to allow future extensions to have new and interoperable IOAM functionality, while the operator-assigned range is intended to be domain-specific, and managed by the network operator. The Namespace-ID value of 0x0000 is the "Default-Namespace-ID". The Default-Namespace-ID indicates that no specific namespace is associated with the IOAM data fields in the packet. The Default-Namespace-ID MUST be supported by all nodes implementing IOAM. A use-case for the Default-Namespace-ID are deployments which do not leverage specific namespaces for some or all of their packets that carry IOAM data fields.

Namespace identifiers allow devices which are IOAM capable to determine:

- o whether IOAM-Option-Type(s) need to be processed by a device: If the Namespace-ID contained in a packet does not match any Namespace-ID the node is configured to operate on, then the node MUST NOT change the contents of the IOAM-Data-Fields.
- o which IOAM-Option-Type needs to be processed/updated in case there are multiple IOAM-Option-Types present in the packet. Multiple IOAM-Option-Types can be present in a packet in case of overlapping IOAM-Domains or in case of a layered IOAM deployment.
- o whether IOAM-Option-Type(s) have to be removed from the packet, e.g., at a domain edge or domain boundary.

IOAM-Namespaces support several different uses:

- o IOAM-Namespaces can be used by an operator to distinguish different IOAM-domains. Devices at edges of an IOAM-domain can filter on Namespace-IDs to provide for proper IOAM-domain isolation.
- o IOAM-Namespaces provide additional context for IOAM-Data-Fields and thus can be used to ensure that IOAM-Data-Fields are unique and are interpreted properly by management stations or network controllers. The node identifier field (`node_id`, see below) does not need to be unique in a deployment. This could be the case if an operator wishes to use different node identifiers for different IOAM layers, even within the same device or node identifiers might not be unique for other organizational reasons, such as after a merger of two formerly separated organizations. The Namespace-ID can be used as a context identifier, such that the combination of `node_id` and Namespace-ID will always be unique.
- o Similarly, IOAM-Namespaces can be used to define how certain IOAM-Data-Fields are interpreted: IOAM offers three different timestamp format options. The Namespace-ID can be used to determine the timestamp format. IOAM-Data-Fields (e.g., buffer occupancy) which do not have a unit associated are to be interpreted within the context of a IOAM-Namespace.
- o IOAM-Namespaces can be used to identify different sets of devices (e.g., different types of devices) in a deployment: If an operator desires to insert different IOAM-Data-Fields based on the device, the devices could be grouped into multiple IOAM-Namespaces. This could be due to the fact that the IOAM feature set differs between different sets of devices, or it could be for reasons of optimized space usage in the packet header. It could also stem from hardware or operational limitations on the size of the trace data that can be added and processed, preventing collection of a full trace for a flow.
- o By assigning different IOAM Namespace-IDs to different sets of nodes or network partitions and using a separate instance of an IOAM-Option-Type for each Namespace-ID, a full trace for a flow could be collected and constructed via partial traces from each IOAM-Option-Type in each of the packets in the flow. Example: An operator could choose to group the devices of a domain into two IOAM-Namespaces, in a way that each IOAM-Namespace is represented by one of two IOAM-Option-Types in the packet. Each node would record data only for the IOAM-Namespace that it belongs to, ignoring the other IOAM-Option-Type with a IOAM-Namespace to which it doesn't belong. To retrieve a full view of the deployment, the

captured IOAM-Data-Fields of the two IOAM-Namespaces need to be correlated.

5.4. IOAM Trace Option-Types

In a typical deployment, all nodes in an IOAM-Domain would participate in IOAM and thus be IOAM transit nodes, IOAM encapsulating or IOAM decapsulating nodes. If not all nodes within a domain support IOAM functionality as defined in this document, IOAM tracing information (i.e., node data, see below) can only be collected on those nodes which support IOAM functionality as defined in this document. Nodes which do not support IOAM functionality as defined in this document will forward the packet without any changes to the IOAM-Data-Fields. The maximum number of hops and the minimum path MTU of the IOAM-domain is assumed to be known. An overflow indicator (O-bit) is defined as one of the ways to deal with situations where the PMTU was underestimated, i.e., where the number of hops which are IOAM capable exceeds the available space in the packet.

To optimize hardware and software implementations, IOAM tracing is defined as two separate options. A deployment can choose to configure and support one or both of the following options.

Pre-allocated Trace-Option: This trace option is defined as a container of node data fields (see below) with pre-allocated space for each node to populate its information. This option is useful for implementations where it is efficient to allocate the space once and index into the array to populate the data during transit (e.g., software forwarders often fall into this class). The IOAM encapsulating node allocates space for Pre-allocated Trace Option-Type in the packet and sets corresponding fields in this IOAM-Option-Type. The IOAM encapsulating node allocates an array which is used to store operational data retrieved from every node while the packet traverses the domain. IOAM transit nodes update the content of the array, and possibly update the checksums of outer headers. A pointer which is part of the IOAM trace data, points to the next empty slot in the array. An IOAM transit node that updates the content of the pre-allocated option also updates the value of the pointer, which specifies where the next IOAM transit node fills in its data. The "node data list" array (see below) in the packet is populated iteratively as the packet traverses the network, starting with the last entry of the array, i.e., "node data list [n]" is the first entry to be populated, "node data list [n-1]" is the second one, etc.

Incremental Trace-Option: This trace option is defined as a container of node data fields where each node allocates and pushes

its node data immediately following the option header. This type of trace recording is useful for some of the hardware implementations as it eliminates the need for the transit network elements to read the full array in the option and allows for arbitrarily long packets as the MTU allows. The IOAM encapsulating node allocates space for the Incremental Trace Option-Type. Based on operational state and configuration, the IOAM encapsulating node sets the fields in the Option-Type that control what IOAM-Data-Fields have to be collected and how large the node data list can grow. IOAM transit nodes push their node data to the node data list subject to any protocol constraints of the encapsulating layer. They then decrease the remaining length available to subsequent nodes and adjust the lengths and possibly checksums in outer headers.

IOAM encapsulating nodes and IOAM decapsulating nodes which support tracing MUST support both Trace-Option-Types. For IOAM transit nodes it is sufficient to support one of the Trace-Option-Types. In the event that both options are utilized in a deployment at the same time, the Incremental Trace-Option MUST be placed before the Pre-allocated Trace-Option. Deployments which mix devices with either the Incremental Trace-Option or the Pre-allocated Trace-Option could result in both Option-Types being present in a packet. Given that the operator knows which equipment is deployed in a particular IOAM-domain, the operator will decide by means of configuration which type(s) of trace options will be used for a particular domain.

Every node data entry holds information for a particular IOAM transit node that is traversed by a packet. The IOAM decapsulating node removes the IOAM-Option-Type(s) and processes and/or exports the associated data. Like all IOAM-Data-Fields, the IOAM-Data-Fields of the IOAM-Trace-Option-Types are defined in the context of an IOAM-Namespace.

IOAM tracing can collect the following types of information:

- o Identification of the IOAM node. An IOAM node identifier can match to a device identifier or a particular control point or subsystem within a device.
- o Identification of the interface that a packet was received on, i.e., ingress interface.
- o Identification of the interface that a packet was sent out on, i.e., egress interface.
- o Time of day when the packet was processed by the node as well as the transit delay. Different definitions of processing time are

feasible and expected, though it is important that all devices of an IOAM-domain follow the same definition.

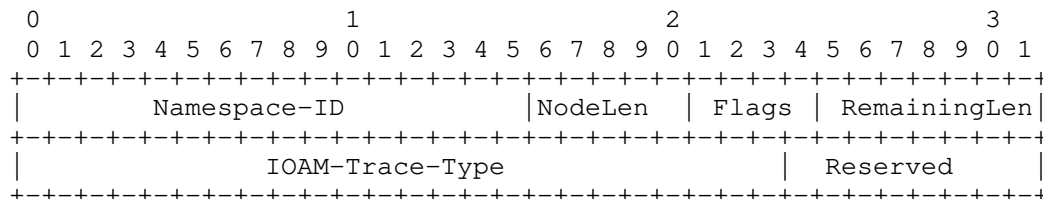
- o Generic data: Format-free information where syntax and semantic of the information is defined by the operator in a specific deployment. For a specific IOAM-Namespace, all IOAM nodes have to interpret the generic data the same way. Examples for generic IOAM data include geo-location information (location of the node at the time the packet was processed), buffer queue fill level or cache fill level at the time the packet was processed, or even a battery charge level.
- o Information to detect whether IOAM trace data was added at every hop or whether certain hops in the domain weren't IOAM transit nodes.

It should be noted that the semantics of some of the node data fields that are defined below, such as the queue depth and buffer occupancy, are implementation specific. This approach is intended to allow IOAM nodes with various different architectures.

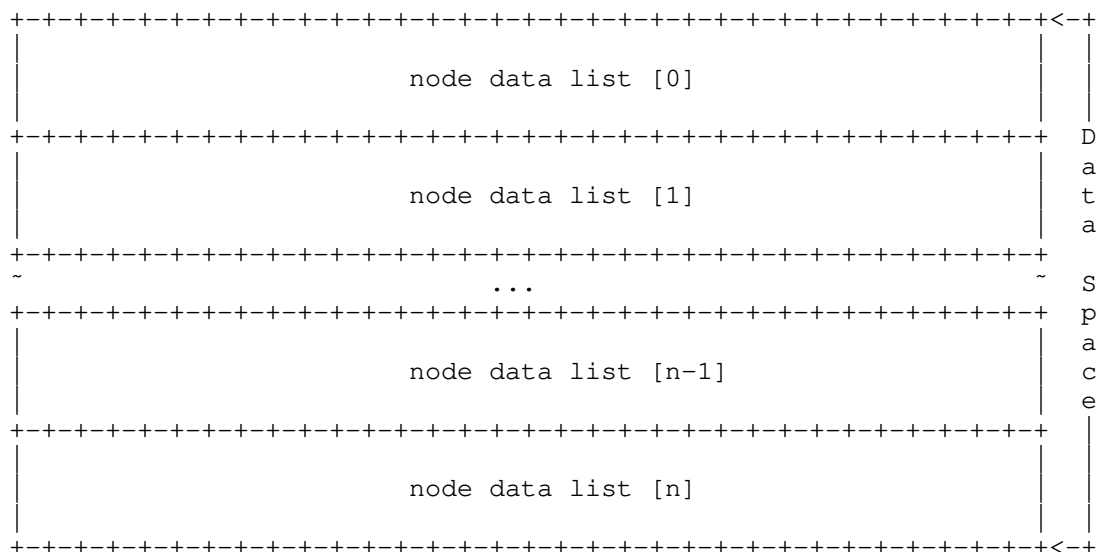
5.4.1. Pre-allocated and Incremental Trace Option-Types

The IOAM Pre-allocated Trace-Option and the IOAM Incremental Trace-Option have similar formats. Except where noted below, the internal formats and fields of the two trace options are identical. Both Trace-Options consist of a fixed size "trace option header" and a variable data space to store gathered data, the "node data list". An IOAM transit node (that is not an IOAM encapsulating node or IOAM decapsulating node) MUST NOT modify any of the fields in the fixed size "trace option header", other than "flags" and "RemainingLen", i.e., an IOAM transit node MUST NOT modify the Namespace-ID, NodeLen, IOAM-Trace-Type, or Reserved fields.

Pre-allocated and incremental trace option headers:



The trace option data MUST be 4-octet aligned:



Namespace-ID: 16-bit identifier of an IOAM-Namespace. The Namespace-ID value of 0x0000 is defined as the "Default-Namespace-ID" (see Section 5.3) and MUST be known to all the nodes implementing IOAM. For any other Namespace-ID value that does not match any Namespace-ID the node is configured to operate on, the node MUST NOT change the contents of the IOAM-Data-Fields.

NodeLen: 5-bit unsigned integer. This field specifies the length of data added by each node in multiples of 4-octets, excluding the length of the "Opaque State Snapshot" field.

If IOAM-Trace-Type bit 22 is not set, then NodeLen specifies the actual length added by each node. If IOAM-Trace-Type bit 22 is

set, then the actual length added by a node would be (NodeLen + length of the "Opaque State Snapshot" field) in 4 octet units.

For example, if 3 IOAM-Trace-Type bits are set and none of them are in wide format, then NodeLen would be 3. If 3 IOAM-Trace-Type bits are set and 2 of them are wide, then NodeLen would be 5.

An IOAM encapsulating node MUST set NodeLen.

A node receiving an IOAM Pre-allocated or Incremental Trace-Option relies on the NodeLen value.

Flags 4-bit field. Flags are allocated by IANA, as specified in Section 8.3. This document allocates a single flag as follows:

Bit 0 "Overflow" (O-bit) (most significant bit). In case a network element is supposed to add node data to a packet, but detects that there are not enough octets left to record the node data, the network element MUST NOT add any fields and MUST set the overflow "O-bit" to "1" in the IOAM-Trace-Option header. This is useful for transit nodes to ignore further processing of the option.

RemainingLen: 7-bit unsigned integer. This field specifies the data space in multiples of 4-octets remaining for recording the node data, before the node data list is considered to have overflowed. The sender MUST assign the initial value of the RemainingLen field. The sender MAY calculate the value of the RemainingLen field by computing the number of node data bytes allowed before exceeding the path MTU (PMTU), given that the PMTU is known to the sender. Subsequent nodes can carry out a simple comparison between RemainingLen and NodeLen, along with the length of the "Opaque State Snapshot" if applicable, to determine whether or not data can be added by this node. When node data is added, the node MUST decrease RemainingLen by the amount of data added. In the pre-allocated trace option, RemainingLen is used to derive the offset in data space to record the node data element. Specifically, the recording of the node data element would start from RemainingLen - NodeLen - sizeof(opaque snapshot) in 4 octet units. If RemainingLen in a pre-allocated trace option exceeds the length of the option, as specified in the lower layer header (which is not within the scope of this document), then the node MUST NOT add any fields.

IOAM-Trace-Type: A 24-bit identifier which specifies which data types are used in this node data list.

The IOAM-Trace-Type value is a bit field. The following bits are defined in this document, with details on each bit described in the Section 5.4.2. The order of packing the data fields in each node data element follows the bit order of the IOAM-Trace-Type field, as follows:

- Bit 0 (Most significant bit) When set, indicates presence of Hop_Lim and node_id (short format) in the node data.
- Bit 1 When set, indicates presence of ingress_if_id and egress_if_id (short format) in the node data.
- Bit 2 When set, indicates presence of timestamp seconds in the node data.
- Bit 3 When set, indicates presence of timestamp fraction in the node data.
- Bit 4 When set, indicates presence of transit delay in the node data.
- Bit 5 When set, indicates presence of IOAM-Namespace specific data (short format) in the node data.
- Bit 6 When set, indicates presence of queue depth in the node data.
- Bit 7 When set, indicates presence of the Checksum Complement node data.
- Bit 8 When set, indicates presence of Hop_Lim and node_id in wide format in the node data.
- Bit 9 When set, indicates presence of ingress_if_id and egress_if_id in wide format in the node data.
- Bit 10 When set, indicates presence of IOAM-Namespace specific data in wide format in the node data.
- Bit 11 When set, indicates presence of buffer occupancy in the node data.
- Bit 12-21 Undefined. These values are available for future assignment in the IOAM Trace-Type Registry (Section 8.2). Every future node data field corresponding to one of these bits MUST be 4-octets long. An IOAM encapsulating node MUST set the value of each undefined bit to 0. If

an IOAM transit node receives a packet with one or more of these bits set to 1, it MUST either:

1. Add corresponding node data filled with the reserved value 0xFFFFFFFF, after the node data fields for the IOAM-Trace-Type bits defined above, such that the total node data added by this node in units of 4-octets is equal to NodeLen, or
2. Not add any node data fields to the packet, even for the IOAM-Trace-Type bits defined above.

Bit 22 When set, indicates presence of variable length Opaque State Snapshot field.

Bit 23 Reserved: MUST be set to zero upon transmission and ignored upon receipt. This bit is reserved to allow for future extensions of the IOAM-Trace-Type bit field.

Section 5.4.2 describes the IOAM-Data-Types and their formats. Within an IOAM-Domain possible combinations of these bits making the IOAM-Trace-Type can be restricted by configuration knobs.

Reserved: 8-bits. An IOAM encapsulating node MUST set the value to zero upon transmission. IOAM transit nodes MUST ignore the received value.

Node data List [n]: Variable-length field. This is a list of node data elements where the content of each node data element is determined by the IOAM-Trace-Type. The order of packing the data fields in each node data element follows the bit order of the IOAM-Trace-Type field. Each node MUST prepend its node data element in front of the node data elements that it received, such that the transmitted node data list begins with this node's data element as the first populated element in the list. The last node data element in this list is the node data of the first IOAM capable node in the path. Populating the node data list in this way ensures that the order of node data list is the same for incremental and pre-allocated trace options. In the pre-allocated trace option, the index contained in RemainingLen identifies the offset for current active node data to be populated.

5.4.2. IOAM node data fields and associated formats

All the IOAM-Data-Fields MUST be 4-octet aligned. If a node which is supposed to update an IOAM-Data-Field is not capable of populating the value of a field set in the IOAM-Trace-Type, the field value MUST be set to 0xFFFFFFFF for 4-octet fields or 0xFFFFFFFFFFFFFFFF for

8-octet fields, indicating that the value is not populated, except when explicitly specified in the field description below.

Some IOAM-Data-Fields defined below, such as interface identifiers or IOAM-Namespace specific data, are defined in both "short format" as well as "wide format". The use of "short format" or "wide format" is not mutually exclusive. A deployment could choose to leverage both. For example, `ingress_if_id`(short format) could be an identifier for the physical interface, whereas `ingress_if_id`(wide format) could be an identifier for a logical sub-interface of that physical interface.

Data fields and associated data types for each of the IOAM-Data-Fields are specified in the following sections. The definition of IOAM-Data-Fields focuses on the syntax of the data-fields and avoids specifying the semantics where feasible. This is why no units are defined for data-fields like e.g., "buffer occupancy" or "queue depth". With this approach, nodes can supply the information in their native format and are not required to perform unit or format conversions. Systems that further process IOAM information, like e.g., a network management system are assumed to also handle unit conversions as part of their IOAM data-fields processing. The combination of a particular data-field and the namespace-id provides for the context to interpret the provided data appropriately.

5.4.2.1. Hop_Lim and node_id short format

The "Hop_Lim and node_id short format" field is a 4-octet field that is defined as follows:

```

 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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Hop_Lim | node_id |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Hop_Lim: 1-octet unsigned integer. It is set to the Hop Limit value in the packet at egress from the node that records this data. Hop Limit information is used to identify the location of the node in the communication path. This is copied from the lower layer, e.g., TTL value in IPv4 header or hop limit field from IPv6 header of the packet when the packet is ready for transmission. The semantics of the Hop_Lim field depend on the lower layer protocol that IOAM is encapsulated into, and therefore its specific semantics are outside the scope of this memo. The value of this field MUST be set to 0xff when the lower level does not have a TTL/Hop limit equivalent field.

node_id: 3-octet unsigned integer. Node identifier field to uniquely identify a node within the IOAM-Namespace and associated

IOAM-Domain. The procedure to allocate, manage and map the `node_ids` is beyond the scope of this document. See [I-D.ietf-ippm-ioam-deployment] for a discussion of deployment related aspects of the `node_id`.

5.4.2.2. `ingress_if_id` and `egress_if_id`

The "`ingress_if_id` and `egress_if_id`" field is a 4-octet field that is defined as follows:

```

 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
+-----+-----+-----+-----+-----+-----+-----+-----+
|           ingress_if_id           |           egress_if_id           |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

`ingress_if_id`: 2-octet unsigned integer. Interface identifier to record the ingress interface the packet was received on.

`egress_if_id`: 2-octet unsigned integer. Interface identifier to record the egress interface the packet is forwarded out of.

Note that due to the fact that IOAM uses its own IOAM-Namespaces for IOAM-Data-Fields, data fields like interface identifiers can be used in a flexible way to represent system resources that are associated with ingressing or egressing packets, i.e., `ingress_if_id` could represent a physical interface, a virtual or logical interface, or even a queue.

5.4.2.3. `timestamp seconds`

The "`timestamp seconds`" field is a 4-octet unsigned integer field. It contains the absolute timestamp in seconds that specifies the time at which the packet was received by the node. This field has three possible formats; based on either PTP (see e.g., [RFC8877]), NTP [RFC5905], or POSIX [POSIX]. The three timestamp formats are specified in Section 6. In all three cases, the Timestamp Seconds field contains the 32 most significant bits of the timestamp format that is specified in Section 6. If a node is not capable of populating this field, it assigns the value 0xFFFFFFFF. Note that this is a legitimate value that is valid for 1 second in approximately 136 years; the analyzer has to correlate several packets or compare the timestamp value to its own time-of-day in order to detect the error indication.

5.4.2.4. timestamp fraction

The "timestamp fraction" field is a 4-octet unsigned integer field. Fraction specifies the fractional portion of the number of seconds since the NTP epoch [RFC8877]. The field specifies the time at which the packet was received by the node. This field has three possible formats; based on either PTP (see e.g., [RFC8877]), NTP [RFC5905], or POSIX [POSIX]. The three timestamp formats are specified in Section 6. In all three cases, the Timestamp fraction field contains the 32 least significant bits of the timestamp format that is specified in Section 6. If a node is not capable of populating this field, it assigns the value 0xFFFFFFFF. Note that this is a legitimate value in the NTP format, valid for approximately 233 picoseconds in every second. If the NTP format is used the analyzer has to correlate several packets in order to detect the error indication.

5.4.2.5. transit delay

The "transit delay" field is a 4-octet unsigned integer in the range 0 to $2^{31}-1$. It is the time in nanoseconds the packet spent in the transit node. This can serve as an indication of the queuing delay at the node. If the transit delay exceeds $2^{31}-1$ nanoseconds then the top bit 'O' is set to indicate overflow and value set to 0x80000000. When this field is part of the data field but a node populating the field is not able to fill it, the field position in the field MUST be filled with value 0xFFFFFFFF to mean not populated.

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|O|                                     transit delay                      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

5.4.2.6. namespace specific data

The "namespace specific data" field is a 4-octet field which can be used by the node to add IOAM-Namespace specific data. This represents a "free-format" 4-octet bit field with its semantics defined in the context of a specific IOAM-Namespace.

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     namespace specific data                      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

5.4.2.7. queue depth

The "queue depth" field is a 4-octet unsigned integer field. This field indicates the current length of the egress interface queue of the interface from where the packet is forwarded out. The queue depth is expressed as the current amount of memory buffers used by the queue (a packet could consume one or more memory buffers, depending on its size).

```

 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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     queue depth                             |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

5.4.2.8. Checksum Complement

The "Checksum Complement" field is a 4-octet node data which contains a 4-octet Checksum Complement field. The Checksum Complement is useful when IOAM is transported over encapsulations that make use of a UDP transport, such as VXLAN-GPE or Geneve. Without the Checksum Complement, nodes adding IOAM node data update the UDP Checksum field following the recommendation of the encapsulation protocols. When the Checksum Complement is present, an IOAM encapsulating node or IOAM transit node adding node data MUST carry out one of the following two alternatives in order to maintain the correctness of the UDP Checksum value:

1. Recompute the UDP Checksum field.
2. Use the Checksum Complement to make a checksum-neutral update in the UDP payload; the Checksum Complement is assigned a value that complements the rest of the node data fields that were added by the current node, causing the existing UDP Checksum field to remain correct.

IOAM decapsulating nodes MUST recompute the UDP Checksum field, since they do not know whether previous hops modified the UDP Checksum field or the Checksum Complement field.

Checksum Complement fields are used in a similar manner in [RFC7820] and [RFC7821].

```

 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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Checksum Complement                             |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

5.4.2.9. Hop_Lim and node_id wide

The "Hop_Lim and node_id wide" field is an 8-octet field defined as follows:

```

  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
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |   Hop_Lim   |                               node_id           |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  ~                               node_id (contd)                   |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

```

Hop_Lim: 1-octet unsigned integer. See Section 5.4.2.1 for the definition of the field.

node_id: 7-octet unsigned integer. Node identifier field to uniquely identify a node within the IOAM-Namespace and associated IOAM-Domain. The procedure to allocate, manage and map the node_ids is beyond the scope of this document.

5.4.2.10. ingress_if_id and egress_if_id wide

The "ingress_if_id and egress_if_id wide" field is an 8-octet field which is defined as follows:

```

  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
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                               ingress_if_id                   |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                               egress_if_id                     |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

```

ingress_if_id: 4-octet unsigned integer. Interface identifier to record the ingress interface the packet was received on.

egress_if_id: 4-octet unsigned integer. Interface identifier to record the egress interface the packet is forwarded out of.

5.4.2.11. namespace specific data wide

The "namespace specific data wide" field is an 8-octet field which can be used by the node to add IOAM-Namespace specific data. This represents a "free-format" 8-octet bit field with its semantics defined in the context of a specific IOAM-Namespace.

```

 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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     namespace specific data                                     ~
+-----+-----+-----+-----+-----+-----+-----+-----+
~                                     namespace specific data (contd)                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

5.4.2.12. buffer occupancy

The "buffer occupancy" field is a 4-octet unsigned integer field. This field indicates the current status of the occupancy of the common buffer pool used by a set of queues. The units of this field are implementation specific. Hence, the units are interpreted within the context of an IOAM-Namespace and/or node-id if used. The authors acknowledge that in some operational cases there is a need for the units to be consistent across a packet path through the network, hence it is recommended for implementations to use standard units such as Bytes.

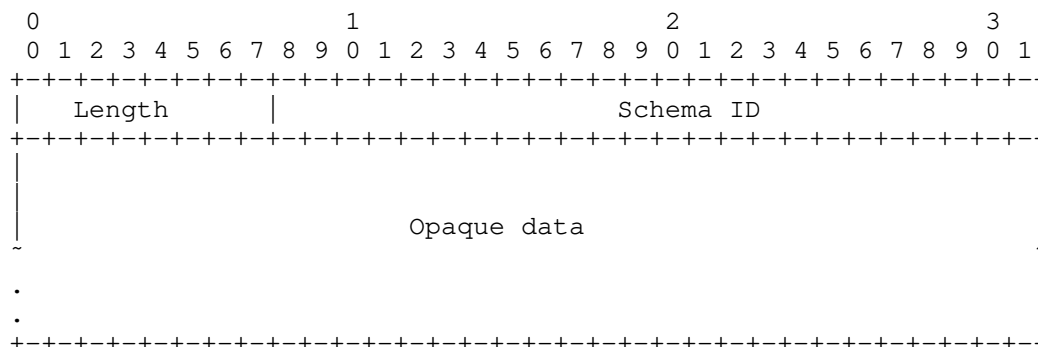
```

 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
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     buffer occupancy                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

5.4.2.13. Opaque State Snapshot

The "Opaque State Snapshot" is a variable length field and follows the fixed length IOAM-Data-Fields defined above. It allows the network element to store an arbitrary state in the node data field, without a pre-defined schema. The schema is to be defined within the context of an IOAM-Namespace. The schema needs to be made known to the analyzer by some out-of-band mechanism. The specification of this mechanism is beyond the scope of this document. A 24-bit "Schema Id" field, interpreted within the context of an IOAM-Namespace, indicates which particular schema is used, and has to be configured on the network element by the operator.



Length: 1-octet unsigned integer. It is the length in multiples of 4-octets of the Opaque data field that follows Schema Id.

Schema ID: 3-octet unsigned integer identifying the schema of Opaque data.

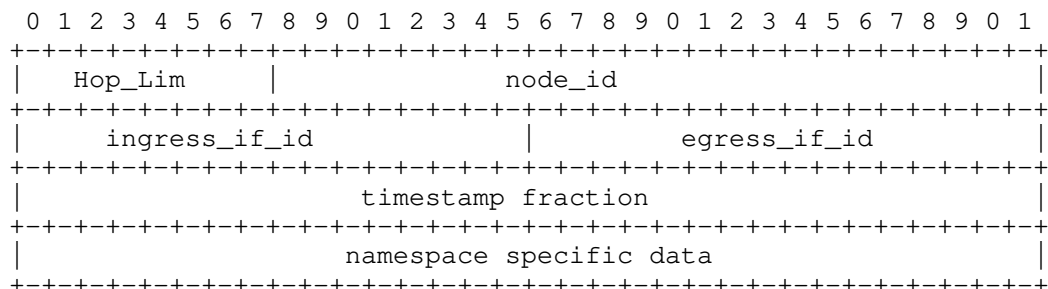
Opaque data: Variable length field. This field is interpreted as specified by the schema identified by the Schema ID.

When this field is part of the data field but a node populating the field has no opaque state data to report, the Length MUST be set to 0 and the Schema ID MUST be set to 0xFFFFFFFF to mean no schema.

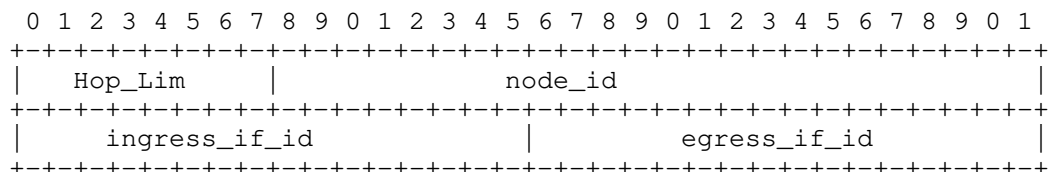
5.4.3. Examples of IOAM node data

The format used for the entries in a packet's "node data list" array can vary from packet to packet and deployment to deployment". Some deployments might only be interested in recording the node identifiers, whereas others might be interested in recording node identifiers and timestamps. This section provides example entries of the "node data list".

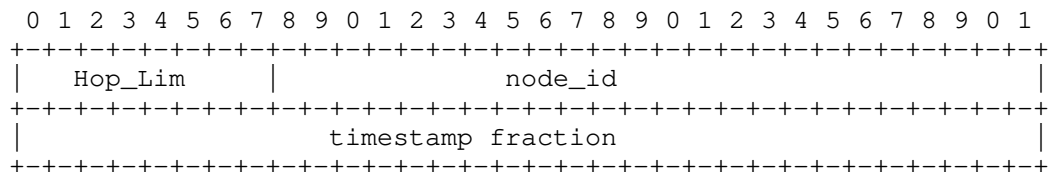
0xD40000: IOAM-Trace-Type is 0xD40000 (0b11010100000000000000000000000000) then the format of node data is:



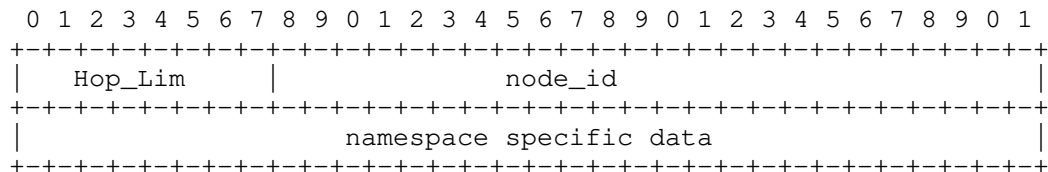
0xC00000: IOAM-Trace-Type is 0xC00000 (0b110000000000000000000000)
then the format is:



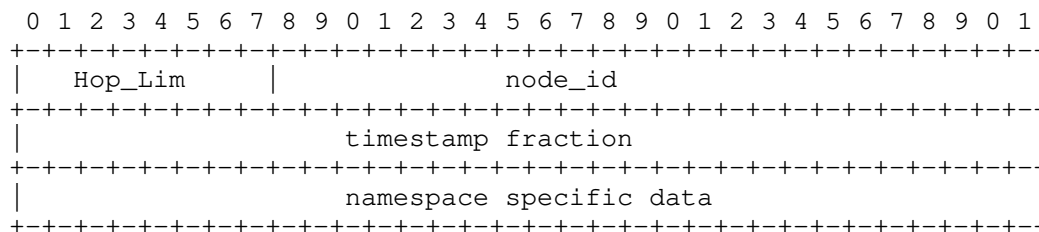
0x900000: IOAM-Trace-Type is 0x900000 (0b100100000000000000000000)
then the format is:



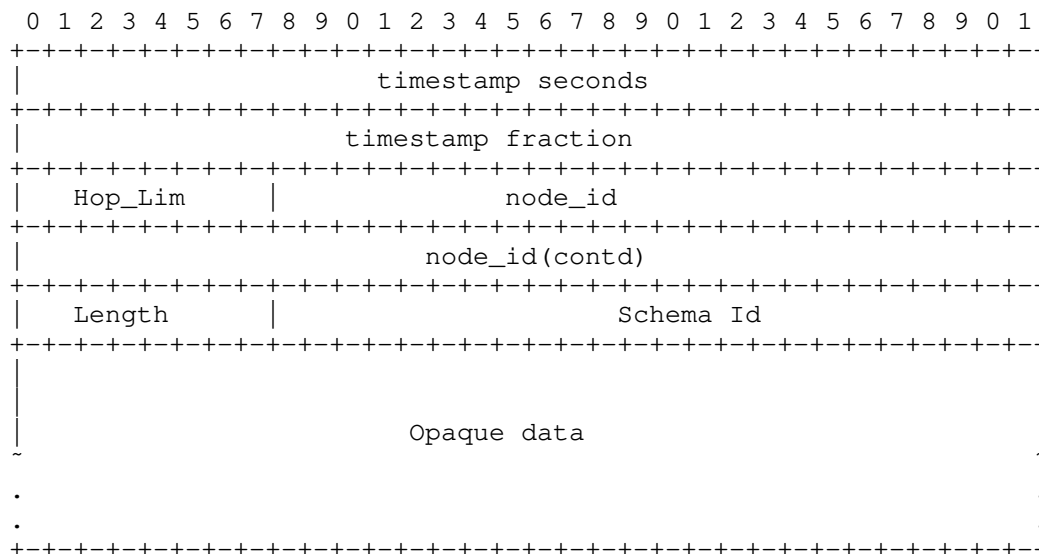
0x840000: IOAM-Trace-Type is 0x840000 (0b100001000000000000000000)
then the format is:



0x940000: IOAM-Trace-Type is 0x940000 (0b100101000000000000000000)
then the format is:



0x308002: IOAM-Trace-Type is 0x308002 (0b00110000010000000000000010)
then the format is:



5.5. IOAM Proof of Transit Option-Type

IOAM Proof of Transit Option-Type is used to support path or service function chain [RFC7665] verification use cases, i.e., prove that traffic transited a defined path. While details on how the IOAM data for the Proof-of-transit option is processed at IOAM encapsulating, decapsulating and transit nodes are outside the scope of the document, proof of transit approaches share the need to uniquely identify a packet as well as iteratively operate on a set of information that is handed from node to node. Correspondingly, two pieces of information are added as IOAM-Data-Fields to the packet:

- o PktID: Unique identifier for the packet.

- o Cumulative: Information which is handed from node to node and updated by every node according to a verification algorithm.

The IOAM Proof-of-Transit Option-Type consist of a fixed size "IOAM proof of transit option header" and "IOAM proof of transit option data fields":

IOAM proof of transit option header:

```

      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
+-----+-----+-----+-----+-----+-----+-----+-----+
|           Namespace-ID           | IOAM POT Type | IOAM POT flags |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

IOAM proof of transit Option-Type IOAM-Data-Fields MUST be 4-octet aligned:

```

      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
+-----+-----+-----+-----+-----+-----+-----+-----+
|           POT Option data field determined by IOAM-POT-Type           |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Namespace-ID: 16-bit identifier of an IOAM-Namespace. The Namespace-ID value of 0x0000 is defined as the "Default-Namespace-ID" (see Section 5.3) and MUST be known to all the nodes implementing IOAM. For any other Namespace-ID value that does not match any Namespace-ID the node is configured to operate on, the node MUST NOT change the contents of the IOAM-Data-Fields.

IOAM POT Type: 8-bit identifier of a particular POT variant that specifies the POT data that is included. This document defines POT Type 0:

0: POT data is a 16 Octet field to carry data associated to POT procedures.

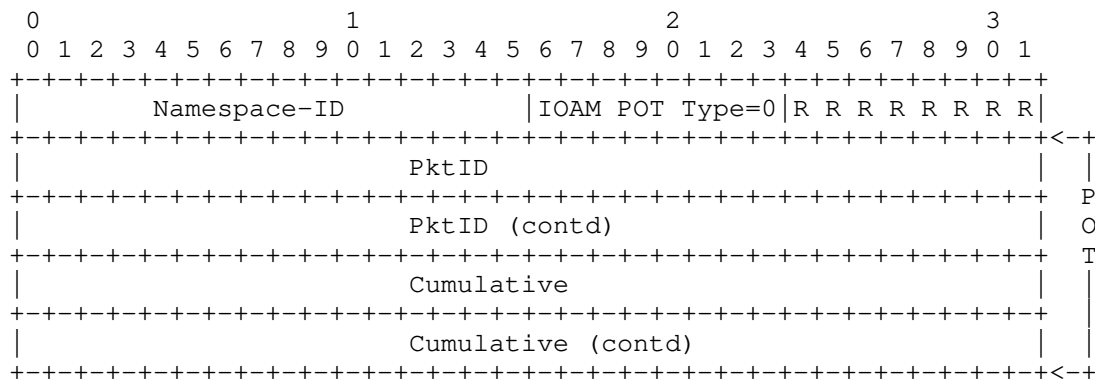
If a node receives an IOAM POT Type value that it does not understand, the node MUST NOT change, add to, or remove the contents of the OAM-Data-Fields.

IOAM POT flags: 8-bit. This document does not define any flags. Bits 0-7 These bits are available for assignment, see Section 8.5. Bits which have not been assigned MUST be set to zero upon transmission and ignored upon receipt.

POT Option data: Variable-length field. The type of which is determined by the IOAM-POT-Type.

5.5.1. IOAM Proof of Transit Type 0

IOAM proof of transit option of IOAM POT Type 0:



Namespace-ID: 16-bit identifier of an IOAM-Namespace (see Section 5.5 above).

IOAM POT Type: 8-bit identifier of a particular POT variant that specifies the POT data that is included (see Section 5.5 above). For this case here, IOAM POT Type is set to the value 0.

Bit 0-7: Undefined (see Section 5.5 above).

PktID: 64-bit packet identifier.

Cumulative: 64-bit Cumulative that is updated at specific nodes by processing per packet PktID field and configured parameters.

Note: Larger or smaller sizes of "PktID" and "Cumulative" data are feasible and could be required for certain deployments, e.g., in case of space constraints in the encapsulation protocols used. Future documents could introduce different sizes of data for "proof of transit".

5.6. IOAM Edge-to-Edge Option-Type

The IOAM Edge-to-Edge Option-Type is to carry data that is added by the IOAM encapsulating node and interpreted by IOAM decapsulating

node. The IOAM transit nodes MAY process the data but MUST NOT modify it.

The IOAM Edge-to-Edge Option-Type consist of a fixed size "IOAM Edge-to-Edge Option-Type header" and "IOAM Edge-to-Edge Option-Type data fields":

IOAM Edge-to-Edge Option-Type header:

```

      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
+-----+-----+-----+-----+-----+-----+-----+-----+
|           Namespace-ID           |           IOAM-E2E-Type           |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

IOAM Edge-to-Edge Option-Type IOAM-Data-Fields MUST be 4-octet aligned:

```

      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
+-----+-----+-----+-----+-----+-----+-----+-----+
|           E2E Option data field determined by IOAM-E2E-Type           |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Namespace-ID: 16-bit identifier of an IOAM-Namespace. The Namespace-ID value of 0x0000 is defined as the "Default-Namespace-ID" (see Section 5.3) and MUST be known to all the nodes implementing IOAM. For any other Namespace-ID value that does not match any Namespace-ID the node is configured to operate on, then the node MUST NOT change the contents of the IOAM-Data-Fields.

IOAM-E2E-Type: A 16-bit identifier which specifies which data types are used in the E2E option data. The IOAM-E2E-Type value is a bit field. The order of packing the E2E option data field elements follows the bit order of the IOAM-E2E-Type field, as follows:

- Bit 0 (Most significant bit) When set indicates presence of a 64-bit sequence number added to a specific "packet group" which is used to detect packet loss, packet reordering, or packet duplication within the group. The "packet group" is deployment dependent and defined at the IOAM encapsulating node, e.g., by n-tuple based classification of packets. When this bit is set, "Bit 1" (for 32-bit sequence number, see below) MUST be zero.
- Bit 1 When set indicates presence of a 32-bit sequence number added to a specific "packet group" which is used to

detect packet loss, packet reordering, or packet duplication within that group. The "packet group" is deployment dependent and defined at the IOAM encapsulating node, e.g., by n-tuple based classification of packets. When this bit is set, "Bit 0" (for 64-bit sequence number, see above) MUST be zero.

- Bit 2 When set indicates presence of timestamp seconds, representing the time at which the packet entered the IOAM-domain. Within the IOAM encapsulating node, the time that the timestamp is retrieved can depend on the implementation. Some possibilities are: 1) the time at which the packet was received by the node, 2) the time at which the packet was transmitted by the node, 3) when a tunnel encapsulation is used, the point at which the packet is encapsulated into the tunnel. Each implementation has to document when the E2E timestamp that is going to be put in the packet is retrieved. This 4-octet field has three possible formats; based on either PTP (see e.g., [RFC8877]), NTP [RFC5905], or POSIX [POSIX]. The three timestamp formats are specified in Section 6. In all three cases, the Timestamp Seconds field contains the 32 most significant bits of the timestamp format that is specified in Section 6. If a node is not capable of populating this field, it assigns the value 0xFFFFFFFF. Note that this is a legitimate value that is valid for 1 second in approximately 136 years; the analyzer has to correlate several packets or compare the timestamp value to its own time-of-day in order to detect the error indication.
- Bit 3 When set indicates presence of timestamp fraction, representing the time at which the packet entered the IOAM-domain. This 4-octet field has three possible formats; based on either PTP (see e.g., [RFC8877]), NTP [RFC5905], or POSIX [POSIX]. The three timestamp formats are specified in Section 6. In all three cases, the Timestamp fraction field contains the 32 least significant bits of the timestamp format that is specified in Section 6. If a node is not capable of populating this field, it assigns the value 0xFFFFFFFF. Note that this is a legitimate value in the NTP format, valid for approximately 233 picoseconds in every second. If the NTP format is used the analyzer has to correlate several packets in order to detect the error indication.

Bit 4-15 Undefined. An IOAM encapsulating node MUST set the value of these bits to zero upon transmission and ignore upon receipt.

E2E Option data: Variable-length field. The type of which is determined by the IOAM-E2E-Type.

6. Timestamp Formats

The IOAM-Data-Fields include a timestamp field which is represented in one of three possible timestamp formats. It is assumed that the management plane is responsible for determining which timestamp format is used.

6.1. PTP Truncated Timestamp Format

The Precision Time Protocol (PTP) uses an 80-bit timestamp format. The truncated timestamp format is a 64-bit field, which is the 64 least significant bits of the 80-bit PTP timestamp. The PTP truncated format is specified in Section 4.3 of [RFC8877], and the details are presented below for the sake of completeness.

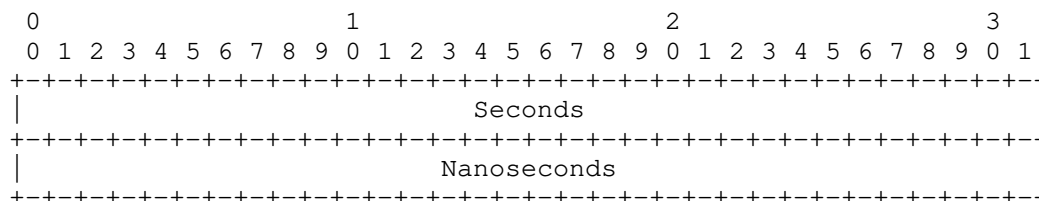


Figure 1: PTP Truncated Timestamp Format

Timestamp field format:

Seconds: specifies the integer portion of the number of seconds since the PTP epoch.

+ Size: 32 bits.

+ Units: seconds.

Nanoseconds: specifies the fractional portion of the number of seconds since the PTP epoch.

+ Size: 32 bits.

+ Units: nanoseconds. The value of this field is in the range 0 to $(10^9)-1$.

Epoch:

PTP epoch. For details see e.g., [RFC8877].

Resolution:

The resolution is 1 nanosecond.

Wraparound:

This time format wraps around every 2^{32} seconds, which is roughly 136 years. The next wraparound will occur in the year 2106.

Synchronization Aspects:

It is assumed that nodes that run this protocol are synchronized among themselves. Nodes MAY be synchronized to a global reference time. Note that if PTP is used for synchronization, the timestamp MAY be derived from the PTP-synchronized clock, allowing the timestamp to be measured with respect to the clock of an PTP Grandmaster clock.

6.2. NTP 64-bit Timestamp Format

The Network Time Protocol (NTP) [RFC5905] timestamp format is 64 bits long. This specification uses the NTP timestamp format that is specified in Section 4.2.1 of [RFC8877], and the details are presented below for the sake of completeness.

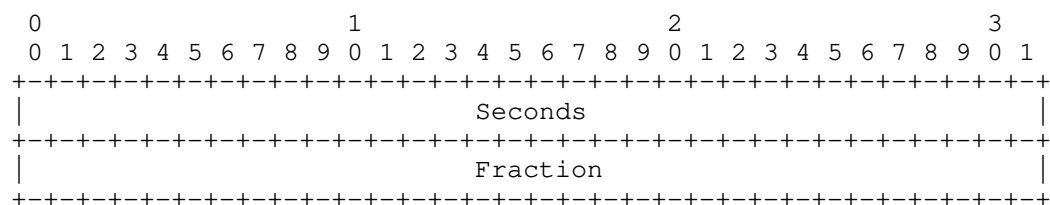


Figure 2: NTP [RFC5905] 64-bit Timestamp Format

Timestamp field format:

Seconds: specifies the integer portion of the number of seconds since the NTP epoch.

+ Size: 32 bits.

+ Units: seconds.

Fraction: specifies the fractional portion of the number of seconds since the NTP epoch.

+ Size: 32 bits.

+ Units: the unit is 2^{-32} seconds, which is roughly equal to 233 picoseconds.

Epoch:

NTP Epoch. For details see [RFC5905].

Resolution:

The resolution is 2^{-32} seconds.

Wraparound:

This time format wraps around every 2^{32} seconds, which is roughly 136 years. The next wraparound will occur in the year 2036.

Synchronization Aspects:

Nodes that use this timestamp format will typically be synchronized to UTC using NTP [RFC5905]. Thus, the timestamp MAY be derived from the NTP-synchronized clock, allowing the timestamp to be measured with respect to the clock of an NTP server.

6.3. POSIX-based Timestamp Format

This timestamp format is based on the POSIX time format [POSIX]. The detailed specification of the timestamp format used in this document is presented below.

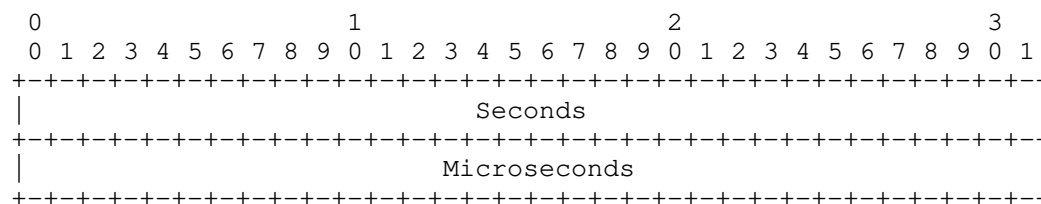


Figure 3: POSIX-based Timestamp Format

Timestamp field format:

Seconds: specifies the integer portion of the number of seconds since the POSIX epoch.

+ Size: 32 bits.

+ Units: seconds.

Microseconds: specifies the fractional portion of the number of seconds since the POSIX epoch.

+ Size: 32 bits.

+ Units: the unit is microseconds. The value of this field is in the range 0 to $(10^6)-1$.

Epoch:

POSIX epoch. For details, see [POSIX], appendix A.4.16.

Resolution:

The resolution is 1 microsecond.

Wraparound:

This time format wraps around every 2^{32} seconds, which is roughly 136 years. The next wraparound will occur in the year 2106.

Synchronization Aspects:

It is assumed that nodes that use this timestamp format run the Linux operating system, and hence use the POSIX time. In some cases nodes MAY be synchronized to UTC using a synchronization mechanism that is outside the scope of this document, such as NTP [RFC5905]. Thus, the timestamp MAY be derived from the NTP-synchronized clock, allowing the timestamp to be measured with respect to the clock of an NTP server.

7. IOAM Data Export

IOAM nodes collect information for packets traversing a domain that supports IOAM. IOAM decapsulating nodes as well as IOAM transit nodes can choose to retrieve IOAM information from the packet, process the information further and export the information using e.g., IPFIX. The mechanisms and associated data formats for exporting IOAM data is outside the scope of this document.

A way to perform raw data export of IOAM data using IPFIX is discussed in [I-D.spiegel-ippm-ioam-rawexport].

8. IANA Considerations

This document requests the following IANA Actions.

IANA is requested to define a registry group named "In-Situ OAM (IOAM) Protocol Parameters".

This group will include the following registries:

- IOAM Option-Type

- IOAM Trace-Type

- IOAM Trace-Flags

- IOAM POT-Type

- IOAM POT-Flags

- IOAM E2E-Type

- IOAM Namespace-ID

The subsequent sub-sections detail the registries herein contained.

8.1. IOAM Option-Type Registry

This registry defines 128 code points for the IOAM Option-Type field for identifying IOAM Option-Types as explained in Section 5. The following code points are defined in this draft:

- 0 IOAM Pre-allocated Trace Option-Type

- 1 IOAM Incremental Trace Option-Type

- 2 IOAM POT Option-Type

- 3 IOAM E2E Option-Type

4 - 127 are available for assignment via "IETF Review" process as per [RFC8126].

New registration requests MUST use the following template:

Name: Name of the newly registered Option-Type.

Code point: Desired value of the requested code point.

Description: Brief description of the newly registered Option-Type.

Reference: Reference to the document that defines the new Option-Type.

The evaluation of a new registration request MUST also include checking whether the new IOAM Option-Type includes an IOAM-Namespace field and that the IOAM-Namespace field is the first field in the newly defined header of the new Option-Type.

8.2. IOAM Trace-Type Registry

This registry defines code point for each bit in the 24-bit IOAM-Trace-Type field for Pre-allocated Trace-Option-Type and Incremental Trace-Option-Type defined in Section 5.4. The meaning of Bits 0 - 11 is defined in this document in Paragraph 5 of Section 5.4.1:

Bit 0 hop_Lim and node_id in short format

Bit 1 ingress_if_id and egress_if_id in short format

Bit 2 timestamp seconds

Bit 3 timestamp fraction

Bit 4 transit delay

Bit 5 namespace specific data in short format

Bit 6 queue depth

Bit 7 checksum complement

Bit 8 hop_Lim and node_id in wide format

Bit 9 ingress_if_id and egress_if_id in wide format

Bit 10 namespace specific data in wide format

Bit 11 buffer occupancy

Bit 22 variable length Opaque State Snapshot

Bit 23 reserved

The meaning for Bits 12 - 21 are available for assignment via "IETF Review" process as per [RFC8126].

New registration requests MUST use the following template:

Bit: Desired bit to be allocated in the 24-bit IOAM Trace-Option-Type field for Pre-allocated Trace-Option-Type and Incremental Trace-Option-Type.

Description: Brief description of the newly registered bit.

Reference: Reference to the document that defines the new bit.

8.3. IOAM Trace-Flags Registry

This registry defines code points for each bit in the 4 bit flags for the Pre-allocated trace option and for the Incremental trace option defined in Section 5.4. The meaning of Bit 0 (the most significant bit) for trace flags is defined in this document in Paragraph 3 of Section 5.4.1:

Bit 0 "Overflow" (O-bit)

Bit 1 - 3 are available for assignment via "IETF Review" process as per [RFC8126].

New registration requests MUST use the following template:

Bit: Desired bit to be allocated in the 8 bit flags field of the Pre-allocated Trace-Option-Type and for the Incremental Trace-Option-Type.

Description: Brief description of the newly registered bit.

Reference: Reference to the document that defines the new bit.

8.4. IOAM POT-Type Registry

This registry defines 256 code points to define IOAM POT Type for IOAM proof of transit option Section 5.5. The code point value 0 is defined in this document:

0: 16 Octet POT data

1 - 255 are available for assignment via "IETF Review" process as per [RFC8126].

New registration requests MUST use the following template:

Name: Name of the newly registered POT-Type.

Code point: Desired value of the requested code point.

Description: Brief description of the newly registered POT-Type.

Reference: Reference to the document that defines the new POT-Type.

8.5. IOAM POT-Flags Registry

This registry defines code points for each bit in the 8 bit flags for IOAM POT Option-Type defined in Section 5.5.

The meaning for Bits 0 - 7 are available for assignment via "IETF Review" process as per [RFC8126].

New registration requests MUST use the following template:

Bit: Desired bit to be allocated in the 8 bit flags field of the IOAM POT Option-Type.

Description: Brief description of the newly registered bit.

Reference: Reference to the document that defines the new bit.

8.6. IOAM E2E-Type Registry

This registry defines code points for each bit in the 16 bit IOAM-E2E-Type field for IOAM E2E option Section 5.6. The meaning of Bit 0 - 3 are defined in this document:

Bit 0 64-bit sequence number

Bit 1 32-bit sequence number

Bit 2 timestamp seconds

Bit 3 timestamp fraction

The meaning of Bits 4 - 15 are available for assignment via "IETF Review" process as per [RFC8126].

New registration requests MUST use the following template:

Bit: Desired bit to be allocated in the 16 bit IOAM-E2E-Type field.

Description: Brief description of the newly registered bit.

Reference: Reference to the document that defines the new bit.

8.7. IOAM Namespace-ID Registry

IANA is requested to set up an "IOAM Namespace-ID Registry", containing 16-bit values and following the template for requests shown below. The meaning of 0x0000 is defined in this document. IANA is requested to reserve the values 0x0001 to 0x7FFF for private use (managed by operators), as specified in Section 5.3 of the current document. Registry entries for the values 0x8000 to 0xFFFF are to be assigned via the "Expert Review" policy defined in [RFC8126].

Upon receiving a new allocation request, a designated expert will perform the following:

- o Review whether the request is complete, i.e., the registration template has been filled in. The expert will send incomplete requests back to the requestor.
- o Check whether the request is neither a duplicate of nor conflicting with either an already existing allocation or a pending allocation. In case of duplicates or conflicts, the expert will ask the requestor to update the allocation request accordingly.
- o Solicit feedback from relevant working groups and communities to ensure that the new allocation request has been properly reviewed and that rough consensus on the request exists. At a minimum, the expert will solicit feedback from the IPPM working group in the IETF by posting the request to the `ippm@ietf.org` mailing list. The expert will allow for a 3-week review period on the mailing lists. If the feedback received from the relevant working groups and communities within the review period indicates rough consensus on the request, the expert will approve the request and ask IANA for allocating the new Namespace-ID. In case the expert senses a lack of consensus from the feedback received, the expert will ask the requestor to engage with the corresponding working groups and communities to further review and refine the request.

It is intended that any allocation will be accompanied by a published RFC. In order to allow for the allocation of code points prior to the RFC being approved for publication, the designated expert can approve allocations once it seems clear that an RFC will be published.

0x0000: default namespace (known to all IOAM nodes)

0x0001 - 0x7FFF: reserved for private use

0x8000 - 0xFFFF: unassigned

New registration requests MUST use the following template:

Name: Name of the newly registered Namespace-ID.

Code point: Desired value of the requested Namespace-ID.

Description: Brief description of the newly registered Namespace-ID.

Reference: Reference to the document that defines the new Namespace-ID.

Status of the registration: Status can be either "permanent" or "provisional". Namespace-ID registrations following a successful expert review will have the status "provisional". Once the RFC, which defines the new Namespace-ID is published, the status is changed to "permanent".

9. Management and Deployment Considerations

This document defines the structure and use of IOAM data fields. This document does not define the encapsulation of IOAM data fields into different protocols. Management and deployment aspects for IOAM have to be considered within the context of the protocol IOAM data fields are encapsulated into and as such, are out of scope for this document. For a discussion of IOAM deployment, please also refer to [I-D.ietf-ippm-ioam-deployment], which outlines a framework for IOAM deployment and provides best current practices.

10. Security Considerations

As discussed in [RFC7276], a successful attack on an OAM protocol in general, and specifically on IOAM, can prevent the detection of failures or anomalies, or create a false illusion of nonexistent ones. In particular, these threats are applicable by compromising the integrity of IOAM data, either by maliciously modifying IOAM options in transit, or by injecting packets with maliciously generated IOAM options. All nodes in the path of a IOAM carrying packet can perform such an attack.

The Proof of Transit Option-Type (see Section 5.5) is used for verifying the path of data packets, i.e., proving that packets transited through a defined set of nodes.

In case an attacker gains access to several nodes in a network and would be able to change the system software of these nodes, IOAM data fields could be misused and repurposed for a use different from what is specified in this document. One type of misuse is the implementation of a covert channel between network nodes.

From a confidentiality perspective, although IOAM options are not expected to contain user data, they can be used for network reconnaissance, allowing attackers to collect information about network paths, performance, queue states, buffer occupancy and other information. Moreover, if IOAM data leaks from the IOAM-domain it could enable reconnaissance beyond the scope of the IOAM-domain. One possible application of such reconnaissance is to gauge the effectiveness of an ongoing attack, e.g., if buffers and queues are overflowing.

IOAM can be used as a means for implementing Denial of Service (DoS) attacks, or for amplifying them. For example, a malicious attacker can add an IOAM header to packets in order to consume the resources of network devices that take part in IOAM or entities that receive, collect or analyze the IOAM data. Another example is a packet length attack, in which an attacker pushes headers associated with IOAM Option-Types into data packets, causing these packets to be increased beyond the MTU size, resulting in fragmentation or in packet drops. In case POT is used, an attacker could corrupt the POT data fields in the packet, resulting in a verification failure of the POT data, even if the packet followed the correct path.

Since IOAM options can include timestamps, if network devices use synchronization protocols then any attack on the time protocol [RFC7384] can compromise the integrity of the timestamp-related data fields.

At the management plane, attacks can be set up by misconfiguring or by maliciously configuring IOAM-enabled nodes in a way that enables other attacks. IOAM configuration should only be managed by authorized processes or users.

IETF protocols require features to ensure their security. While IOAM data fields don't represent a protocol by themselves, the IOAM data fields add to the protocol that the IOAM data fields are encapsulated into. Any specification that defines how IOAM data fields are carried in an encapsulating protocol MUST provide for a mechanism for cryptographic integrity protection of the IOAM data fields. Cryptographic integrity protection could be either achieved through a mechanism of the encapsulating protocol or it could incorporate the mechanisms specified in [I-D.ietf-ippm-ioam-data-integrity].

The current document does not define a specific IOAM encapsulation. It has to be noted that some IOAM encapsulation types can introduce specific security considerations. A specification that defines an IOAM encapsulation is expected to address the respective encapsulation-specific security considerations.

Notably, IOAM is expected to be deployed in limited domains, thus confining the potential attack vectors to within the limited domain. A limited administrative domain provides the operator with the means to select, monitor, and control the access of all the network devices, making these devices trusted by the operator. Indeed, in order to limit the scope of threats mentioned above to within the current limited domain the network operator is expected to enforce policies that prevent IOAM traffic from leaking outside of the IOAM domain, and prevent IOAM data from outside the domain to be processed and used within the domain.

This document does not define the data contents of custom fields like "Opaque State Snapshot" and "namespace specific data" IOAM data fields. These custom data fields will have security considerations corresponding to their defined data contents that need to be described where those formats are defined.

IOAM deployments which leverage both IOAM Trace Option-Types, i.e., the Pre-allocated Trace Option-Type and Incremental Trace Option-Type can suffer from incomplete visibility if the information gathered via the two Trace Option-Types is not correlated and aggregated appropriately. If IOAM transit nodes leverage the IOAM data fields in the packet for further actions or insights, then IOAM transit nodes which only support one IOAM Trace Option-Type in an IOAM deployment which leverages both Trace Option-Types, have limited visibility and thus can draw inappropriate conclusions or take wrong actions.

The security considerations of a system that deploys IOAM, much like any system, has to be reviewed on a per-deployment-scenario basis, based on a systems-specific threat analysis, which can lead to specific security solutions that are beyond the scope of the current document. Specifically, in an IOAM deployment that is not confined to a single LAN, but spans multiple inter-connected sites (for example, using an overlay network), the inter-site links can be secured (e.g., by IPsec) in order to avoid external threats.

IOAM deployment considerations, including approaches to mitigate the above discussed threads and potential attacks are outside the scope of this document. IOAM deployment considerations are discussed in [I-D.ietf-ippm-ioam-deployment].

11. Acknowledgements

The authors would like to thank Eric Vyncke, Nalini Elkins, Srihari Raghavan, Ranganathan T S, Karthik Babu Harichandra Babu, Akshaya Nadahalli, LJ Wobker, Erik Nordmark, Vengada Prasad Govindan, Andrew Yourtchenko, Aviv Kfir, Tianran Zhou, Zhenbin (Robin) and Greg Mirsky for the comments and advice.

This document leverages and builds on top of several concepts described in [I-D.kitamura-ipv6-record-route]. The authors would like to acknowledge the work done by the author Hiroshi Kitamura and people involved in writing it.

The authors would like to gracefully acknowledge useful review and insightful comments received from Joe Clarke, Al Morton, Tom Herbert, Carlos Bernardos, Haoyu Song, Mickey Spiegel, Roman Danyliw, Benjamin Kaduk, Murray S. Kucherawy, Ian Swett, Martin Duke, Francesca Palombini, Lars Eggert, Alvaro Retana, Erik Kline, Robert Wilton, Zaheduzzaman Sarker, Dan Romascanu and Barak Gafni.

12. References

12.1. Normative References

- [POSIX] Institute of Electrical and Electronics Engineers, "IEEE Std 1003.1-2017 (Revision of IEEE Std 1003.1-2017) - IEEE Standard for Information Technology - Portable Operating System Interface (POSIX(TM) Base Specifications, Issue 7)", IEEE Std 1003.1-2017, 2017, <<https://standards.ieee.org/findstds/standard/1003.1-2017.html>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <<https://www.rfc-editor.org/info/rfc5905>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.

- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

12.2. Informative References

- [I-D.ietf-ippm-ioam-data-integrity]
Brockners, F., Bhandari, S., and T. Mizrahi, "Integrity of In-situ OAM Data Fields", draft-ietf-ippm-ioam-data-integrity-00 (work in progress), October 2021.
- [I-D.ietf-ippm-ioam-deployment]
Brockners, F., Bhandari, S., Bernier, D., and T. Mizrahi, "In-situ OAM Deployment", draft-ietf-ippm-ioam-deployment-00 (work in progress), October 2021.
- [I-D.ietf-nvo3-vxlan-gpe]
(Editor), F. M., (editor), L. K., and U. E. (editor), "Generic Protocol Extension for VXLAN (VXLAN-GPE)", draft-ietf-nvo3-vxlan-gpe-12 (work in progress), September 2021.
- [I-D.kitamura-ipv6-record-route]
Kitamura, H., "Record Route for IPv6 (PR6) Hop-by-Hop Option Extension", draft-kitamura-ipv6-record-route-00 (work in progress), November 2000.
- [I-D.spiegel-ippm-ioam-rawexport]
Spiegel, M., Brockners, F., Bhandari, S., and R. Sivakolundu, "In-situ OAM raw data export with IPFIX", draft-spiegel-ippm-ioam-rawexport-05 (work in progress), July 2021.
- [RFC7276] Mizrahi, T., Sprecher, N., Bellagamba, E., and Y. Weingarten, "An Overview of Operations, Administration, and Maintenance (OAM) Tools", RFC 7276, DOI 10.17487/RFC7276, June 2014, <<https://www.rfc-editor.org/info/rfc7276>>.
- [RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", RFC 7384, DOI 10.17487/RFC7384, October 2014, <<https://www.rfc-editor.org/info/rfc7384>>.
- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<https://www.rfc-editor.org/info/rfc7665>>.

- [RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <<https://www.rfc-editor.org/info/rfc7799>>.
- [RFC7820] Mizrahi, T., "UDP Checksum Complement in the One-Way Active Measurement Protocol (OWAMP) and Two-Way Active Measurement Protocol (TWAMP)", RFC 7820, DOI 10.17487/RFC7820, March 2016, <<https://www.rfc-editor.org/info/rfc7820>>.
- [RFC7821] Mizrahi, T., "UDP Checksum Complement in the Network Time Protocol (NTP)", RFC 7821, DOI 10.17487/RFC7821, March 2016, <<https://www.rfc-editor.org/info/rfc7821>>.
- [RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <<https://www.rfc-editor.org/info/rfc8300>>.
- [RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <<https://www.rfc-editor.org/info/rfc8799>>.
- [RFC8877] Mizrahi, T., Fabini, J., and A. Morton, "Guidelines for Defining Packet Timestamps", RFC 8877, DOI 10.17487/RFC8877, September 2020, <<https://www.rfc-editor.org/info/rfc8877>>.
- [RFC8926] Gross, J., Ed., Ganga, I., Ed., and T. Sridhar, Ed., "Geneve: Generic Network Virtualization Encapsulation", RFC 8926, DOI 10.17487/RFC8926, November 2020, <<https://www.rfc-editor.org/info/rfc8926>>.

Contributors' Addresses

Carlos Pignataro
Cisco Systems, Inc.
7200-11 Kit Creek Road
Research Triangle Park, NC 27709
United States

Email: cpignata@cisco.com

Mickey Spiegel
Barefoot Networks, an Intel company
4750 Patrick Henry Drive
Santa Clara, CA 95054

US

Email: mickey.spiegel@intel.com

Barak Gafni
Nvidia
350 Oakmead Parkway, Suite 100
Sunnyvale, CA 94085
U.S.A.

Email: gbarak@nvidia.com

Jennifer Lemon
Broadcom
270 Innovation Drive
San Jose, CA 95134
US

Email: jennifer.lemon@broadcom.com

Hannes Gredler
RtBrick Inc.

Email: hannes@rtbrick.com

John Leddy
United States

Email: john@leddy.net

Stephen Youell
JP Morgan Chase
25 Bank Street
London E14 5JP
United Kingdom

Email: stephen.youell@jpmorgan.com

David Mozes

Email: mosesster@gmail.com

Petr Lapukhov
Facebook
1 Hacker Way
Menlo Park, CA 94025
US

Email: petr@fb.com

Remy Chang
Barefoot Networks
4750 Patrick Henry Drive
Santa Clara, CA 95054
US

Email: remy@barefootnetworks.com

Daniel Bernier
Bell Canada
Canada

Email: daniel.bernier@bell.ca

Authors' Addresses

Frank Brockners (editor)
Cisco Systems, Inc.
Hansaallee 249, 3rd Floor
DUESSELDORF, NORDRHEIN-WESTFALEN 40549
Germany

Email: fbrockne@cisco.com

Shwetha Bhandari (editor)
Thoughtspot
3rd Floor, Indiqube Orion, 24th Main Rd, Garden Layout, HSR Layout
Bangalore, KARNATAKA 560 102
India

Email: shwetha.bhandari@thoughtspot.com

Tal Mizrahi (editor)
Huawei
8-2 Matam
Haifa 3190501
Israel

Email: tal.mizrahi.phd@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2020

M. Bagnulo
UC3M
B. Claise
Cisco Systems, Inc.
P. Eardley
BT
A. Morton
AT&T Labs
A. Akhter
Consultant
March 9, 2020

Registry for Performance Metrics
draft-ietf-ippm-metric-registry-24

Abstract

This document defines the format for the IANA Performance Metrics Registry. This document also gives a set of guidelines for Registered Performance Metric requesters and reviewers.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 10, 2020.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	4
2. Terminology	5
3. Scope	7
4. Motivation for a Performance Metrics Registry	8
4.1. Interoperability	8
4.2. Single point of reference for Performance Metrics	9
4.3. Side benefits	9
5. Criteria for Performance Metrics Registration	9
6. Performance Metric Registry: Prior attempt	10
6.1. Why this Attempt Should Succeed	11
7. Definition of the Performance Metric Registry	11
7.1. Summary Category	13
7.1.1. Identifier	13
7.1.2. Name	13
7.1.3. URI	17
7.1.4. Description	17
7.1.5. Reference	17
7.1.6. Change Controller	17
7.1.7. Version (of Registry Format)	18
7.2. Metric Definition Category	18
7.2.1. Reference Definition	18
7.2.2. Fixed Parameters	18
7.3. Method of Measurement Category	19
7.3.1. Reference Method	19
7.3.2. Packet Stream Generation	19
7.3.3. Traffic Filter	20
7.3.4. Sampling Distribution	20
7.3.5. Run-time Parameters	21
7.3.6. Role	22
7.4. Output Category	22
7.4.1. Type	22
7.4.2. Reference Definition	23
7.4.3. Metric Units	23
7.4.4. Calibration	23
7.5. Administrative information	24
7.5.1. Status	24
7.5.2. Requester	24
7.5.3. Revision	24
7.5.4. Revision Date	24
7.6. Comments and Remarks	24

8.	Processes for Managing the Performance Metric Registry Group	24
8.1.	Adding new Performance Metrics to the Performance Metrics Registry	25
8.2.	Revising Registered Performance Metrics	26
8.3.	Deprecating Registered Performance Metrics	28
9.	Security considerations	28
10.	IANA Considerations	29
10.1.	Registry Group	29
10.2.	Performance Metric Name Elements	29
10.3.	New Performance Metrics Registry	30
11.	Blank Registry Template	32
11.1.	Summary	32
11.1.1.	ID (Identifier)	32
11.1.2.	Name	32
11.1.3.	URI	32
11.1.4.	Description	32
11.1.5.	Change Controller	32
11.1.6.	Version (of Registry Format)	32
11.2.	Metric Definition	32
11.2.1.	Reference Definition	32
11.2.2.	Fixed Parameters	32
11.3.	Method of Measurement	33
11.3.1.	Reference Method	33
11.3.2.	Packet Stream Generation	33
11.3.3.	Traffic Filtering (observation) Details	33
11.3.4.	Sampling Distribution	33
11.3.5.	Run-time Parameters and Data Format	33
11.3.6.	Roles	33
11.4.	Output	33
11.4.1.	Type	34
11.4.2.	Reference Definition	34
11.4.3.	Metric Units	34
11.4.4.	Calibration	34
11.5.	Administrative items	34
11.5.1.	Status	34
11.5.2.	Requester	34
11.5.3.	Revision	34
11.5.4.	Revision Date	34
11.6.	Comments and Remarks	34
12.	Acknowledgments	34
13.	References	35
13.1.	Normative References	35
13.2.	Informative References	36
	Authors' Addresses	37

1. Introduction

The IETF specifies and uses Performance Metrics of protocols and applications transported over its protocols. Performance metrics are important part of network operations using IETF protocols, and [RFC6390] specifies guidelines for their development.

The definition and use of Performance Metrics in the IETF has been fostered in various working groups (WG), most notably:

The "IP Performance Metrics" (IPPM) WG is the WG primarily focusing on Performance Metrics definition at the IETF.

The "Benchmarking Methodology" WG (BMWG) defines many Performance Metrics for use in laboratory benchmarking of inter-networking technologies.

The "Metric Blocks for use with RTCP's Extended Report Framework" (XRBLOCK) WG (concluded) specified many Performance Metrics related to "RTP Control Protocol Extended Reports (RTCP XR)" [RFC3611], which establishes a framework to allow new information to be conveyed in RTCP, supplementing the original report blocks defined in "RTP: A Transport Protocol for Real-Time Applications", [RFC3550].

The "IP Flow Information eXport" (IPFIX) concluded WG specified an IANA process for new Information Elements. Some Performance Metrics related Information Elements are proposed on regular basis.

The "Performance Metrics for Other Layers" (PMOL) a concluded WG defined some Performance Metrics related to Session Initiation Protocol (SIP) voice quality [RFC6035].

It is expected that more Performance Metrics will be defined in the future, not only IP-based metrics, but also metrics which are protocol-specific and application-specific.

Despite the importance of Performance Metrics, there are two related problems for the industry. First, ensuring that when one party requests another party to measure (or report or in some way act on) a particular Performance Metric, then both parties have exactly the same understanding of what Performance Metric is being referred to. Second, discovering which Performance Metrics have been specified, to avoid developing a new Performance Metric that is very similar, but not quite inter-operable. These problems can be addressed by creating a registry of performance metrics. The usual way in which the IETF organizes registries is with Internet Assigned Numbers

Authority (IANA), and there is currently no Performance Metrics Registry maintained by the IANA.

This document requests that IANA create and maintain a Performance Metrics Registry, according to the maintenance procedures and the Performance Metrics Registry format defined in this memo. The resulting Performance Metrics Registry is for use by the IETF and others. Although the Registry formatting specifications herein are primarily for registry creation by IANA, any other organization that wishes to create a performance metrics registry may use the same formatting specifications for their purposes. The authors make no guarantee of the registry format's applicability to any possible set of Performance Metrics envisaged by other organizations, but encourage others to apply it. In the remainder of this document, unless we explicitly say otherwise, we will refer to the IANA-maintained Performance Metrics Registry as simply the Performance Metrics Registry.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Performance Metric: A Performance Metric is a quantitative measure of performance, targeted to an IETF-specified protocol or targeted to an application transported over an IETF-specified protocol. Examples of Performance Metrics are the FTP response time for a complete file download, the DNS response time to resolve the IP address(es), a database logging time, etc. This definition is consistent with the definition of metric in [RFC2330] and broader than the definition of performance metric in [RFC6390].

Registered Performance Metric: A Registered Performance Metric is a Performance Metric expressed as an entry in the Performance Metrics Registry, administered by IANA. Such a performance metric has met all the registry review criteria defined in this document in order to be included in the registry.

Performance Metrics Registry: The IANA registry containing Registered Performance Metrics.

Proprietary Registry: A set of metrics that are registered in a proprietary registry, as opposed to Performance Metrics Registry.

Performance Metrics Experts: The Performance Metrics Experts is a group of designated experts [RFC8126] selected by the IESG to validate the Performance Metrics before updating the Performance Metrics Registry. The Performance Metrics Experts work closely with IANA.

Parameter: A Parameter is an input factor defined as a variable in the definition of a Performance Metric. A Parameter is a numerical or other specified factor forming one of a set that defines a metric or sets the conditions of its operation. All Parameters must be known in order to make a measurement using a metric and interpret the results. There are two types of Parameters: Fixed and Run-time parameters. For the Fixed Parameters, the value of the variable is specified in the Performance Metrics Registry entry and different Fixed Parameter values results in different Registered Performance Metrics. For the Run-time Parameters, the value of the variable is defined when the metric measurement method is executed and a given Registered Performance Metric supports multiple values for the parameter. Although Run-time Parameters do not change the fundamental nature of the Performance Metric's definition, some have substantial influence on the network property being assessed and interpretation of the results.

Note: Consider the case of packet loss in the following two Active Measurement Method cases. The first case is packet loss as background loss where the Run-time Parameter set includes a very sparse Poisson stream, and only characterizes the times when packets were lost. Actual user streams likely see much higher loss at these times, due to tail drop or radio errors. The second case is packet loss as inverse of throughput where the Run-time Parameter set includes a very dense, bursty stream, and characterizes the loss experienced by a stream that approximates a user stream. These are both "loss metrics", but the difference in interpretation of the results is highly dependent on the Run-time Parameters (at least), to the extreme where we are actually using loss to infer its compliment: delivered throughput.

Active Measurement Method: Methods of Measurement conducted on traffic which serves only the purpose of measurement and is generated for that reason alone, and whose traffic characteristics are known a priori. The complete definition of Active Methods is specified in section 3.4 of [RFC7799]. Examples of Active Measurement Methods are the measurement methods for the One way delay metric defined in [RFC7679] and the one for round trip delay defined in [RFC2681].

Passive Measurement Method: Methods of Measurement conducted on network traffic, generated either from the end users or from network elements that would exist regardless whether the measurement was being conducted or not. The complete definition of Passive Methods is specified in section 3.6 of [RFC7799]. One characteristic of Passive Measurement Methods is that sensitive information may be observed, and as a consequence, stored in the measurement system.

Hybrid Measurement Method: Hybrid Methods are Methods of Measurement that use a combination of Active Methods and Passive Methods, to assess Active Metrics, Passive Metrics, or new metrics derived from the a priori knowledge and observations of the stream of interest. The complete definition of Hybrid Methods is specified in section 3.8 of [RFC7799].

3. Scope

This document is intended for two different audiences:

1. For those defining new Registered Performance Metrics, it provides specifications and best practices to be used in deciding which Registered Performance Metrics are useful for a measurement study, instructions for writing the text for each column of the Registered Performance Metrics, and information on the supporting documentation required for the new Performance Metrics Registry entry (up to and including the publication of one or more immutable documents such as an RFC).
2. For the appointed Performance Metrics Experts and for IANA personnel administering the new IANA Performance Metrics Registry, it defines a set of acceptance criteria against which these proposed Registered Performance Metrics should be evaluated.

In addition, this document may be useful for other organizations who are defining a Performance Metric registry of their own, and may re-use the features of the Performance Metrics Registry defined in this document.

This Performance Metrics Registry is applicable to Performance Metrics issued from Active Measurement, Passive Measurement, and any other form of Performance Metric. This registry is designed to encompass Performance Metrics developed throughout the IETF and especially for the technologies specified in the following working groups: IPPM, XRBLOCK, IPFIX, and BMWG. This document analyzes a prior attempt to set up a Performance Metrics Registry, and the reasons why this design was inadequate [RFC6248]. Finally, this

document gives a set of guidelines for requesters and expert reviewers of candidate Registered Performance Metrics.

This document makes no attempt to populate the Performance Metrics Registry with initial entries; the related memo [I-D.ietf-ippm-initial-registry] proposes the initial set of registry entries.

4. Motivation for a Performance Metrics Registry

In this section, we detail several motivations for the Performance Metrics Registry.

4.1. Interoperability

As with any IETF registry, the primary intention is to manage registration of identifiers for use within one or more protocols. In the particular case of the Performance Metrics Registry, there are two types of protocols that will use the Performance Metrics in the Performance Metrics Registry during their operation (by referring to the Index values):

- o Control protocol: This type of protocol used to allow one entity to request another entity to perform a measurement using a specific metric defined by the Performance Metrics Registry. One particular example is the LMAP framework [RFC7594]. Using the LMAP terminology, the Performance Metrics Registry is used in the LMAP Control protocol to allow a Controller to schedule a measurement task for one or more Measurement Agents. In order to enable this use case, the entries of the Performance Metrics Registry must be sufficiently defined to allow a Measurement Agent implementation to trigger a specific measurement task upon the reception of a control protocol message. This requirement heavily constrains the type of entries that are acceptable for the Performance Metrics Registry.
- o Report protocol: This type of protocol is used to allow an entity to report measurement results to another entity. By referencing to a specific Performance Metrics Registry, it is possible to properly characterize the measurement result data being reported. Using the LMAP terminology, the Performance Metrics Registry is used in the Report protocol to allow a Measurement Agent to report measurement results to a Collector.

It should be noted that the LMAP framework explicitly allows for using not only the IANA-maintained Performance Metrics Registry but also other registries containing Performance Metrics, either defined by other organizations or private ones. However, others who are

creating Registries to be used in the context of an LMAP framework are encouraged to use the Registry format defined in this document, because this makes it easier for developers of LMAP Measurement Agents (MAs) to programmatically use information found in those other Registries' entries.

4.2. Single point of reference for Performance Metrics

A Performance Metrics Registry serves as a single point of reference for Performance Metrics defined in different working groups in the IETF. As we mentioned earlier, there are several WGs that define Performance Metrics in the IETF and it is hard to keep track of all them. This results in multiple definitions of similar Performance Metrics that attempt to measure the same phenomena but in slightly different (and incompatible) ways. Having a registry would allow the IETF community and others to have a single list of relevant Performance Metrics defined by the IETF (and others, where appropriate). The single list is also an essential aspect of communication about Performance Metrics, where different entities that request measurements, execute measurements, and report the results can benefit from a common understanding of the referenced Performance Metric.

4.3. Side benefits

There are a couple of side benefits of having such a registry. First, the Performance Metrics Registry could serve as an inventory of useful and used Performance Metrics, that are normally supported by different implementations of measurement agents. Second, the results of measurements using the Performance Metrics should be comparable even if they are performed by different implementations and in different networks, as the Performance Metric is properly defined. BCP 176 [RFC6576] examines whether the results produced by independent implementations are equivalent in the context of evaluating the completeness and clarity of metric specifications. This BCP defines the standards track advancement testing for (active) IPPM metrics, and the same process will likely suffice to determine whether Registered Performance Metrics are sufficiently well specified to result in comparable (or equivalent) results. Registered Performance Metrics which have undergone such testing SHOULD be noted, with a reference to the test results.

5. Criteria for Performance Metrics Registration

It is neither possible nor desirable to populate the Performance Metrics Registry with all combinations of Parameters of all Performance Metrics. The Registered Performance Metrics SHOULD be:

1. interpretable by the user.
2. implementable by the software or hardware designer,
3. deployable by network operators,
4. accurate in terms of producing equivalent results, and for interoperability and deployment across vendors,
5. Operationally useful, so that it has significant industry interest and/or has seen deployment,
6. Sufficiently tightly defined, so that different values for the Run-time Parameters does not change the fundamental nature of the measurement, nor change the practicality of its implementation.

In essence, there needs to be evidence that a candidate Registered Performance Metric has significant industry interest, or has seen deployment, and there is agreement that the candidate Registered Performance Metric serves its intended purpose.

6. Performance Metric Registry: Prior attempt

There was a previous attempt to define a metric registry RFC 4148 [RFC4148]. However, it was obsoleted by RFC 6248 [RFC6248] because it was "found to be insufficiently detailed to uniquely identify IPPM metrics... [there was too much] variability possible when characterizing a metric exactly" which led to the RFC4148 registry having "very few users, if any".

A couple of interesting additional quotes from RFC 6248 [RFC6248] might help to understand the issues related to that registry.

1. "It is not believed to be feasible or even useful to register every possible combination of Type P, metric parameters, and Stream parameters using the current structure of the IPPM Metrics Registry."
2. "The registry structure has been found to be insufficiently detailed to uniquely identify IPPM metrics."
3. "Despite apparent efforts to find current or even future users, no one responded to the call for interest in the RFC 4148 registry during the second half of 2010."

The current approach learns from this by tightly defining each Registered Performance Metric with only a few variable (Run-time) Parameters to be specified by the measurement designer, if any. The

idea is that entries in the Performance Metrics Registry stem from different measurement methods which require input (Run-time) parameters to set factors like source and destination addresses (which do not change the fundamental nature of the measurement). The downside of this approach is that it could result in a large number of entries in the Performance Metrics Registry. There is agreement that less is more in this context - it is better to have a reduced set of useful metrics rather than a large set of metrics, some with questionable usefulness.

6.1. Why this Attempt Should Succeed

As mentioned in the previous section, one of the main issues with the previous registry was that the metrics contained in the registry were too generic to be useful. This document specifies stricter criteria for performance metric registration (see section 5), and imposes a group of Performance Metrics Experts that will provide guidelines to assess if a Performance Metric is properly specified.

Another key difference between this attempt and the previous one is that in this case there is at least one clear user for the Performance Metrics Registry: the LMAP framework and protocol. Because the LMAP protocol will use the Performance Metrics Registry values in its operation, this actually helps to determine if a metric is properly defined. In particular, since we expect that the LMAP control protocol will enable a controller to request a measurement agent to perform a measurement using a given metric by embedding the Performance Metrics Registry identifier in the protocol. Such a metric and method are properly specified if they are defined well-enough so that it is possible (and practical) to implement them in the measurement agent. This was the failure of the previous attempt: a registry entry with an undefined Type-P (section 13 of RFC 2330 [RFC2330]) allows implementation to be ambiguous.

7. Definition of the Performance Metric Registry

This Performance Metrics Registry is applicable to Performance Metrics used for Active Measurement, Passive Measurement, and any other form of Performance Measurement. Each category of measurement has unique properties, so some of the columns defined below are not applicable for a given metric category. In this case, the column(s) SHOULD be populated with the "NA" value (Non Applicable). However, the "NA" value MUST NOT be used by any metric in the following columns: Identifier, Name, URI, Status, Requester, Revision, Revision Date, Description. In the future, a new category of metrics could require additional columns, and adding new columns is a recognized form of registry extension. The specification defining the new

column(s) MUST give general guidelines for populating the new column(s) for existing entries.

The columns of the Performance Metrics Registry are defined below. The columns are grouped into "Categories" to facilitate the use of the registry. Categories are described at the 7.x heading level, and columns are at the 7.x.y heading level. The Figure below illustrates this organization. An entry (row) therefore gives a complete description of a Registered Performance Metric.

Each column serves as a check-list item and helps to avoid omissions during registration and expert review.

=====

Legend:

Registry Categories and Columns are shown below as:

Category

-----...

Column | Column |...

=====

Summary

Identifier	Name	URI	Desc.	Reference	Change Controller	Ver
------------	------	-----	-------	-----------	-------------------	-----

Metric Definition

Reference Definition	Fixed Parameters
----------------------	------------------

Method of Measurement

Reference Method	Packet Stream Generation	Traffic Filter	Sampling Distribution	Run-time Parameters	Role
------------------	--------------------------	----------------	-----------------------	---------------------	------

Output

Type	Reference Definition	Units	Calibration
------	----------------------	-------	-------------

Administrative Information

Status	Requester	Rev	Rev.Date
--------	-----------	-----	----------

Comments and Remarks

There is a blank template of the Registry template provided in Section 11 of this memo.

7.1. Summary Category

7.1.1. Identifier

A numeric identifier for the Registered Performance Metric. This identifier **MUST** be unique within the Performance Metrics Registry.

The Registered Performance Metric unique identifier is an unbounded integer (range 0 to infinity).

The Identifier 0 should be Reserved. The Identifier values from 64512 to 65536 are reserved for private or experimental use, and the user may encounter overlapping uses.

When adding newly Registered Performance Metrics to the Performance Metrics Registry, IANA **SHOULD** assign the lowest available identifier to the new Registered Performance Metric.

If a Performance Metrics Expert providing review determines that there is a reason to assign a specific numeric identifier, possibly leaving a temporary gap in the numbering, then the Performance Expert **SHALL** inform IANA of this decision.

7.1.2. Name

As the name of a Registered Performance Metric is the first thing a potential human implementor will use when determining whether it is suitable for their measurement study, it is important to be as precise and descriptive as possible. In future, users will review the names to determine if the metric they want to measure has already been registered, or if a similar entry is available as a basis for creating a new entry.

Names are composed of the following elements, separated by an underscore character "_":

MetricType_Method_SubTypeMethod_... Spec_Units_Output

- o MetricType: a combination of the directional properties and the metric measured, such as and not limited to:

- RTDelay (Round Trip Delay)

- RTDNS (Response Time Domain Name Service)

- RLDNS (Response Loss Domain Name Service)

- OWDelay (One Way Delay)

RTLoss (Round Trip Loss)

OWLoss (One Way Loss)

OWPDV (One Way Packet Delay Variation)

OWIPDV (One Way Inter-Packet Delay Variation)

OWReorder (One Way Packet Reordering)

OWDuplic (One Way Packet Duplication)

OWBTC (One Way Bulk Transport Capacity)

OWMBM (One Way Model Based Metric)

SPMonitor (Single Point Monitor)

MPMonitor (Multi-Point Monitor)

- o Method: One of the methods defined in [RFC7799], such as and not limited to:

Active (depends on a dedicated measurement packet stream and observations of the stream)

Passive (depends **solely** on observation of one or more existing packet streams)

HybridType1 (observations on one stream that combine both active and passive methods)

HybridType2 (observations on two or more streams that combine both active and passive methods)

Spatial (Spatial Metric of RFC5644)

- o SubTypeMethod: One or more sub-types to further describe the features of the entry, such as and not limited to:

ICMP (Internet Control Message Protocol)

IP (Internet Protocol)

DSCPxx (where xx is replaced by a Diffserv code point)

UDP (User Datagram Protocol)

TCP (Transport Control Protocol)

QUIC (QUIC transport protocol)

HS (Hand-Shake, such as TCP's 3-way HS)

Poisson (Packet generation using Poisson distribution)

Periodic (Periodic packet generation)

SendOnRcv (Sender keeps one packet in-transit by sending when previous packet arrives)

PayloadxxxxB (where xxxx is replaced by an integer, the number of octets in the Payload))

SustainedBurst (Capacity test, worst case)

StandingQueue (test of bottleneck queue behavior)

SubTypeMethod values are separated by a hyphen "-" character, which indicates that they belong to this element, and that their order is unimportant when considering name uniqueness.

- o Spec: An immutable document identifier combined with a document section identifier. For RFCs, this consists of the RFC number and major section number that specifies this Registry entry in the form RFCXXXXsecY, such as RFC7799sec3. Note: the RFC number is not the Primary Reference specification for the metric definition, such as [RFC7679] for One-way Delay; it will contain the placeholder "RFCXXXXsecY" until the RFC number is assigned to the specifying document, and would remain blank in private registry entries without a corresponding RFC. Anticipating the "RFC10K" problem, the number of the RFC continues to replace RFCXXXX regardless of the number of digits in the RFC number. Anticipating Registry Entries from other standards bodies, the form of this Name Element MUST be proposed and reviewed for consistency and uniqueness by the Expert Reviewer.
- o Units: The units of measurement for the output, such as and not limited to:

Seconds

Ratio (unitless)

Percent (value multiplied by 100%)

Logical (1 or 0)

Packets

BPS (Bits per Second)

PPS (Packets per Second)

EventTotal (for unit-less counts)

Multiple (more than one type of unit)

Enumerated (a list of outcomes)

Unitless

- o Output: The type of output resulting from measurement, such as and not limited to:

Singleton

Raw (multiple Singletons)

Count

Minimum

Maximum

Median

Mean

95Percentile (95th Percentile)

99Percentile (99th Percentile)

StdDev (Standard Deviation)

Variance

PFI (Pass, Fail, Inconclusive)

FlowRecords (descriptions of flows observed)

LossRatio (lost packets to total packets, <=1)

An example is:

RTDelay_Active_IP-UDP-Periodic_RFCXXXXsecY_Seconds_95Percentile

as described in section 4 of [I-D.ietf-ippm-initial-registry].

Note that private registries following the format described here SHOULD use the prefix "Priv_" on any name to avoid unintended conflicts (further considerations are described in section 10). Private registry entries usually have no specifying RFC, thus the Spec: element has no clear interpretation.

7.1.3. URI

The URIs column MUST contain a URL [RFC3986] that uniquely identifies and locates the metric entry so it is accessible through the Internet. The URL points to a file containing all the human-readable information for one registry entry. The URL SHALL reference a target file that is preferably HTML-formatted and contains URLs to referenced sections of HTML-ized RFCs, or other reference specifications. These target files for different entries can be more easily edited and re-used when preparing new entries. The exact form of the URL for each target file, and the target file itself, will be determined by IANA and reside on "iana.org". The major sections of [I-D.ietf-ippm-initial-registry] provide an example of a target file in HTML form (sections 4 and higher).

7.1.4. Description

A Registered Performance Metric description is a written representation of a particular Performance Metrics Registry entry. It supplements the Registered Performance Metric name to help Performance Metrics Registry users select relevant Registered Performance Metrics.

7.1.5. Reference

This entry gives the specification containing the candidate registry entry which was reviewed and agreed, if such an RFC or other specification exists.

7.1.6. Change Controller

This entry names the entity responsible for approving revisions to the registry entry, and SHALL provide contact information (for an individual, where appropriate).

7.1.7. Version (of Registry Format)

This entry gives the version number for the registry format used. Formats complying with this memo MUST use 1.0. The version number SHALL NOT change unless a new RFC is published that changes the registry format. The version number of registry entries SHALL NOT change unless the registry entry is updated (following procedures in section 8).

7.2. Metric Definition Category

This category includes columns to prompt all necessary details related to the metric definition, including the immutable document reference and values of input factors, called fixed parameters, which are left open in the immutable document, but have a particular value defined by the performance metric.

7.2.1. Reference Definition

This entry provides a reference (or references) to the relevant section(s) of the document(s) that define the metric, as well as any supplemental information needed to ensure an unambiguous definition for implementations. The reference needs to be an immutable document, such as an RFC; for other standards bodies, it is likely to be necessary to reference a specific, dated version of a specification.

7.2.2. Fixed Parameters

Fixed Parameters are Parameters whose value must be specified in the Performance Metrics Registry. The measurement system uses these values.

Where referenced metrics supply a list of Parameters as part of their descriptive template, a sub-set of the Parameters will be designated as Fixed Parameters. As an example for active metrics, Fixed Parameters determine most or all of the IPPM Framework convention "packets of Type-P" as described in [RFC2330], such as transport protocol, payload length, TTL, etc. An example for passive metrics is for RTP packet loss calculation that relies on the validation of a packet as RTP which is a multi-packet validation controlled by MIN_SEQUENTIAL as defined by [RFC3550]. Varying MIN_SEQUENTIAL values can alter the loss report and this value could be set as a Fixed Parameter.

Parameters MUST have well-defined names. For human readers, the hanging indent style is preferred, and any Parameter names and

definitions that do not appear in the Reference Method Specification MUST appear in this column (or Run-time Parameters column).

Parameters MUST have a well-specified data format.

A Parameter which is a Fixed Parameter for one Performance Metrics Registry entry may be designated as a Run-time Parameter for another Performance Metrics Registry entry.

7.3. Method of Measurement Category

This category includes columns for references to relevant sections of the immutable document(s) and any supplemental information needed to ensure an unambiguous method for implementations.

7.3.1. Reference Method

This entry provides references to relevant sections of immutable documents, such as RFC(s) (for other standards bodies, it is likely to be necessary to reference a specific, dated version of a specification) describing the method of measurement, as well as any supplemental information needed to ensure unambiguous interpretation for implementations referring to the immutable document text.

Specifically, this section should include pointers to pseudocode or actual code that could be used for an unambiguous implementation.

7.3.2. Packet Stream Generation

This column applies to Performance Metrics that generate traffic as part of their Measurement Method, including but not necessarily limited to Active metrics. The generated traffic is referred as a stream and this column describes its characteristics.

Each entry for this column contains the following information:

- o Value: The name of the packet stream scheduling discipline
- o Reference: the specification where the parameters of the stream are defined

The packet generation stream may require parameters such as the average packet rate and distribution truncation value for streams with Poisson-distributed inter-packet sending times. In case such parameters are needed, they should be included either in the Fixed parameter column or in the run time parameter column, depending on whether they will be fixed or will be an input for the metric.

The simplest example of stream specification is Singleton scheduling (see [RFC2330]), where a single atomic measurement is conducted. Each atomic measurement could consist of sending a single packet (such as a DNS request) or sending several packets (for example, to request a webpage). Other streams support a series of atomic measurements in a "sample", with a schedule defining the timing between each transmitted packet and subsequent measurement. Principally, two different streams are used in IPPM metrics, Poisson distributed as described in [RFC2330] and Periodic as described in [RFC3432]. Both Poisson and Periodic have their own unique parameters, and the relevant set of parameters names and values should be included either in the Fixed Parameters column or in the Run-time parameter column.

7.3.3. Traffic Filter

This column applies to Performance Metrics that observe packets flowing through (the device with) the measurement agent i.e. that is not necessarily addressed to the measurement agent. This includes but is not limited to Passive Metrics. The filter specifies the traffic that is measured. This includes protocol field values/ranges, such as address ranges, and flow or session identifiers.

The traffic filter itself depends on needs of the metric itself and a balance of an operator's measurement needs and a user's need for privacy. Mechanics for conveying the filter criteria might be the BPF (Berkley Packet Filter) or PSAMP [RFC5475] Property Match Filtering which reuses IPFIX [RFC7012]. An example BPF string for matching TCP/80 traffic to remote destination net 192.0.2.0/24 would be "dst net 192.0.2.0/24 and tcp dst port 80". More complex filter engines might be supported by the implementation that might allow for matching using Deep Packet Inspection (DPI) technology.

The traffic filter includes the following information:

Type: the type of traffic filter used, e.g. BPF, PSAMP, OpenFlow rule, etc. as defined by a normative reference

Value: the actual set of rules expressed

7.3.4. Sampling Distribution

The sampling distribution defines out of all the packets that match the traffic filter, which one of those are actually used for the measurement. One possibility is "all" which implies that all packets matching the Traffic filter are considered, but there may be other sampling strategies. It includes the following information:

Value: the name of the sampling distribution

Reference definition: pointer to the specification where the sampling distribution is properly defined.

The sampling distribution may require parameters. In case such parameters are needed, they should be included either in the Fixed parameter column or in the run time parameter column, depending on whether they will be fixed or will be an input for the metric.

Sampling and Filtering Techniques for IP Packet Selection are documented in the PSAMP (Packet Sampling) [RFC5475], while the Framework for Packet Selection and Reporting, [RFC5474] provides more background information. The sampling distribution parameters might be expressed in terms of the Information Model for Packet Sampling Exports, [RFC5477], and the Flow Selection Techniques, [RFC7014].

7.3.5. Run-time Parameters

Run-Time Parameters are Parameters that must be determined, configured into the measurement system, and reported with the results for the context to be complete. However, the values of these parameters is not specified in the Performance Metrics Registry (like the Fixed Parameters), rather these parameters are listed as an aid to the measurement system implementer or user (they must be left as variables, and supplied on execution).

Where metrics supply a list of Parameters as part of their descriptive template, a sub-set of the Parameters will be designated as Run-Time Parameters.

Parameters MUST have well defined names. For human readers, the hanging indent style is preferred, and the names and definitions that do not appear in the Reference Method Specification MUST appear in this column.

A Data Format for each Run-time Parameter MUST be specified in this column, to simplify the control and implementation of measurement devices. For example, parameters that include an IPv4 address can be encoded as a 32 bit integer (i.e. binary base64 encoded value) or ip-address as defined in [RFC6991]. The actual encoding(s) used must be explicitly defined for each Run-time parameter. IPv6 addresses and options MUST be accommodated, allowing Registered Metrics to be used in that address family. Other address families are permissable.

Examples of Run-time Parameters include IP addresses, measurement point designations, start times and end times for measurement, and other information essential to the method of measurement.

7.3.6. Role

In some methods of measurement, there may be several roles defined, e.g., for a one-way packet delay active measurement there is one measurement agent that generates the packets and another agent that receives the packets. This column contains the name of the Role(s) for this particular entry. In the one-way delay example above, there should be two entries in the Role registry column, one for each Role (Source and Destination). When a measurement agent is instructed to perform the "Source" Role for one-way delay metric, the agent knows that it is required to generate packets. The values for this field are defined in the reference method of measurement (and this frequently results in abbreviated role names such as "Src").

When the Role column of a registry entry defines more than one Role, then the Role SHALL be treated as a Run-time Parameter and supplied for execution. It should be noted that the LMAP framework [RFC7594] distinguishes the Role from other Run-time Parameters, and defines a special parameter "Roles" inside the registry-grouping function list in the LMAP YANG model[RFC8194].

7.4. Output Category

For entries which involve a stream and many singleton measurements, a statistic may be specified in this column to summarize the results to a single value. If the complete set of measured singletons is output, this will be specified here.

Some metrics embed one specific statistic in the reference metric definition, while others allow several output types or statistics.

7.4.1. Type

This column contains the name of the output type. The output type defines a single type of result that the metric produces. It can be the raw results (packet send times and singleton metrics), or it can be a summary statistic. The specification of the output type MUST define the format of the output. In some systems, format specifications will simplify both measurement implementation and collection/storage tasks. Note that if two different statistics are required from a single measurement (for example, both "Xth percentile mean" and "Raw"), then a new output type must be defined ("Xth percentile mean AND Raw"). See the Naming section above for a list of Output Types.

7.4.2. Reference Definition

This column contains a pointer to the specification(s) where the output type and format are defined.

7.4.3. Metric Units

The measured results must be expressed using some standard dimension or units of measure. This column provides the units.

When a sample of singletons (see Section 11 of [RFC2330] for definitions of these terms) is collected, this entry will specify the units for each measured value.

7.4.4. Calibration

Some specifications for Methods of Measurement include the possibility to perform an error calibration. Section 3.7.3 of [RFC7679] is one example. In the registry entry, this field will identify a method of calibration for the metric, and when available, the measurement system SHOULD perform the calibration when requested and produce the output with an indication that it is the result of a calibration method. In-situ calibration could be enabled with an internal loopback that includes as much of the measurement system as possible, performs address manipulation as needed, and provides some form of isolation (e.g., deterministic delay) to avoid send-receive interface contention. Some portion of the random and systematic error can be characterized this way.

For one-way delay measurements, the error calibration must include an assessment of the internal clock synchronization with its external reference (this internal clock is supplying timestamps for measurement). In practice, the time offsets of clocks at both the source and destination are needed to estimate the systematic error due to imperfect clock synchronization (the time offsets are smoothed, thus the random variation is not usually represented in the results).

Both internal loopback calibration and clock synchronization can be used to estimate the *available accuracy* of the Output Metric Units. For example, repeated loopback delay measurements will reveal the portion of the Output result resolution which is the result of system noise, and thus inaccurate.

7.5. Administrative information

7.5.1. Status

The status of the specification of this Registered Performance Metric. Allowed values are 'current' and 'deprecated'. All newly defined Information Elements have 'current' status.

7.5.2. Requester

The requester for the Registered Performance Metric. The requester MAY be a document, such as RFC, or person.

7.5.3. Revision

The revision number of a Registered Performance Metric, starting at 0 for Registered Performance Metrics at time of definition and incremented by one for each revision.

7.5.4. Revision Date

The date of acceptance or the most recent revision for the Registered Performance Metric. The date SHALL be determined by IANA and the reviewing Performance Metrics Expert.

7.6. Comments and Remarks

Besides providing additional details which do not appear in other categories, this open Category (single column) allows for unforeseen issues to be addressed by simply updating this informational entry.

8. Processes for Managing the Performance Metric Registry Group

Once a Performance Metric or set of Performance Metrics has been identified for a given application, candidate Performance Metrics Registry entry specifications prepared in accordance with Section 7 should be submitted to IANA to follow the process for review by the Performance Metric Experts, as defined below. This process is also used for other changes to the Performance Metrics Registry, such as deprecation or revision, as described later in this section.

It is desirable that the author(s) of a candidate Performance Metrics Registry entry seek review in the relevant IETF working group, or offer the opportunity for review on the working group mailing list.

8.1. Adding new Performance Metrics to the Performance Metrics Registry

Requests to add Registered Performance Metrics in the Performance Metrics Registry SHALL be submitted to IANA, which forwards the request to a designated group of experts (Performance Metric Experts) appointed by the IESG; these are the reviewers called for by the Specification Required [RFC8126] policy defined for the Performance Metrics Registry. The Performance Metric Experts review the request for such things as compliance with this document, compliance with other applicable Performance Metric-related RFCs, and consistency with the currently defined set of Registered Performance Metrics. The most efficient path for submission begins with preparation of an Internet Draft containing the proposed Performance Metrics Registry entry using the template in Section 11, so that the submission formatting will benefit from the normal IETF Internet Draft submission processing (including HTML-ization).

Submission to IANA may be during IESG review (leading to IETF Standards Action), where an Internet Draft proposes one or more Registered Performance Metrics to be added to the Performance Metrics Registry, including the text of the proposed Registered Performance Metric(s).

If an RFC-to-be includes a Performance Metric and a proposed Performance Metrics Registry entry, but the Performance Metric Expert review determines that one or more of the Section 5 criteria have not been met, then the proposed Performance Metrics Registry entry MUST be removed from the text. Once evidence exists that the Performance Metric meets the criteria in section 5, the proposed Performance Metrics Registry entry SHOULD be submitted to IANA to be evaluated in consultation with the Performance Metric Experts for registration at that time.

Authors of proposed Registered Performance Metrics SHOULD review compliance with the specifications in this document to check their submissions before sending them to IANA.

At least one Performance Metric Expert should endeavor to complete referred reviews in a timely manner. If the request is acceptable, the Performance Metric Experts signify their approval to IANA, and IANA updates the Performance Metrics Registry. If the request is not acceptable, the Performance Metric Experts MAY coordinate with the requester to change the request to be compliant, otherwise IANA SHALL coordinate resolution of issues on behalf of the expert. The Performance Metric Experts MAY choose to reject clearly frivolous or inappropriate change requests outright, but such exceptional circumstances should be rare.

This process should not in any way be construed as allowing the Performance Metric Experts to overrule IETF consensus. Specifically, any Registered Performance Metrics that were added to the Performance Metrics Registry with IETF consensus require IETF consensus for revision or deprecation.

Decisions by the Performance Metric Experts may be appealed as in Section 7 of [RFC8126].

8.2. Revising Registered Performance Metrics

A request for Revision is only permitted when the requested changes maintain backward-compatibility with implementations of the prior Performance Metrics Registry entry describing a Registered Performance Metric (entries with lower revision numbers, but the same Identifier and Name).

The purpose of the Status field in the Performance Metrics Registry is to indicate whether the entry for a Registered Performance Metric is 'current' or 'deprecated'.

In addition, no policy is defined for revising the Performance Metric entries in the IANA Registry or addressing errors therein. To be clear, changes and deprecations within the Performance Metrics Registry are not encouraged, and should be avoided to the extent possible. However, in recognition that change is inevitable, the provisions of this section address the need for revisions.

Revisions are initiated by sending a candidate Registered Performance Metric definition to IANA, as in Section 8.1, identifying the existing Performance Metrics Registry entry, and explaining how and why the existing entry should be revised.

The primary requirement in the definition of procedures for managing changes to existing Registered Performance Metrics is avoidance of measurement interoperability problems; the Performance Metric Experts must work to maintain interoperability above all else. Changes to Registered Performance Metrics may only be done in an interoperable way; necessary changes that cannot be done in a way to allow interoperability with unchanged implementations MUST result in the creation of a new Registered Performance Metric (with a new Name, replacing the RFCXXXXsecY portion of the name) and possibly the deprecation of the earlier metric.

A change to a Registered Performance Metric SHALL be determined to be backward-compatible when:

1. it involves the correction of an error that is obviously only editorial; or
2. it corrects an ambiguity in the Registered Performance Metric's definition, which itself leads to issues severe enough to prevent the Registered Performance Metric's usage as originally defined; or
3. it corrects missing information in the metric definition without changing its meaning (e.g., the explicit definition of 'quantity' semantics for numeric fields without a Data Type Semantics value); or
4. it harmonizes with an external reference that was itself corrected.

If a Performance Metric revision is deemed permissible and backward-compatible by the Performance Metric Experts, according to the rules in this document, IANA SHOULD execute the change(s) in the Performance Metrics Registry. The requester of the change is appended to the original requester in the Performance Metrics Registry. The Name of the revised Registered Performance Metric, including the RFCXXXsecY portion of the name, SHALL remain unchanged (even when the change is the result of IETF Standards Action; the revised registry entry SHOULD reference the new immutable document, such as an RFC or for other standards bodies, it is likely to be necessary to reference a specific, dated version of a specification, in an appropriate category and column).

Each Registered Performance Metric in the Performance Metrics Registry has a revision number, starting at zero. Each change to a Registered Performance Metric following this process increments the revision number by one.

When a revised Registered Performance Metric is accepted into the Performance Metrics Registry, the date of acceptance of the most recent revision is placed into the revision Date column of the registry for that Registered Performance Metric.

Where applicable, additions to Registered Performance Metrics in the form of text Comments or Remarks should include the date, but such additions may not constitute a revision according to this process.

Older version(s) of the updated metric entries are kept in the registry for archival purposes. The older entries are kept with all fields unmodified (version, revision date) except for the status field that SHALL be changed to "Deprecated".

8.3. Deprecating Registered Performance Metrics

Changes that are not permissible by the above criteria for Registered Performance Metric's revision may only be handled by deprecation. A Registered Performance Metric MAY be deprecated and replaced when:

1. the Registered Performance Metric definition has an error or shortcoming that cannot be permissibly changed as in Section 8.2 Revising Registered Performance Metrics; or
2. the deprecation harmonizes with an external reference that was itself deprecated through that reference's accepted deprecation method.

A request for deprecation is sent to IANA, which passes it to the Performance Metric Experts for review. When deprecating an Performance Metric, the Performance Metric description in the Performance Metrics Registry must be updated to explain the deprecation, as well as to refer to any new Performance Metrics created to replace the deprecated Performance Metric.

The revision number of a Registered Performance Metric is incremented upon deprecation, and the revision Date updated, as with any revision.

The intentional use of deprecated Registered Performance Metrics should result in a log entry or human-readable warning by the respective application.

Names and Metric IDs of deprecated Registered Performance Metrics must not be reused.

The deprecated entries are kept with all fields unmodified, except the version, revision date, and the status field (changed to "Deprecated").

9. Security considerations

This draft defines a registry structure, and does not itself introduce any new security considerations for the Internet. The definition of Performance Metrics for this registry may introduce some security concerns, but the mandatory references should have their own considerations for security, and such definitions should be reviewed with security in mind if the security considerations are not covered by one or more reference standards.

The aggregated results of the performance metrics described in this registry might reveal network topology information that may be

considered sensitive. If such cases are found, then access control mechanisms should be applied.

10. IANA Considerations

With the background and processes described in earlier sections, this document requests the following IANA Actions.

Editor's Note: Mock-ups of the implementation of this set of requests have been prepared with IANA's help during development of this memo, and have been captured in the Proceedings of IPPM working group sessions. IANA is currently preparing a mock-up. A recent version is available here: <http://encrypted.net/IETFMetricsRegistry-106.html>

10.1. Registry Group

The new registry group SHALL be named, "PERFORMANCE METRICS Group".

Registration Procedure: Specification Required

Reference: <This RFC>

Experts: Performance Metrics Experts

Note: TBD

10.2. Performance Metric Name Elements

This document specifies the procedure for Performance Metrics Name Element Registry setup. IANA is requested to create a new set of registries for Performance Metric Name Elements called "Registered Performance Metric Name Elements". Each Registry, whose names are listed below:

MetricType:

Method:

SubTypeMethod:

Spec:

Units:

Output:

will contain the current set of possibilities for Performance Metrics Registry Entry Names.

To populate the Registered Performance Metric Name Elements at creation, the IANA is asked to use the lists of values for each name element listed in Section 7.1.2. The Name Elements in each registry are case-sensitive.

When preparing a Metric entry for Registration, the developer SHOULD choose Name elements from among the registered elements. However, if the proposed metric is unique in a significant way, it may be necessary to propose a new Name element to properly describe the metric, as described below.

A candidate Metric Entry RFC or immutable document for IANA and Expert Review would propose one or more new element values required to describe the unique entry, and the new name element(s) would be reviewed along with the metric entry. New assignments for Registered Performance Metric Name Elements will be administered by IANA through Specification Required policy (which includes Expert Review) [RFC8126], i.e., review by one of a group of experts, the Performance Metric Experts, who are appointed by the IESG upon recommendation of the Transport Area Directors.

10.3. New Performance Metrics Registry

This document specifies the procedure for Performance Metrics Registry setup. IANA is requested to create a new registry for Performance Metrics called "Performance Metrics Registry". This Registry will contain the following Summary columns:

Identifier:

Name:

URI:

Description:

Reference:

Change Controller:

Version:

Descriptions of these columns and additional information found in the template for registry entries (categories and columns) are further defined in section Section 7.

The Identifier 0 should be Reserved. The Registered Performance Metric unique identifier is an unbounded integer (range 0 to

infinity). The Identifier values from 64512 to 65536 are reserved for private or experimental use, and the user may encounter overlapping uses. When adding newly Registered Performance Metrics to the Performance Metrics Registry, IANA SHOULD assign the lowest available identifier to the new Registered Performance Metric. If a Performance Metrics Expert providing review determines that there is a reason to assign a specific numeric identifier, possibly leaving a temporary gap in the numbering, then the Performance Expert SHALL inform IANA of this decision.

Names starting with the prefix Priv_ are reserved for private use, and are not considered for registration. The "Name" column entries are further defined in section Section 7.

The "URI" column will have a URL to the full template of each registry entry. The Registry Entry text SHALL be HTML-ized to aid the reader, with links to reference RFCs (similar to the way that Internet Drafts are HTML-ized, the same tool can perform the function) or immutable document.

The "Reference" column will include an RFC number, an approved specification designator from another standards body, or other immutable document.

New assignments for Performance Metrics Registry will be administered by IANA through Specification Required policy (which includes Expert Review) [RFC8126], i.e., review by one of a group of experts, the Performance Metric Experts, who are appointed by the IESG upon recommendation of the Transport Area Directors, or by Standards Action. The experts can be initially drawn from the Working Group Chairs, document editors, and members of the Performance Metrics Directorate, among other sources of experts.

Extensions of the Performance Metrics Registry require IETF Standards Action. Only one form of registry extension is envisaged:

1. Adding columns, or both categories and columns, to accommodate unanticipated aspects of new measurements and metric categories.

If the Performance Metrics Registry is extended in this way, the Version number of future entries complying with the extension SHALL be incremented (either in the unit or tenths digit, depending on the degree of extension).

11. Blank Registry Template

This section provides a blank template to help IANA and registry entry writers.

11.1. Summary

This category includes multiple indexes to the registry entry: the element ID and metric name.

11.1.1. ID (Identifier)

<insert a numeric identifier, an integer, TBD>

11.1.2. Name

<insert name according to metric naming convention>

11.1.3. URI

URL: <https://www.iana.org/> ... <name>

11.1.4. Description

<provide a description>

11.1.5. Change Controller

11.1.6. Version (of Registry Format)

11.2. Metric Definition

This category includes columns to prompt the entry of all necessary details related to the metric definition, including the immutable document reference and values of input factors, called fixed parameters.

11.2.1. Reference Definition

<Full bibliographic reference to an immutable doc.>

<specific section reference and additional clarifications, if needed>

11.2.2. Fixed Parameters

<list and specify Fixed Parameters, input factors that must be determined and embedded in the measurement system for use when needed>

11.3. Method of Measurement

This category includes columns for references to relevant sections of the immutable documents(s) and any supplemental information needed to ensure an unambiguous methods for implementations.

11.3.1. Reference Method

<for metric, insert relevant section references and supplemental info>

11.3.2. Packet Stream Generation

<list of generation parameters and section/spec references if needed>

11.3.3. Traffic Filtering (observation) Details

The measured results based on a filtered version of the packets observed, and this section provides the filter details (when present).

<section reference>.

11.3.4. Sampling Distribution

<insert time distribution details, or how this is diff from the filter>

11.3.5. Run-time Parameters and Data Format

Run-time Parameters are input factors that must be determined, configured into the measurement system, and reported with the results for the context to be complete.

<list of run-time parameters, and their data formats>

11.3.6. Roles

<lists the names of the different roles from the measurement method>

11.4. Output

This category specifies all details of the Output of measurements using the metric.

11.4.1. Type

<insert name of the output type, raw or a selected summary statistic>

11.4.2. Reference Definition

<describe the reference data format for each type of result>

11.4.3. Metric Units

<insert units for the measured results, and the reference specification>.

11.4.4. Calibration

<insert information on calibration>

11.5. Administrative items

11.5.1. Status

<current or deprecated>

11.5.2. Requester

<name or RFC, etc.>

11.5.3. Revision

<1.0>

11.5.4. Revision Date

<format YYYY-MM-DD>

11.6. Comments and Remarks

<Additional (Informational) details for this entry>

12. Acknowledgments

Thanks to Brian Trammell and Bill Cervený, IPPM chairs, for leading some brainstorming sessions on this topic. Thanks to Barbara Stark and Juergen Schoenwaelder for the detailed feedback and suggestions. Thanks to Andrew McGregor for suggestions on metric naming. Thanks to Michelle Cotton for her early IANA review, and to Amanda Barber for answering questions related to the presentation of the registry and accessibility of the complete template via URL. Thanks to Roni

Even for his review and suggestions to generalize the procedures.
Thanks to ~all the Area Directors for their reviews.

13. References

13.1. Normative References

- [RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, DOI 10.17487/RFC2026, October 1996, <<https://www.rfc-editor.org/info/rfc2026>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, DOI 10.17487/RFC2330, May 1998, <<https://www.rfc-editor.org/info/rfc2330>>.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <<https://www.rfc-editor.org/info/rfc3986>>.
- [RFC6390] Clark, A. and B. Claise, "Guidelines for Considering New Performance Metric Development", BCP 170, RFC 6390, DOI 10.17487/RFC6390, October 2011, <<https://www.rfc-editor.org/info/rfc6390>>.
- [RFC6576] Geib, R., Ed., Morton, A., Fardid, R., and A. Steinmitz, "IP Performance Metrics (IPPM) Standard Advancement Testing", BCP 176, RFC 6576, DOI 10.17487/RFC6576, March 2012, <<https://www.rfc-editor.org/info/rfc6576>>.
- [RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <<https://www.rfc-editor.org/info/rfc7799>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

13.2. Informative References

- [I-D.ietf-ippm-initial-registry]
Morton, A., Bagnulo, M., Eardley, P., and K. D'Souza,
"Initial Performance Metrics Registry Entries", draft-
ietf-ippm-initial-registry-15 (work in progress), December
2019.
- [RFC2681] Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip
Delay Metric for IPPM", RFC 2681, DOI 10.17487/RFC2681,
September 1999, <<https://www.rfc-editor.org/info/rfc2681>>.
- [RFC3432] Raisanen, V., Grotefeld, G., and A. Morton, "Network
performance measurement with periodic streams", RFC 3432,
DOI 10.17487/RFC3432, November 2002,
<<https://www.rfc-editor.org/info/rfc3432>>.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V.
Jacobson, "RTP: A Transport Protocol for Real-Time
Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550,
July 2003, <<https://www.rfc-editor.org/info/rfc3550>>.
- [RFC3611] Friedman, T., Ed., Caceres, R., Ed., and A. Clark, Ed.,
"RTP Control Protocol Extended Reports (RTCP XR)",
RFC 3611, DOI 10.17487/RFC3611, November 2003,
<<https://www.rfc-editor.org/info/rfc3611>>.
- [RFC4148] Stephan, E., "IP Performance Metrics (IPPM) Metrics
Registry", BCP 108, RFC 4148, DOI 10.17487/RFC4148, August
2005, <<https://www.rfc-editor.org/info/rfc4148>>.
- [RFC5474] Duffield, N., Ed., Chiou, D., Claise, B., Greenberg, A.,
Grossglauser, M., and J. Rexford, "A Framework for Packet
Selection and Reporting", RFC 5474, DOI 10.17487/RFC5474,
March 2009, <<https://www.rfc-editor.org/info/rfc5474>>.
- [RFC5475] Zseby, T., Molina, M., Duffield, N., Niccolini, S., and F.
Raspall, "Sampling and Filtering Techniques for IP Packet
Selection", RFC 5475, DOI 10.17487/RFC5475, March 2009,
<<https://www.rfc-editor.org/info/rfc5475>>.
- [RFC5477] Dietz, T., Claise, B., Aitken, P., Dressler, F., and G.
Carle, "Information Model for Packet Sampling Exports",
RFC 5477, DOI 10.17487/RFC5477, March 2009,
<<https://www.rfc-editor.org/info/rfc5477>>.

- [RFC6035] Pendleton, A., Clark, A., Johnston, A., and H. Sinnreich, "Session Initiation Protocol Event Package for Voice Quality Reporting", RFC 6035, DOI 10.17487/RFC6035, November 2010, <<https://www.rfc-editor.org/info/rfc6035>>.
- [RFC6248] Morton, A., "RFC 4148 and the IP Performance Metrics (IPPM) Registry of Metrics Are Obsolete", RFC 6248, DOI 10.17487/RFC6248, April 2011, <<https://www.rfc-editor.org/info/rfc6248>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7012] Claise, B., Ed. and B. Trammell, Ed., "Information Model for IP Flow Information Export (IPFIX)", RFC 7012, DOI 10.17487/RFC7012, September 2013, <<https://www.rfc-editor.org/info/rfc7012>>.
- [RFC7014] D'Antonio, S., Zseby, T., Henke, C., and L. Peluso, "Flow Selection Techniques", RFC 7014, DOI 10.17487/RFC7014, September 2013, <<https://www.rfc-editor.org/info/rfc7014>>.
- [RFC7594] Eardley, P., Morton, A., Bagnulo, M., Burbridge, T., Aitken, P., and A. Akhter, "A Framework for Large-Scale Measurement of Broadband Performance (LMAP)", RFC 7594, DOI 10.17487/RFC7594, September 2015, <<https://www.rfc-editor.org/info/rfc7594>>.
- [RFC7679] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <<https://www.rfc-editor.org/info/rfc7679>>.
- [RFC8194] Schoenwaelder, J. and V. Bajpai, "A YANG Data Model for LMAP Measurement Agents", RFC 8194, DOI 10.17487/RFC8194, August 2017, <<https://www.rfc-editor.org/info/rfc8194>>.

Authors' Addresses

Marcelo Bagnulo
Universidad Carlos III de Madrid
Av. Universidad 30
Leganes, Madrid 28911
SPAIN

Phone: 34 91 6249500
Email: marcelo@it.uc3m.es
URI: <http://www.it.uc3m.es>

Benoit Claise
Cisco Systems, Inc.
De Kleetlaan 6a b1
1831 Diegem
Belgium

Email: bclaise@cisco.com

Philip Eardley
BT
Adastral Park, Martlesham Heath
Ipswich
ENGLAND

Email: philip.eardley@bt.com

Al Morton
AT&T Labs
200 Laurel Avenue South
Middletown, NJ
USA

Email: acmorton@att.com

Aamer Akhter
Consultant
118 Timber Hitch
Cary, NC
USA

Email: aakhter@gmail.com

Network Working Group
Internet-Draft
Updates: 2330 (if approved)
Intended status: Standards Track
Expires: February 14, 2021

J. Alvarez-Hamelin
Universidad de Buenos Aires
A. Morton
AT&T Labs
J. Fabini
TU Wien
C. Pignataro
Cisco Systems, Inc.
R. Geib
Deutsche Telekom
August 13, 2020

Advanced Unidirectional Route Assessment (AURA)
draft-ietf-ippm-route-10

Abstract

This memo introduces an advanced unidirectional route assessment (AURA) metric and associated measurement methodology, based on the IP Performance Metrics (IPPM) Framework RFC 2330. This memo updates RFC 2330 in the areas of path-related terminology and path description, primarily to include the possibility of parallel subpaths between a given Source and Destination pair, owing to the presence of multi-path technologies.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on February 14, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Issues with Earlier Work to Define a Route Metric	3
1.2. Requirements Language	4
2. Scope	4
3. Route Metric Specifications	5
3.1. Terms and Definitions	5
3.2. Formal Name	6
3.3. Parameters	6
3.4. Metric Definitions	7
3.5. Related Round-Trip Delay and Loss Definitions	9
3.6. Discussion	10
3.7. Reporting the Metric	10
4. Route Assessment Methodologies	11
4.1. Active Methodologies	11
4.1.1. Temporal Composition for Route Metrics	13
4.1.2. Routing Class Identification	15
4.1.3. Intermediate Observation Point Route Measurement	16
4.2. Hybrid Methodologies	16
4.3. Combining Different Methods	17
5. Background on Round-Trip Delay Measurement Goals	17
6. RTD Measurements Statistics	18
7. Security Considerations	20
8. IANA Considerations	21
9. Acknowledgements	21
10. Appendix I MPLS Methods for Route Assessment	21
11. References	22
11.1. Normative References	22
11.2. Informative References	24
Authors' Addresses	26

1. Introduction

The IETF IP Performance Metrics (IPPM) working group first created a framework for metric development in [RFC2330]. This framework has stood the test of time and enabled development of many fundamental metrics. It has been updated in the area of metric composition

[RFC5835], and in several areas related to active stream measurement of modern networks with reactive properties [RFC7312].

The [RFC2330] framework motivated the development of "performance and reliability metrics for paths through the Internet," and Section 5 of [RFC2330] defines terms that support description of a path under test. However, metrics for assessment of paths and related performance aspects had not been attempted in IPPM when the [RFC2330] framework was written.

This memo takes up the route measurement challenge and specifies a new route metric, two practical frameworks for methods of measurement (using either active or hybrid active-passive methods [RFC7799]), and Round-Trip Delay and link information discovery using the results of measurements. All route measurements are limited by the willingness of hosts along the path to be discovered, to cooperate with the methods used, or to recognize that the measurement operation is taking place (such as when tunnels are present).

1.1. Issues with Earlier Work to Define a Route Metric

Section 7 of [RFC2330] presented a simple example of a "route" metric along with several other examples. The example is reproduced below (where the reference is to Section 5 of [RFC2330]):

"route: The path, as defined in Section 5, from A to B at a given time."

This example provides a starting point to develop a more complete definition of route. Areas needing clarification include:

Time: In practice, the route will be assessed over a time interval, because active path detection methods like Paris Traceroute [PT] rely on hop limits for their operation and cannot accomplish discovery of all hosts using a single packet.

Type-P: The legacy route definition lacks the option to cater for packet-dependent routing. In this memo, we assess the route for a specific packet of Type-P, and reflect this in the metric definition. The methods of measurement determine the specific Type-P used.

Parallel Paths: Parallel paths are a reality of the Internet and a strength of advanced route assessment methods, so the metric must acknowledge this possibility. Use of Equal Cost Multi-Path (ECMP) and Unequal Cost Multi-Path (UCMP) technologies are common sources of parallel subpaths.

Cloud Subpath: May contain hosts that do not decrement hop limit, but may have two or more exchange links connecting "discoverable" hosts or routers. Parallel subpaths contained within clouds cannot be discovered. The assessment methods only discover hosts or routers on the path that decrement hop limit, or cooperate with interrogation protocols. The presence of tunnels and nested tunnels further complicate assessment by hiding hops.

Hop: Although the [RFC2330] definition of a hop was a link-host pair, only hosts that are discoverable or have the capability to cooperate with interrogation protocols where link information may be exposed.

The refined definition of Route metrics begins in the sections that follow.

1.2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14[RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Scope

The purpose of this memo is to add new route metrics and methods of measurement to the existing set of IPPM metrics.

The scope is to define route metrics that can identify the path taken by a packet or a flow traversing the Internet between two hosts. Although primarily intended for hosts communicating on the Internet, the definitions and metrics are constructed to be applicable to other network domains, if desired. The methods of measurement to assess the path may not be able to discover all hosts comprising the path, but such omissions are often deterministic and explainable sources of error.

This memo also specifies a framework for active methods of measurement which uses the techniques described in [PT], as well as a framework for hybrid active-passive methods of measurement such as the Hybrid Type I method [RFC7799] described in [I-D.ietf-ippm-ioam-data]. Methods using [I-D.ietf-ippm-ioam-data] are intended only for single administrative domains that provide a protocol for explicit interrogation of nodes on a path. Combinations of active methods and hybrid active-passive methods are also in-scope.

Further, this memo provides additional analysis of the round-trip delay measurements made possible by the methods, in an effort to discover more details about the path, such as the link technology in use.

This memo updates Section 5 of [RFC2330] in the areas of path-related terminology and path description, primarily to include the possibility of parallel subpaths between a given Source and Destination address pair (possibly resulting from Equal Cost Multi-Path (ECMP) and Unequal Cost Multi-Path (UCMP) technologies).

There are several simple non-goals of this memo. There is no attempt to assess the reverse path from any host on the path to the host attempting the path measurement. The reverse path contribution to delay will be that experienced by ICMP packets (in active methods), and may be different from delays experienced by UDP or TCP packets. Also, the round trip delay will include an unknown contribution of processing time at the host that generates the ICMP response. Therefore, the ICMP-based active methods are not supposed to yield accurate, reproducible estimations of the Round-Trip Delay that UDP or TCP packets will experience.

3. Route Metric Specifications

This section sets requirements for the components of the Route Metric.

3.1. Terms and Definitions

Host A Host as defined in [RFC2330] (a computer capable of IP communication, includes routers), a.k.a. RFC 2330 Host.

Node A Node is any network function on the path capable of IP-layer Communication, includes RFC 2330 Hosts.

Node Identity The unique address for Nodes communicating within the network domain. For Nodes communicating on the Internet with IP, it is the globally routable IP address which the Node uses when communicating with other Nodes under normal or error conditions. The Node Identity revealed (and its connection to a Node Name through reverse DNS) determines whether interfaces to parallel links can be associated with a single Node, or appear to identify unique Nodes.

Discoverable Node Nodes that convey their Node Identity according to the requirements of their network domain, such as when error conditions are detected by that Node. For Nodes communicating with IP packets, compliance with Section 3.2.2.4 of [RFC1122] when

discarding a packet due to TTL or Hop Limit Exceeded condition, MUST result in sending the corresponding Time Exceeded message (containing a form of Node identity) to the source. This requirement is also consistent with section 5.3.1 of [RFC1812] for routers.

Cooperating Node Nodes that respond to direct queries for their Node identity as part of a previously agreed and established interrogation protocol. Nodes SHOULD also provide information such as arrival/departure interface identification, arrival timestamp, and any relevant information about the Node or specific link which delivered the query to the Node.

Hop specification A Hop specification MUST contain a Node Identity, and MAY contain arrival and/or departure interface identification, round trip delay, and an arrival timestamp.

Routing Class A route that treats equally a class of different types of packets, designated "C" (unrelated to address classes of the past) [RFC2330] [RFC8468]. Knowledge of such a class allows any one of the types of packets within that class to be used for subsequent measurement of the route. The designator "class C" is used for historical reasons, see [RFC2330].

3.2. Formal Name

The formal name of the metric is:

Type-P-Route-Ensemble-Method-Variant

abbreviated as Route Ensemble.

Note that Type-P depends heavily on the chosen method and variant.

3.3. Parameters

This section lists the REQUIRED input factors to define and measure a Route metric, as specified in this memo.

- o Src, the address of a Node (such as the globally routable IP address).
- o Dst, the address of a Node (such as the globally routable IP address).
- o i, the limit on the number of Hops a specific packet may visit as it traverses from the Node at Src to the Node at Dst (such as the TTL or Hop Limit).

- o MaxHops, the maximum value of i used, ($i=1,2,3,\dots\text{MaxHops}$).
- o T_0 , a time (start of measurement interval)
- o T_f , a time (end of measurement interval)
- o $\text{MP}(\text{address})$, Measurement Point at address, such as Src or Dst, usually at the same node stack layer as "address".
- o T , the Node time of a packet as measured at $\text{MP}(\text{Src})$, meaning Measurement Point at the Source.
- o T_a , the Node time of a reply packet's *arrival* as measured at $\text{MP}(\text{Src})$, assigned to packets that arrive within a "reasonable" time (see parameter below).
- o T_{max} , a maximum waiting time for reply packets to return to the source, set sufficiently long to disambiguate packets with long delays from packets that are discarded (lost), such that the distribution of Round-Trip Delay is not truncated.
- o F , the number of different flows simulated by the method and variant.
- o flow, the stream of packets with the same n -tuple of designated header fields that (when held constant) result in identical treatment in a multi-path decision (such as the decision taken in load balancing). Note: The IPv6 flow label MAY be included in the flow definition if the $\text{MP}(\text{Src})$ is a Tunnel End Point (TEP) complying with [RFC6438] guidelines.
- o Type-P, the complete description of the packets for which this assessment applies (including the flow-defining fields).

3.4. Metric Definitions

This section defines the REQUIRED measurement components of the Route metrics (unless otherwise indicated):

M , the total number of packets sent between T_0 and T_f .

N , the smallest value of i needed for a packet to be received at Dst (sent between T_0 and T_f).

N_{max} , the largest value of i needed for a packet to be received at Dst (sent between T_0 and T_f). N_{max} may be equal to N .

Next define a **singleton** definition for a Node on the path, with sufficient indexes to identify all Nodes identified in a measurement interval (where **singleton** is part of the IPPM Framework [RFC2330]).

A Hop Specification, designated $h(i,j)$, the IP address and/or identity of Discoverable Nodes (or Cooperating Nodes) that are i hops away from the Node with address = Src and part of Route j during the measurement interval, T_0 to T_f . As defined here, a Hop singleton measurement MUST contain a Node Identity, $hid(i,j)$, and MAY contain one or more of the following attributes:

- o $a(i,j)$ Arrival Interface ID (e.g., when [RFC5837] is supported)
- o $d(i,j)$ Departure Interface ID (e.g., when [RFC5837] is supported)
- o $t(i,j)$ Arrival Timestamp, where $t(i,j)$ is ideally supplied by the Hop. (Note that $t(i,j)$ might be approximated from the sending time of the packet that revealed the Hop, e.g., when the round trip response time is available and divided by 2.)
- o Measurements of Round-Trip Delay (for each packet that reveals the same Node Identity and flow attributes, then this attribute is computed, see next section)

Node Identities and related information can be ordered by their distance from the Node with address Src in Hops $h(i,j)$. Based on this, two forms of Routes are distinguished:

A Route Ensemble is defined as the combination of all routes traversed by different flows from the Node at Src address to the Node at Dst address. A single Route traversed by a single flow (determined by an unambiguous tuple of addresses Src and Dst, and other identical flow criteria) is a member of the Route Ensemble and called a Member Route.

Using $h(i,j)$ and components and parameters, further define:

When considering the set of Hops in the context of a single flow, a Member Route j is an ordered list $\{h(1,j), \dots, h(N_j, j)\}$ where $h(i-1, j)$ and $h(i, j)$ are 1 hop away from each other and N_j satisfying $h(N_j, j) = \text{Dst}$ is the minimum count of Hops needed by the packet on Member Route j to reach Dst. Member Routes must be unique. The uniqueness property requires that any two Member routes j and k that are part of the same Route Ensemble differ either in terms of minimum hop count N_j and N_k to reach the destination Dst, or, in the case of identical hop count $N_j = N_k$, they have at least one distinct Hop: $h(i, j) \neq h(i, k)$ for at least one i ($i=1..N_j$).

All the optional information collected to describe a Member Route, such as the arrival interface, departure interface, and Round Trip Delay at each Hop, turns each list item into a rich structure. There may be information on the links between Hops, possibly information on the routing (arrival interface and departure interface), an estimate of distance between Hops based on Round-Trip Delay measurements and calculations, and a time stamp indicating when all these additional details were valid.

The Route Ensemble from Src to Dst, during the measurement interval T_0 to T_f , is the aggregate of all m distinct Member Routes discovered between the two Nodes with Src and Dst addresses. More formally, with the Node having address Src omitted:

```
Route Ensemble = {
{h(1,1), h(2,1), h(3,1), ... h(N1,1)=Dst},
{h(1,2), h(2,2), h(3,2), ..., h(N2,2)=Dst},
...
{h(1,m), h(2,m), h(3,m), ....h(Nm,m)=Dst}
}
```

where the following conditions apply: $i \leq N_j \leq N_{max}$ ($j=1..m$)

Note that some $h(i,j)$ may be empty (null) in the case that systems do not reply (not discoverable, or not cooperating).

$h(i-1,j)$ and $h(i,j)$ are the Hops on the same Member Route one hop away from each other.

Hop $h(i,j)$ may be identical with $h(k,l)$ for $i \neq k$ and $j \neq l$; which means there may be portions shared among different Member Routes (parts of Member Routes may overlap).

3.5. Related Round-Trip Delay and Loss Definitions

RTD(i,j,T) is defined as a singleton of the [RFC2681] Round-Trip Delay between the Node with address = Src and the Node at Hop $h(i,j)$ at time T .

RTL(i,j,T) is defined as a singleton of the [RFC6673] Round-trip Loss between the Node with address = Src and the Node at Hop $h(i,j)$ at time T .

3.6. Discussion

Depending on the way that Node Identity is revealed, it may be difficult to determine parallel subpaths between the same pair of Nodes (i.e. multiple parallel links). It is easier to detect parallel subpaths involving different Nodes.

- o If a pair of discovered Nodes identify two different addresses (IP or not), then they will appear to be different Nodes. See item below.
- o If a pair of discovered Nodes identify two different IP addresses, and the IP addresses resolve to the same Node name (in the DNS), then they will appear to be the same Nodes.
- o If a discovered Node always replies using the same network address, regardless of the interface a packet arrives on, then multiple parallel links cannot be detected in that network domain. This condition may apply to traceroute-style methods, but may not apply to other hybrid methods based on In-situ Operations, Administration, and Maintenance (IOAM). For example, if the [RFC5837] ICMP extension mechanism is implemented, then parallel links can be detected with the discovery traceroute-style methods.
- o If parallel links between routers are aggregated below the IP layer, then from Node point of view, all these links share the same pair of IP addresses. The existence of these parallel links can't be detected at the IP layer. This applies to other network domains with layers below them, as well. This condition may apply to traceroute-style methods, but may not apply to other hybrid methods based on IOAM.

When a route assessment employs IP packets (for example), the reality of flow assignment to parallel subpaths involves layers above IP. Thus, the measured Route Ensemble is applicable to IP and higher layers (as described in the methodology's packet of Type-P and flow parameters).

3.7. Reporting the Metric

An Information Model and an XML Data Model for Storing Traceroute Measurements is available in [RFC5388]. The measured information at each hop includes four pieces of information: a one-dimensional hop index, Node symbolic address, Node IP address, and RTD for each response.

The description of Hop information that may be collected according to this memo covers more dimensions, as defined in Section 3.4 above.

For example, the Hop index is two-dimensional to capture the complexity of a Route Ensemble, and it contains corresponding Node identities at a minimum. The models need to be expanded to include these features, as well as Arrival Interface ID, Departure Interface ID, and Arrival Timestamp, when available. The original sending Timestamp from the Src Node anchors a particular measurement in time.

4. Route Assessment Methodologies

There are two classes of methods described in this section, active methods relying on the reaction to TTL or Hop Limit Exceeded condition to discover Nodes on a path, and Hybrid active-passive methods that involve direct interrogation of cooperating Nodes (usually within a single domain). Description of these methods follow.

4.1. Active Methodologies

This section describes the method employed by current open source tools, thereby providing a practical framework for further advanced techniques to be included as method variants. This method is applicable for use across multiple administrative domains.

Internet routing is complex because it depends on the policies of thousands of Autonomous Systems (AS). Most routers perform load balancing on flows using a form of Equal Cost Multiple Path (ECMP). [RFC2991] describes a number of flow-based or hashed approaches (e.g., Modulo-N Hash, Hash-Threshold, Highest Random Weight (HRW)), and makes some good suggestions. Flow-based ECMP avoids increased packet delay variation and possibly overwhelming levels of packet reordering in flows.

A few routers still divide the workload through packet-based techniques, such as a round-robin scheme to distribute every new outgoing packet to multiple links, as explained in [RFC2991]. The methods described in this section assume flow-based ECMP.

Taking into account that Internet protocol was designed under the "end-to-end" principle, the IP payload and its header do not provide any information about the routes or path necessary to reach some destination. For this reason, the popular tool traceroute was developed to gather the IP addresses of each hop along a path using the ICMP protocol [RFC0792]. Traceroute also measures RTD from each hop. However, the growing complexity of the Internet makes it more challenging to develop an accurate traceroute implementation. For instance, the early traceroute tools would be inaccurate in the current network, mainly because they were not designed to retain a flow state. However, evolved traceroute tools, such as Paris-

traceroute [PT] [MLB] and Scamper [SCAMPER], expect to encounter ECMP and achieve more accurate results when they do, where Scamper ensures traceroute packets will follow the same path in 98% of cases[SCAMPER].

Today's traceroute tools send Type-P of packets, either ICMP, UDP, or TCP. UDP and TCP are used when a particular characteristic needs to be verified, such as filtering or traffic shaping on specific ports (i.e., services). UDP and TCP traceroute are also used when ICMP responses are not received. [SCAMPER] supports IPv6 traceroute measurements, keeping the FlowLabel constant in all packets.

Paris-traceroute allows its users to measure the RTD to every Node of the path for a particular flow. Furthermore, either Paris-traceroute or Scamper is capable of unveiling the many available paths between a source and destination (which are visible to active methods). This task is accomplished by repeating complete traceroute measurements with different flow parameters for each measurement; Paris-traceroute provides "exhaustive" mode while scamper provides "tracelb" (stands for traceroute load balance). The Framework for IP Performance Metrics (IPPM) ([RFC2330] updated by[RFC7312]) has the flexibility to require that the Round-Trip Delay measurement [RFC2681] uses packets with the constraints to assure that all packets in a single measurement appear as the same flow. This flexibility covers ICMP, UDP, and TCP. The accompanying methodology of [RFC2681] needs to be expanded to report the sequential hop identifiers along with RTD measurements, but no new metric definition is needed.

The advanced route assessment methods used in Paris-traceroute [PT] keep the critical fields constant for every packet to maintain the appearance of the same flow. When considering IPv6 headers, it is necessary to ensure that the IP source and destination addresses and the FlowLabel are constant (but note that many routers ignore the FlowLabel field at this time), see [RFC6437]. Use of IPv6 Extension Headers may add critical fields, and SHOULD be avoided. In IPv4, certain fields of the IP header and the first four bytes of the IP payload should remain constant in a flow. In the IPv4 header, the IP source and destination addresses, protocol number, and Diffserv fields identify flows. The first four payload bytes include the UDP and TCP ports, and the ICMP type, code, and checksum fields.

Maintaining a constant ICMP checksum in IPv4 is most challenging, as the ICMP sequence number or identifier fields will usually change for different probes of the same path. Probes should use arbitrary bytes in the ICMP data field to offset changes to sequence number and identifier, thus keeping the checksum constant.

Finally, it is also essential to route the resulting ICMP Time Exceeded messages along a consistent path. In IPv6, the fields above are sufficient. In IPv4, the ICMP Time Exceeded message will contain the IP header and the first eight bytes of the IP payload, which affects its ICMP checksum. The TCP sequence number, UDP Length, and UDP checksum will affect this value, and should remain constant.

Formally, to maintain the same flow in the measurements to a particular hop, the Type-P-Route-Ensemble-Method-Variant packets should be[PT]:

- o TCP case: For IPv4, the fields Src, Dst, port-Src, port-Dst, sequence number, and Diffserv Field SHOULD be the same. For IPv6, the field FlowLabel, Src and Dst SHOULD be the same.
- o UDP case: For IPv4, the fields Src, Dst, port-Src, port-Dst, Diffserv should be the same, and the UDP-checksum SHOULD change to keep the IP checksum of the ICMP time exceeded reply constant. Then, the data length should be fixed, and the data field is used to make it so (consider that ICMP checksum uses its data field, which contains the original IP header plus 8 bytes of UDP, where TTL, IP identification, IP checksum, and UDP checksum changes). For IPv6, the field FlowLabel, and Source and Destination addresses SHOULD be the same.
- o ICMP case: For IPv4, the Data field SHOULD compensate variations on TTL or Hop Limit, IP identification, and IP checksum for every packet. There is no need to consider ICMPv6 because only FlowLabel of IPv6 and Source and Destination addresses are used, and all of them SHOULD be constant.

Then, the way to identify different hops and attempts of the same IPv4 flow is:

- o TCP case: The IP identification field.
- o UDP case: The IP identification field.
- o ICMP case: The IP identification field, and ICMP Sequence number.

4.1.1. Temporal Composition for Route Metrics

The Active Route Assessment Methods described above have the ability to discover portions of a path where ECMP load balancing is present, observed as two or more unique Member Routes having one or more distinct Hops which are part of the Route Ensemble. Likewise, attempts to deliberately vary the flow characteristics to discover

all Member Routes will reveal portions of the path which are flow-invariant.

Section 9.2 of [RFC2330] describes Temporal Composition of metrics, and introduces the possibility of a relationship between earlier measurement results and the results for measurement at the current time (for a given metric). There is value in establishing a Temporal Composition relationship for Route Metrics. However, this relationship does not represent a forecast of future route conditions in any way.

For Route Metric measurements, the value of Temporal Composition is to reduce the measurement iterations required with repeated measurements. Reduced iterations are possible by inferring that current measurements using fixed and previously measured flow characteristics:

- o will have many common hops with previous measurements.
- o will have relatively time-stable results at the ingress and egress portions of the path when measured from user locations, as opposed to measurements of backbone networks and across inter-domain gateways.
- o may have greater potential for time-variation in path portions where ECMP load balancing is observed (because increasing or decreasing the pool of links changes the hash calculations).

Optionally, measurement systems may take advantage of the inferences above when seeking to reduce measurement iterations, after exhaustive measurements indicate that the time-stable properties are present. Repetitive Active Route measurement systems:

1. SHOULD occasionally check path portions which have exhibited stable results over time, particularly ingress and egress portions of the path (e.g., daily checks if measuring many times during a day).
2. SHOULD continue testing portions of the path that have previously exhibited ECMP load balancing.
3. SHALL trigger re-assessment of the complete path and Route Ensemble, if any change in hops is observed for a specific (and previously tested) flow.

4.1.2. Routing Class Identification

There is an opportunity to apply the [RFC2330] notion of equal treatment for a class of packets, "...very useful to know if a given Internet component treats equally a class C of different types of packets", as it applies to Route measurements. The notion of class C was examined further in [RFC8468] as it applied to load-balancing flows over parallel paths, which is the case we develop here. Knowledge of class C parameters (unrelated to address classes of the past) on a path potentially reduces the number of flows required for a given method to assess a Route Ensemble over time.

First, recognize that each Member Route of a Route Ensemble will have a corresponding class C. Class C can be discovered by testing with multiple flows, all of which traverse the unique set of hops that comprise a specific Member Route.

Second, recognize that the different classes depend primarily on the hash functions used at each instance of ECMP load balancing on the path.

Third, recognize the synergy with Temporal Composition methods (described above), where evaluation intends to discover time-stable portions of each Member Route, so that more emphasis can be placed on ECMP portions that also determine class C.

The methods to assess the various class C characteristics benefit from the following measurement capabilities:

- o flows designed to determine which n-tuple header fields are considered by a given hash function and ECMP hop on the path, and which are not. This operation immediately narrows the search space, where possible, and partially defines a class C.
- o a priori knowledge of the possible types of hash functions in use also helps to design the flows for testing (major router vendors publish information about these hash functions, examples are here [LOAD_BALANCE]).
- o ability to direct the emphasis of current measurements on ECMP portions of the path, based on recent past measurement results (the Routing Class of some portions of the path is essentially "all packets").

4.1.3. Intermediate Observation Point Route Measurement

There are many examples where passive monitoring of a flow at an Observation Point within the network can detect unexpected Round Trip Delay or Delay Variation. But how can the cause of the anomalous delay be investigated further *from the Observation Point* possibly located at an intermediate point on the path?

In this case, knowledge that the flow of interest belongs to a specific Routing Class C will enable measurement of the route where anomalous delay has been observed. Specifically, Round-Trip Delay assessment to each Hop on the path between the Observation Point and the Destination for the flow of interest may discover high or variable delay on a specific link and Hop combination.

The determination of a Routing Class C which includes the flow of interest is as described in the section above, aided by computation of the relevant hash function output as the target.

4.2. Hybrid Methodologies

The Hybrid Type I methods provide an alternative method for Route Member assessment. As mentioned in the Scope section, [I-D.ietf-ippm-ioam-data] provides a possible set of data fields that would support route identification.

In general, nodes in the measured domain would be equipped with specific abilities:

- o Store the identity of nodes that a packet has visited in header data fields, in the order the packet visited the nodes.
- o Support of a "Loopback" capability, where a copy of the packet is returned to the encapsulating node, and the packet is processed like any other IOAM packet on the return transfer.

In addition to node identity, nodes may also identify the ingress and egress interfaces utilized by the tracing packet, the absolute time when the packet was processed, and other generic data (as described in section 4 of [I-D.ietf-ippm-ioam-data]). Interface identification isn't necessarily limited to IP, i.e. different links in a bundle (LACP) could be identified. Equally well, links without explicit IP addresses can be identified (like with unnumbered interfaces in an IGP deployment).

Note that the Type-P packet specification for this method will likely be a partial specification, because most of the packet fields are determined by the user traffic. The packet (encapsulation) header(s)

added by the Hybrid method can certainly be specified in Type-P, in unpopulated form.

4.3. Combining Different Methods

In principle, there are advantages if the entity conducting Route measurements can utilize both forms of advanced methods (active and hybrid), and combine the results. For example, if there are Nodes involved in the path that qualify as Cooperating Nodes, but not as Discoverable Nodes, then a more complete view of Hops on the path is possible when a hybrid method (or interrogation protocol) is applied and the results are combined with the active method results collected across all other domains.

In order to combine the results of active and hybrid/interrogation methods, the network Nodes that are part of a domain supporting an interrogation protocol have the following attributes:

1. Nodes at the ingress to the domain SHOULD be both Discoverable and Cooperating.
2. Any Nodes within the domain that are both Discoverable and Cooperating SHOULD reveal the same Node Identity in response to both active and hybrid methods.
3. Nodes at the egress to the domain SHOULD be both Discoverable and Cooperating, and SHOULD reveal the same Node Identity in response to both active and hybrid methods.

When Nodes follow these requirements, it becomes a simple matter to match single domain measurements with the overlapping results from a multidomain measurement.

In practice, Internet users do not typically have the ability to utilize the OAM capabilities of networks that their packets traverse, so the results from a remote domain supporting an interrogation protocol would not normally be accessible. However, a network operator could combine interrogation results from their access domain with other measurements revealing the path outside their domain.

5. Background on Round-Trip Delay Measurement Goals

The aim of this method is to use packet probes to unveil the paths between any two end-Nodes of the network. Moreover, information derived from RTD measurements might be meaningful to identify:

1. Intercontinental submarine links

2. Satellite communications
3. Congestion
4. Inter-domain paths

This categorization is widely accepted in the literature and among operators alike, and it can be trusted with empirical data and several sources as ground of truth (e.g., [RTTSub]) but it is an inference measurement nonetheless [bdrmap][IDCong].

The first two categories correspond to the physical distance dependency on Round-Trip Delay (RTD), the next one binds RTD with queuing delay on routers, and the last one helps to identify different ASes using traceroutes. Due to the significant contribution of propagation delay in long-distance hops, RTD will be on the order of 100ms on transatlantic hops, depending on the geolocation of the vantage points. Moreover, RTD is typically higher than 480ms when two hops are connected using geostationary satellite technology (i.e., their orbit is at 36000km). Detecting congestion with latency implies deeper mathematical understanding since network traffic load is not stationary. Nonetheless, as the first approach, a link seems to be congested if observing different/varying statistical results after sending several traceroute probes (e.g., see [IDCong]). Finally, to recognize distinctive ASes in the same traceroute path is challenging, because more data is needed, like AS relationships and RIR delegations among other (for more detail, please consult [bdrmap]).

6. RTD Measurements Statistics

Several articles have shown that network traffic presents a self-similar nature [SSNT] [MLRM] which is accountable for filling the queues of the routers. Moreover, router queues are designed to handle traffic bursts, which is one of the most remarkable features of self-similarity. Naturally, while queue length increases, the delay to traverse the queue increases as well and leads to an increase on RTD. Due to traffic bursts generating short-term overflow on buffers (spiky patterns), every RTD only depicts the queueing status on the instant when that packet probe was in transit. For this reason, several RTD measurements during a time window could begin to describe the random behavior of latency. Loss must also be accounted for in the methodology.

To understand the ongoing process, examining the quartiles provides a non-parametric way of analysis. Quartiles are defined by five values: minimum RTD (m), RTD value of the 25% of the Empirical Cumulative Distribution Function (ECDF) (Q1), the median value (Q2),

the RTD value of the 75% of the ECDF (Q3) and the maximum RTD (M). Congestion can be inferred when RTD measurements are spread apart, and consequently, the Inter-Quartile Range (IQR), the distance between Q3 and Q1, increases its value.

This procedure requires the algorithm presented in [P2] to compute quartile values "on the fly".

This procedure allows us to update the quartiles value whenever a new measurement arrives, which is radically different from classic methods of computing quartiles because they need to use the whole dataset to compute the values. This way of calculus provides savings in memory and computing time.

To sum up, the proposed measurement procedure consists of performing traceroutes several times to obtain samples of the RTD in every hop from a path, during a time window (W), and compute the quartiles for every hop. This procedure could be done for a single Member Route flow, a non-exhaustive search with parameter E (defined below) set as False, or for every detected Route Ensemble flow (E=True).

The identification of a specific Hop in traceroute is based on the IP origin address of the returned ICMP Time Exceeded packet, and on the distance identified by the value set in the TTL (or Hop Limit) field inserted by traceroute. As this specific Hop can be reached by different paths, also the IP source and destination addresses of the traceroute packet need to be recorded. Finally, different return paths are distinguished by evaluating the ICMP Time Exceeded TTL (or Hop Limit) of the reply message: if this TTL (or Hop Limit) is constant for different paths containing the same Hop, the return paths have the same distance. Moreover, this distance can be estimated considering that the TTL (or Hop Limit) value is normally initialized with values 64, 128, or 255. The 5-tuple (origin IP, destination IP, reply IP, distance, response TTL or Hop Limit) unequivocally identifies every measurement.

This algorithm below runs in the origin of the traceroute. It returns the Qs quartiles for every Hop and Alt (alternative paths because of balancing). Notice that the "Alt" parameter condenses the parameters of the 5-tuple (origin IP, destination IP, reply IP, distance, response TTL), i.e., one for each possible combination.

```

=====
0  input:  W (window time of the measurement)
1          i_t (time between two measurements, set the i_t time
2              long enough to avoid incomplete results)
3          E (True: exhaustive, False: a single path)
4          Dst (destination IP address)
5  output: Qs (quartiles for every Hop and Alt)
=====
6  T := start_timer(W)
7  while T is not finished do:
8      start_timer(i_t)
9      RTD(Hop,Alt) = advanced-traceroute(Dst,E)
10     for each Hop and Alt in RTD do:
11         |   Qs[Dst,Hop,Alt] := ComputeQs(RTD(Hop,Alt))
12     done
13     wait until i_t timer is expired
14 done
15 return (Qs)
=====

```

During the time *W*, lines 6 and 7 assure that the measurement loop is made. Line 8 and 13 set a timer for each cycle of measurements. A cycle comprises the traceroutes packets, considering every possible Hop and the alternatives paths in the Alt variable (ensured in lines 9-12). In line 9, the advance-traceroute could be either Paris-traceroute or Scamper, which will use the "exhaustive" mode or "tracelb" option if *E* is set True, respectively. The procedure returns a list of tuples (*m*,*Q1*,*Q2*,*Q3*,*M*) for each intermediate hop, or "Alt" in as a function of the 5-tuple, in the path towards the Dst. Finally, lines 10 through 12 stores each measurement into the real-time quartiles computation.

Notice there are cases where the even having a unique hop at distance *h* from the Src to Dst, the returning path could have several possibilities, yielding in different total paths. In this situation, the algorithm will return more "Alt" for this particular hop.

7. Security Considerations

The security considerations that apply to any active measurement of live paths are relevant here as well. See [RFC4656] and [RFC5357].

The active measurement process of "changing several fields to keep the checksum of different packets identical" does not require special security considerations because it is part of synthetic traffic generation, and is designed to have minimal to zero impact on network processing (to process the packets for ECMP).

Some of the protocols used (e.g., ICMP) do not provide cryptographic protection for the requested/returned data, and there are risks of processing untrusted data in general, but these are limitations of the existing protocols where we are applying new methods.

For applicable Hybrid methods, the security considerations in[I-D.ietf-ippm-ioam-data] apply.

When considering privacy of those involved in measurement or those whose traffic is measured, the sensitive information available to potential observers is greatly reduced when using active techniques which are within this scope of work. Passive observations of user traffic for measurement purposes raise many privacy issues. We refer the reader to the privacy considerations described in the Large Scale Measurement of Broadband Performance (LMAP) Framework [RFC7594], which covers active and passive techniques.

8. IANA Considerations

This memo makes no requests of IANA. We thank the good folks at IANA for having checked this section anyway.

9. Acknowledgements

The original 3 authors (Ignacio, Al, Joachim) acknowledge Ruediger Geib, for his penetrating comments on the initial draft, and his initial text for the Appendix on MPLS. Carlos Pignataro challenged the authors to consider a wider scope, and applied his substantial expertise with many technologies and their measurement features in his extensive comments. Frank Brockners also shared useful comments, so did Footer Foote. We thank them all!

10. Appendix I MPLS Methods for Route Assessment

A Node assessing an MPLS path must be part of the MPLS domain where the path is implemented. When this condition is met, RFC 8029 provides a powerful set of mechanisms to detect "correct operation of the data plane, as well as a mechanism to verify the data plane against the control plane" [RFC8029].

MPLS routing is based on the presence of a Forwarding Equivalence Class (FEC) Stack in all visited Nodes. Selecting one of several Equal Cost Multi Path (ECMP) is however based on information hidden deeper in the stack. Late deployments may support a so called "Entropy label" for this purpose. State of the art deployments base their choice of an ECMP member interface on the complete MPLS label stack and on IP addresses up to the complete 5 tuple IP header information (see Section 2.4 of [RFC7325]). Load Sharing based on IP

information decouples this function from the actual MPLS routing information. Thus, an MPLS traceroute is able to check how packets with a contiguous number of ECMP relevant IP addresses (and an identical MPLS label stack) are forwarded by a particular router. The minimum number of equivalent MPLS paths traceable at a router should be 32. Implementations supporting more paths are available.

The MPLS echo request and reply messages offering this feature must support the Downstream Detailed Mapping TLV (was Downstream Mapping initially, but the latter has been deprecated). The MPLS echo response includes the incoming interface where a router received the MPLS Echo request. The MPLS Echo reply further informs which of the *n* addresses relevant for the load sharing decision results in a particular next hop interface and contains the next hop's interface address (if available). This ensures that the next hop will receive a properly coded MPLS Echo request in the next step route of assessment.

[RFC8403] explains how a central Path Monitoring System could be used to detect arbitrary MPLS paths between any routers within a single MPLS domain. The combination of MPLS forwarding, Segment Routing and MPLS traceroute offers a simple architecture and a powerful mechanism to detect and validate (segment routed) MPLS paths.

11. References

11.1. Normative References

- [I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-10 (work in progress), July 2020.
- [RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <<https://www.rfc-editor.org/info/rfc792>>.
- [RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <<https://www.rfc-editor.org/info/rfc1122>>.
- [RFC1812] Baker, F., Ed., "Requirements for IP Version 4 Routers", RFC 1812, DOI 10.17487/RFC1812, June 1995, <<https://www.rfc-editor.org/info/rfc1812>>.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, DOI 10.17487/RFC2330, May 1998, <<https://www.rfc-editor.org/info/rfc2330>>.
- [RFC2681] Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, DOI 10.17487/RFC2681, September 1999, <<https://www.rfc-editor.org/info/rfc2681>>.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, DOI 10.17487/RFC4656, September 2006, <<https://www.rfc-editor.org/info/rfc4656>>.
- [RFC5388] Niccolini, S., Tartarelli, S., Quittek, J., Dietz, T., and M. Swamy, "Information Model and XML Data Model for Traceroute Measurements", RFC 5388, DOI 10.17487/RFC5388, December 2008, <<https://www.rfc-editor.org/info/rfc5388>>.
- [RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, DOI 10.17487/RFC6438, November 2011, <<https://www.rfc-editor.org/info/rfc6438>>.
- [RFC6673] Morton, A., "Round-Trip Packet Loss Metrics", RFC 6673, DOI 10.17487/RFC6673, August 2012, <<https://www.rfc-editor.org/info/rfc6673>>.
- [RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <<https://www.rfc-editor.org/info/rfc7799>>.
- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <<https://www.rfc-editor.org/info/rfc8029>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

- [RFC8468] Morton, A., Fabini, J., Elkins, N., Ackermann, M., and V. Hegde, "IPv4, IPv6, and IPv4-IPv6 Coexistence: Updates for the IP Performance Metrics (IPPM) Framework", RFC 8468, DOI 10.17487/RFC8468, November 2018, <<https://www.rfc-editor.org/info/rfc8468>>.

11.2. Informative References

- [bdrmap] Luckie, M., Dhamdhere, A., Huffaker, B., Clark, D., and KC. Claffy, "bdrmap: Inference of Borders Between IP Networks", In Proceedings of the 2016 ACM on Internet Measurement Conference, pp. 381-396. ACM, 2016.
- [IDCong] Luckie, M., Dhamdhere, A., Clark, D., and B. Huffaker, "Challenges in inferring Internet interdomain congestion", In Proceedings of the 2014 Conference on Internet Measurement Conference, pp. 15-22. ACM, 2014.
- [LOAD_BALANCE] Sanguanpong, S., Pittayapitak, W., and K. Kasom Koht-Arsa, "COMPARISON OF HASH STRATEGIES FOR FLOW-BASED LOAD BALANCING", International Journal of Electronic Commerce Studies, Vol.6, No.2, pp.259-268. <http://dx.doi.org/10.7903/ijecs.1346>, 2015.
- [MLB] Augustin, B., Friedman, T., and R. Teixeira, "Measuring load-balanced paths in the Internet", Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, pp. 149-160. ACM, 2007., 2007.
- [MLRM] Fontugne, R., Mazel, J., and K. Fukuda, "An empirical mixture model for large-scale RTT measurements", 2015 IEEE Conference on Computer Communications (INFOCOM), pp. 2470-2478. IEEE, 2015., 2015.
- [P2] Jain, R. and I. Chlamtac, "The P 2 algorithm for dynamic calculation of quartiles and histograms without storing observations", Communications of the ACM 28.10 (1985): 1076-1085, 2015.
- [PT] Augustin, B., Cuvellier, X., Orgogozo, B., Viger, F., Friedman, T., Latapy, M., Magnien, C., and R. Teixeira, "Avoiding traceroute anomalies with Paris traceroute", Proceedings of the 6th ACM SIGCOMM conference on Internet measurement, pp. 153-158. ACM, 2006., 2006.

- [RFC2991] Thaler, D. and C. Hopps, "Multipath Issues in Unicast and Multicast Next-Hop Selection", RFC 2991, DOI 10.17487/RFC2991, November 2000, <<https://www.rfc-editor.org/info/rfc2991>>.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, DOI 10.17487/RFC5357, October 2008, <<https://www.rfc-editor.org/info/rfc5357>>.
- [RFC5835] Morton, A., Ed. and S. Van den Berghe, Ed., "Framework for Metric Composition", RFC 5835, DOI 10.17487/RFC5835, April 2010, <<https://www.rfc-editor.org/info/rfc5835>>.
- [RFC5837] Atlas, A., Ed., Bonica, R., Ed., Pignataro, C., Ed., Shen, N., and JR. Rivers, "Extending ICMP for Interface and Next-Hop Identification", RFC 5837, DOI 10.17487/RFC5837, April 2010, <<https://www.rfc-editor.org/info/rfc5837>>.
- [RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <<https://www.rfc-editor.org/info/rfc6437>>.
- [RFC7312] Fabini, J. and A. Morton, "Advanced Stream and Sampling Framework for IP Performance Metrics (IPPM)", RFC 7312, DOI 10.17487/RFC7312, August 2014, <<https://www.rfc-editor.org/info/rfc7312>>.
- [RFC7325] Villamizar, C., Ed., Kompella, K., Amante, S., Malis, A., and C. Pignataro, "MPLS Forwarding Compliance and Performance Requirements", RFC 7325, DOI 10.17487/RFC7325, August 2014, <<https://www.rfc-editor.org/info/rfc7325>>.
- [RFC7594] Eardley, P., Morton, A., Bagnulo, M., Burbridge, T., Aitken, P., and A. Akhter, "A Framework for Large-Scale Measurement of Broadband Performance (LMAP)", RFC 7594, DOI 10.17487/RFC7594, September 2015, <<https://www.rfc-editor.org/info/rfc7594>>.
- [RFC8403] Geib, R., Ed., Filsfils, C., Pignataro, C., Ed., and N. Kumar, "A Scalable and Topology-Aware MPLS Data-Plane Monitoring System", RFC 8403, DOI 10.17487/RFC8403, July 2018, <<https://www.rfc-editor.org/info/rfc8403>>.

- [RTTSub] Bischof, Z., Rula, J., and F. Bustamante, "In and out of Cuba: Characterizing Cuba's connectivity", In Proceedings of the 2015 ACM Conference on Internet Measurement Conference, pp. 487-493. ACM, 2015.
- [SCAMPER] Matthew Luckie, M., "Scamper: a scalable and extensible packet prober for active measurement of the Internet", Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, pp. 239-245. ACM, 2010., 2010.
- [SSNT] Park, K. and W. Willinger, "Self-Similar Network Traffic and Performance Evaluation (1st ed.)", John Wiley & Sons, Inc., New York, NY, USA, 2000.

Authors' Addresses

J. Ignacio Alvarez-Hamelin
Universidad de Buenos Aires
Av. Paseo Colon 850
Buenos Aires C1063ACV
Argentina

Phone: +54 11 5285-0716
Email: ihameli@cnet.fi.uba.ar
URI: <http://cnet.fi.uba.ar/ignacio.alvarez-hamelin/>

Al Morton
AT&T Labs
200 Laurel Avenue South
Middletown, NJ 07748
USA

Phone: +1 732 420 1571
Fax: +1 732 368 1192
Email: acm@research.att.com

Joachim Fabini
TU Wien
Gusshausstrasse 25/E389
Vienna 1040
Austria

Phone: +43 1 58801 38813
Fax: +43 1 58801 38898
Email: Joachim.Fabini@tuwien.ac.at
URI: <http://www.tc.tuwien.ac.at/about-us/staff/joachim-fabini/>

Carlos Pignataro
Cisco Systems, Inc.
7200-11 Kit Creek Road
Research Triangle Park, NC 27709
USA

Email: cpignata@cisco.com

Ruediger Geib
Deutsche Telekom
Heinrich Hertz Str. 3-7
Darmstadt 64295
Germany

Phone: +49 6151 5812747
Email: Ruediger.Geib@telekom.de

ippm
Internet-Draft
Intended status: Informational
Expires: December 27, 2018

H. Song, Ed.
Z. Li
T. Zhou
Z. Wang
Huawei
June 25, 2018

In-situ OAM Processing in Tunnels
draft-song-ippm-ioam-tunnel-mode-00

Abstract

This document describes the In-situ OAM (iOAM) processing behavior in a network with tunnels. Specifically, the iOAM processing in tunnels with the uniform model and the pipe model is discussed. The procedure is applicable to different type of tunnel protocols.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 27, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Motivation	2
2. Uniform Model	3
2.1. U1: IOAM Domain Starts and Ends outside of a Tunnel	3
2.2. U2: IOAM Domain Starts and Ends within a Tunnel	4
2.3. U3: IOAM Domain Starts and Ends at any Nodes	4
2.4. Discussion	5
3. Pipe Model	5
3.1. P1: IOAM Domain Starts and Ends outside of a Tunnel	5
3.2. P2: IOAM Domain Starts and Ends within a Tunnel	5
3.3. Discussion	5
4. Examples	6
5. Security Considerations	6
6. IANA Considerations	6
7. Contributors	6
8. Acknowledgments	6
9. References	6
9.1. Normative References	6
9.2. Informative References	7
Authors' Addresses	7

1. Motivation

In-situ OAM (ioAM) records OAM data associated with user packets while these packets traverse a network [I-D.brockners-inband-oam-requirements]. The ioAM instruction and data are kept in an ioAM header which is defined in [I-D.ietf-ippm-ioam-data]. The ioAM header needs to be encapsulated in a packet's transport protocol header in order to be processed by the network nodes who are capable of ioAM processing. So far, the ioAM header encapsulation methods have been defined for several protocols, including IPv6, VXLAN-GPE, NSH, SRv6 [I-D.brockners-inband-oam-transport], [I-D.ietf-sfc-ioam-nsh], GENEVE [I-D.brockners-ippm-ioam-geneve], GRE [I-D.weis-ippm-ioam-gre], and some others.

While the original scope of ioAM is purposely confined to a single network domain for simplicity, the authentic E2E data collection capability of ioAM is invaluable to network operators. In reality,

especially in carrier networks, a user packet may traverse several network domains and pass through various tunnels for QoS, traffic engineering, or public network traversal. To extend the scope of iOAM's applicability and fully realize iOAM's potential, we need to consider various network conditions. In this document, we describe how iOAM should be processed in a network with tunnels.

A tunneling protocol usually needs to add another layer of protocol header (i.e., the tunnel header) over the original packet. Within a tunnel, only the outermost tunnel header is supposed to be processed by a network node. Therefore, depending on the locations where the iOAM header is encapsulated/decapsulated and the tunnel operation mode, the iOAM processing is also different.

In general, there are two modes of tunnel operations: the Uniform Model and the Pipe Model. The Uniform Model treats the nodes in a tunnel uniformly as the nodes outside of the tunnel on an E2E path. On the contrary, the Pipe Model abstracts all the nodes between the tunnel ingress and egress as a circuit so no nodes in the tunnel is visible to the nodes outside of the tunnel. The iOAM processing behavior is discussed for each mode as follows.

2. Uniform Model

2.1. U1: IOAM Domain Starts and Ends outside of a Tunnel

In this case, a tunnel is fully in between the head node and the end node of an iOAM path. This includes the situation that the tunnel ingress coincides with the iOAM head node and/or the tunnel egress coincides with the iOAM end node. The iOAM header handling for different situation is described as follows:

- o iOAM head node is outside of the tunnel: The iOAM header is encapsulated into the original packet and processed.
- o iOAM head node is the tunnel ingress: The iOAM header is encapsulated into the original packet and processed. The iOAM header is copied from the original packet and encapsulated into the underlay protocol header.
- o iOAM end node is outside of the tunnel: The iOAM header is decapsulated from the original packet after iOAM processing.
- o iOAM end node is the tunnel egress: The iOAM header in the underlay protocol header is processed as usual. After the tunnel header is removed and the original packet is exposed, the iOAM header is copied to overwrite the original packet's iOAM header.

After the iOAM processing is finished, the iOAM header is removed from the original packet.

- o Other nodes in the iOAM domain: If the node is outside or inside of the tunnel, the iOAM header encapsulated in the outermost protocol header is processed. If the node is the tunnel ingress, the iOAM header in the original packet needs to be copied and encapsulated into the underlay protocol header. If the node is the tunnel egress, the iOAM header in the underlay protocol header needs to be copied to overwrite the iOAM header in the original packet.

2.2. U2: IOAM Domain Starts and Ends within a Tunnel

There is nothing special about this case since the transport network will not be aware of the tunnel. In this case, the iOAM is processed as usual.

2.3. U3: IOAM Domain Starts and Ends at any Nodes

For extra flexibility, the iOAM domain can be configured to start and end at any node (e.g., in or out of a tunnel). The iOAM header handling for different situation is described as follows:

- o iOAM head node is outside of the tunnel: The iOAM header is encapsulated in the original packet.
- o iOAM head node is the tunnel ingress: The iOAM header is encapsulated in the original packet first and processed. Then the iOAM header is copied from the original packet and encapsulated into the underlay protocol header. Meanwhile, the iOAM header in the original packet must be removed.
- o iOAM head node is in the tunnel: The iOAM header is encapsulated in the underlay protocol header and processed.
- o iOAM head node is the tunnel egress: The iOAM header is encapsulated in the underlay protocol header first and processed. When the tunnel header is removed, the iOAM header is copied from the underlay protocol header and encapsulated into the original packet.
- o iOAM end node is outside of the tunnel: The iOAM header is decapsulated from the original packet.
- o iOAM end node is the tunnel ingress: The iOAM header is decapsulated from the original packet.

- o iOAM end node is in the tunnel: The iOAM header is decapsulated from the underlay protocol header.
- o iOAM end node is the tunnel egress: The iOAM header is removed with the underlay protocol header.
- o Tunnel ingress is in the IOAM domain: The iOAM header is decapsulated from the original packet and encapsulated in the underlay protocol header.
- o Tunnel egress is in the iOAM domain: The iOAM header in the underlay protocol header is encapsulated into the original packet.

2.4. Discussion

U1 achieves the best implementation efficiency since it eliminates one encapsulation or decapsulation operation while U3 achieves the best flexibility and reduces the packet overhead.

Since a tunnel usually aggregates multiple flows, so U2 (or U3 when the iOAM head node is in a tunnel) can only conduct iOAM at the tunnel granularity and on aggregated flows.

3. Pipe Model

3.1. P1: IOAM Domain Starts and Ends outside of a Tunnel

This case includes the situation that the tunnel ingress coincides with the iOAM head node and/or the tunnel egress coincides with the iOAM end node.

In this mode, the iOAM header only exists in the original packet. It is not copied to the tunnel header. Within the tunnel, the iOAM header is invisible to the underlay network so it is not processed. At the tunnel ingress, the iOAM header is processed before the tunnel header is applied. At the tunnel egress, the iOAM header is processed after the tunnel header is removed. To the iOAM header, the entire tunnel appears to be just one hop.

3.2. P2: IOAM Domain Starts and Ends within a Tunnel

This mode is identical to U2.

3.3. Discussion

In P1, the hop-by-hop iOAM data is missing for the tunnel. However, this mode also provides a convenient way to pass through third party

tunnels in which either the iOAM is not supported or the tunnel operators do not participate in the iOAM service.

On the other hand, the tunnel operators can support iOAM independently to monitor the tunnel performance using the mode of P2. In this case, U1 can also be applied without any confliction, so both underlay and overlay can be monitored by different entities.

When iOAM works in the E2E operation mode as described in [I-D.ietf-ippm-ioam-data], any tunnel on the path should be configured to the Pipe model in order to avoid the unnecessary iOAM header encapsulation/decapsulation.

4. Examples

Examples will be added in future revisions.

5. Security Considerations

TBD

6. IANA Considerations

N/A

7. Contributors

TBD.

8. Acknowledgments

TBD.

9. References

9.1. Normative References

[I-D.brockners-inband-oam-transport]
Brockners, F., Bhandari, S., Govindan, V., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., and R. Chang, "Encapsulations for In-situ OAM Data", draft-brockners-inband-oam-transport-05 (work in progress), July 2017.

[I-D.brockners-ippm-ioam-geneve]

Brockners, F., Bhandari, S., Govindan, V., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., and R. Chang, "Geneve encapsulation for In-situ OAM Data", draft-brockners-ippm-ioam-geneve-01 (work in progress), June 2018.

[I-D.ietf-ippm-ioam-data]

Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., Chang, R., daniel.bernier@bell.ca, d., and J. Lemon, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-02 (work in progress), March 2018.

[I-D.ietf-sfc-ioam-nsh]

Brockners, F., Bhandari, S., Govindan, V., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., and R. Chang, "NSH Encapsulation for In-situ OAM Data", draft-ietf-sfc-ioam-nsh-00 (work in progress), May 2018.

[I-D.weis-ippm-ioam-gre]

Weis, B., Brockners, F., crhill@cisco.com, c., Bhandari, S., Govindan, V., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A., Gafni, B., Lapukhov, P., and M. Spiegel, "GRE Encapsulation for In-situ OAM Data", draft-weis-ippm-ioam-gre-00 (work in progress), March 2018.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

9.2. Informative References

[I-D.brockners-inband-oam-requirements]

Brockners, F., Bhandari, S., Dara, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mozes, D., Mizrahi, T., <>, P., and r. remy@barefootnetworks.com, "Requirements for In-situ OAM", draft-brockners-inband-oam-requirements-03 (work in progress), March 2017.

Authors' Addresses

Haoyu Song (editor)
Huawei
2330 Central Expressway
Santa Clara
USA

Email: haoyu.song@huawei.com

Zhenbin Li
Huawei
156 Beiqing Road
Beijing, 100095
P.R. China

Email: lizhenbin@huawei.com

Tianran Zhou
Huawei
156 Beiqing Road
Beijing, 100095
P.R. China

Email: zhoutianran@huawei.com

Zhongzhen Wang
Huawei
156 Beiqing Road
Beijing, 100095
P.R. China

Email: wangzhongzhen@huawei.com

ippm
Internet-Draft
Intended status: Informational
Expires: August 25, 2022

M. Spiegel
Barefoot Networks, an Intel company
F. Brockners
Cisco
S. Bhandari
Thoughtspot
R. Sivakolundu
Cisco
February 21, 2022

In-situ OAM raw data export with IPFIX
draft-spiegel-ippm-ioam-rawexport-06

Abstract

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information in the packet while the packet traverses a path between two points in the network. This document discusses how In-situ Operations, Administration, and Maintenance (IOAM) information can be exported in raw, i.e. uninterpreted, format from network devices to systems, such as monitoring or analytics systems using IPFIX.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 25, 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

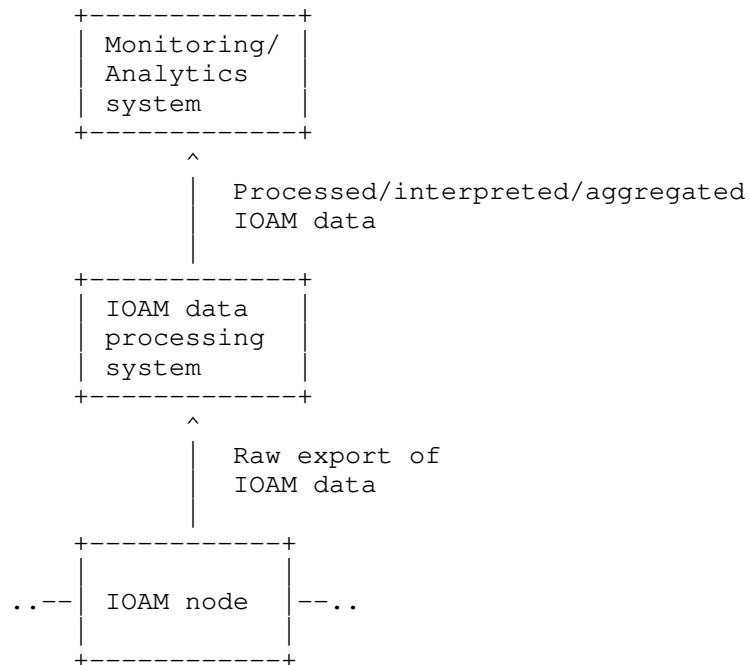
Table of Contents

1. Introduction	3
1.1. Requirements	5
1.2. Scope	5
2. Conventions	6
3. IPFIX for IOAM raw data export	6
3.1. Key IPFIX information elements leveraged for IOAM raw data export	6
3.2. New IPFIX information elements leveraged for IOAM raw data export	7
3.2.1. ioamReportFlags	7
3.2.2. ioamEncapsulationType	8
3.2.3. ioamPreallocatedTraceData	9
3.2.4. ioamIncrementalTraceData	9
3.2.5. ioamE2EData	10
3.2.6. ioamPOTData	10
3.2.7. ioamDirectExportData	11
3.2.8. ipHeaderPacketSectionWithPadding	11
3.2.9. ethernetFrameSection	12
4. Examples	13
4.1. Fixed Length IP Packet	13
4.2. Variable Length IP Packet (length < 255)	14
4.3. Variable Length IP Packet (length > 255)	15
4.4. Variable Length ETHERNET Packet (length < 255)	16
4.5. Variable Length IP Packet with Fixed Length IOAM Incremental Trace Data	17
4.6. Variable Length IP Packet with Variable Length IOAM Incremental Trace Data	18
5. IANA Considerations	19
6. Manageability Considerations	20
7. Security Considerations	20
8. Acknowledgements	20
9. References	20
9.1. Normative References	20
9.2. Informative References	21
Authors' Addresses	22

1. Introduction

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information in the packet while the packet traverses a path between two points in the network. IOAM data fields are defined in [I-D.ietf-ippm-ioam-data]. This document discusses how In-situ Operations, Administration, and Maintenance (IOAM) information can be exported in raw format, i.e. uninterpreted format, from network devices to systems, such as monitoring or analytics systems using IPFIX [RFC7011].

"Raw export of IOAM data" refers to a mode of operation where a node exports the IOAM data as it is received in the packet. The exporting node neither interprets, aggregates nor reformats the IOAM data before it is exported. Raw export of IOAM data is to support an operational model where the processing and interpretation of IOAM data is decoupled from the operation of encapsulating/updating/decapsulating IOAM data, which is also referred to as IOAM data-plane operation. The figure below shows the separation of concerns for IOAM export: Exporting IOAM data is performed by the "IOAM node" which performs IOAM data-plane operation, whereas the interpretation of IOAM data is performed by the IOAM data processing system. The separation of concerns is to off-load interpretation, aggregation and formatting of IOAM data from the node which performs data-plane operations. In other words, a node which is focused on data-plane operations, i.e. forwarding of packets and handling IOAM data will not be tasked to also interpret the IOAM data, but can leave this task to another system. Note that for scalability reasons, a single IOAM node could choose to export IOAM data to several IOAM data processing systems.



IOAM node: IOAM encapsulating, IOAM decapsulating or IOAM transit node.

IOAM data processing system: System that receives raw IOAM data and provides for formatting, aggregation and interpretation of the IOAM data.

Monitoring/Analytics system: System that receives telemetry and other operational information from a variety of sources and provides for correlation and interpretation of the data received.

Raw export of IOAM data is typically generated by network devices at the edges of the network. Deployment and use-case dependent, such as in case of direct export [I-D.ietf-ippm-ioam-direct-export] or in cases where the operator is interested in dropped packets, raw export of IOAM data may be generated by IOAM transit nodes.

1.1. Requirements

Requirements for raw export of IOAM data:

- o Export all IOAM information contained in a packet.
- o Export a specific IOAM data type - Incremental Trace type, Preallocated Trace type, Proof of Transit type, Edge to Edge type, Direct Export type.
- o Export IOAM trace data associated with a packet, even if that data was never included in a transmitted or received packet in the network, for example in case of direct export.
- o Support coalescing of the IOAM data from multiple packets into a single raw export packet.
- o Support export of additional parts of the packet, other than the IOAM data as part of the raw export. This could be parts of the packet header and/or parts of the packet payload. This additional information provides context to the IOAM data (e.g. to be used for flow identification) and is to enable the IOAM data processing system to perform further analysis on the received data.
- o Report the reason why IOAM data was exported. The "reason for export" is to complement the IOAM data retrieved from the packet. For example, if a packet was dropped by a node due to congestion, it could be helpful to export the IOAM data of this dropped packet along with an indication that the packet that the IOAM data belongs to was dropped due to congestion.

1.2. Scope

This document discusses raw export of IOAM data using IPFIX.

The following is considered out of scope for this document:

- o Protocols other than IPFIX for raw export of IOAM data.
- o Interpretation or aggregation of IOAM data prior to exporting.
- o Configuration of network devices so that they can determine when to generate IOAM reports, and what information to include in those reports.
- o Events that trigger generation of IOAM reports.

- o Selection of particular destinations within distributed telemetry monitoring systems, to which IOAM reports will be sent.
- o Export format for flow statistics or processed/interpreted/aggregated IOAM data.

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Abbreviations used in this document:

E2E: Edge to Edge

IOAM: In-situ Operations, Administration, and Maintenance

MTU: Maximum Transmit Unit

OAM: Operations, Administration, and Maintenance

POT: Proof of Transit

3. IPFIX for IOAM raw data export

IPFIX, being a generic export protocol, can export any Information Elements as long as they are described in the information model. The IPFIX protocol is well suited for and is defined as the protocol for exporting packet samples in [RFC5476].

IPFIX/PSAMP [RFC7011], [RFC5476] already define many of the information elements needed for exporting sections of packets needed for deriving context and raw IOAM data export. This document specifies extensions of the IPFIX information model for meeting the requirements in Section 1.1.

3.1. Key IPFIX information elements leveraged for IOAM raw data export

The existing IPFIX Information Elements that are required for IOAM raw data export are listed here. Their details are available in IANA's IPFIX registry [IANA-IPFIX].

The existing IPFIX Information Elements used to carry the sections of the packets including IOAM data within it are as follows:

313 - ipHeaderPacketSection

315 - dataLinkFrameSection

The following Information Elements will be used to provide context to the ipHeaderPacketSection and dataLinkFrameSection as described in [IANA-IPFIX]:

408 - dataLinkFrameType

409 - sectionOffset

410 - sectionExportedOctets

The following Information Element will be used to provide forwarding status of the flow and any attached reasons.

89 - forwardingStatus

3.2. New IPFIX information elements leveraged for IOAM raw data export

IOAM data raw export using IPFIX requires a set of new information elements which are described in this section.

3.2.1. ioamReportFlags

Description:

This Information Element describes properties associated with an IOAM report.

The ioamReportFlags data type is an 8-bit field. The following bits are defined here:

Bit 0 Dropped Association - Dropped packet of interest.

Bit 1 Congested Queue Association - Indicates the presence of congestion on a monitored queue.

Bit 2 Tracked Flow Association - Matched a flow of interest.

Bit 3-7 Reserved

IANA is requested to create a new subregistry for IOAM Report Flags and fill it with the initial list from the description. New assignments for IOAM Encapsulation Types are administered by IANA through Expert Review [RFC5226] i.e., review by one of a group of experts designated by an IETF Area Director.

Abstract Data Type: unsigned8

Data Type Semantics: flags

ElementId: TBD1

Status: current

3.2.2. ioamEncapsulationType

Description:

This Information Element specifies the type of encapsulation to interpret ioamPreallocatedTraceData, ioamIncrementalTraceData, ioamE2EData, ioamPOTData, ioamDirectExportData.

The following ioamEncapsulationType values are defined here:

- 0 None : IOAM data follows format defined in [I-D.ietf-ippm-ioam-data]
- 1 GRE : IOAM data follows format defined in [I-D.weis-ippm-ioam-eth]
- 2 IPv6 : IOAM data follows format defined in [I-D.ietf-ippm-ioam-ipv6-options]
- 3 VXLAN-GPE : IOAM data follows format defined in [I-D.brockners-ippm-ioam-vxlan-gpe]
- 4 GENEVE Option: IOAM data follows format defined in [I-D.brockners-ippm-ioam-geneve]
- 5 GENEVE Next Protocol: IOAM data follows format defined in [I-D.weis-ippm-ioam-eth]
- 6 NSH : IOAM data follows format defined in [I-D.ietf-sfc-ioam-nsh]

IANA is requested to create a new subregistry for IOAM Encapsulation Types and fill it with the initial list from the description. New assignments for IOAM Encapsulation Types are administered by IANA through Expert Review [RFC5226] i.e., review by one of a group of experts designated by an IETF Area Director.

Abstract Data Type: unsigned8

Data Type Semantics: identifier

ElementId: TBD2

Status: current

3.2.3. ioamPreallocatedTraceData

Description:

This Information Element carries n octets of IOAM Preallocated Trace data defined in [I-D.ietf-ippm-ioam-data].

The format of the data is determined by the ioamEncapsulationType information element, if present. When the ioamEncapsulationType information element is present and has a value other than "None", and with sufficient length, this element may also report octets from subsequent headers and payload. If no ioamEncapsulationType information element is present, then the encapsulation type shall be assumed to be "None" and this information element only contains octets from the IOAM Preallocated Trace Option.

Abstract Data Type: octetArray

ElementId: TBD3

Status: current

3.2.4. ioamIncrementalTraceData

Description:

This Information Element carries n octets of IOAM Incremental Trace data defined in [I-D.ietf-ippm-ioam-data].

The format of the data is determined by the ioamEncapsulationType information element, if present. When the ioamEncapsulationType information element is present and has a value other than "None", and with sufficient length, this element may also report octets from subsequent headers and payload. If no ioamEncapsulationType information element is present, then the encapsulation type shall be assumed to be "None" and this information element only contains octets from the IOAM Incremental Trace Option.

Abstract Data Type: octetArray

ElementId: TBD4

Status: current

3.2.5. ioamE2EData

Description:

This Information Element carries n octets of IOAM E2E data defined in [I-D.ietf-ippm-ioam-data].

The format of the data is determined by the ioamEncapsulationType information element, if present. When the ioamEncapsulationType information element is present and has a value other than "None", and with sufficient length, this element may also report octets from subsequent headers and payload. If no ioamEncapsulationType information element is present, then the encapsulation type shall be assumed to be "None" and this information element only contains octets from the IOAM Edge-to-Edge Option.

Abstract Data Type: octetArray

ElementId: TBD5

Status: current

3.2.6. ioamPOTData

Description:

This Information Element carries n octets of IOAM POT data defined in [I-D.ietf-ippm-ioam-data].

The format of the data is determined by the ioamEncapsulationType information element, if present. When the ioamEncapsulationType information element is present and has a value other than "None", and with sufficient length, this element may also report octets from subsequent headers and payload. If no ioamEncapsulationType information element is present, then the encapsulation type shall be assumed to be "None" and this information element only contains octets from the IOAM Proof of Transit Option.

Abstract Data Type: octetArray

ElementId: TBD6

Status: current

3.2.7. ioamDirectExportData

Description:

This Information Element carries n octets of IOAM Direct Export data defined in [I-D.ietf-ippm-ioam-direct-export].

In addition to the fields from the IOAM Direct Export Option header in the packet, this information element includes all of the trace data from the exporting node, based on the IOAM-Trace-Type value. This data is appended inside ioamDirectExportData following the bit order of the IOAM-Trace-Type field, similar to the way that IOAM encapsulating nodes append trace data in Incremental Trace Option headers.

The format of the data is determined by the ioamEncapsulationType information element, if present. When the ioamEncapsulationType information element is present and has a value other than "None", and with sufficient length, this element may also report octets from subsequent headers and payload. If no ioamEncapsulationType information element is present, then the encapsulation type shall be assumed to be "None" and this information element only contains octets from the IOAM Direct Export Option plus the corresponding trace data.

Abstract Data Type: octetArray

ElementId: TBD7

Status: current

3.2.8. ipHeaderPacketSectionWithPadding

Description:

This Information Element carries a series of n octets from the IP header of a sampled packet, starting sectionOffset octets into the IP header.

However, if no sectionOffset field corresponding to this Information Element is present, then a sectionOffset of zero applies, and the octets MUST be from the start of the IP header.

With sufficient length, this element also reports octets from the IP payload. However, full packet capture of arbitrary packet streams is explicitly out of scope per the Security Considerations sections of [RFC5477] and [RFC2804].

When this Information Element has a fixed length, this MAY include padding octets that are used to fill out that fixed length.

When this information element has a variable length, the variable length MAY include up to 3 octets of padding, used to preserve 4-octet alignment of subsequent Information Elements or subsequent records within the same set.

In either case of fixed or variable length, the amount of populated octets MAY be specified in the sectionExportedOctets field corresponding to this Information Element, in which case the remainder (if any) MUST be padding. If there is no sectionExportedOctets field corresponding to this Information Element, then all octets MUST be populated unless the total length of the IP packet is less than the fixed length of this Information Element, in which case the remainder MUST be padding.

Abstract Data Type: octetArray

ElementId: TBD8

Status: current

3.2.9. ethernetFrameSection

Description:

This Information Element carries a series of n octets from the IEEE 802.3 Ethernet frame of a sampled packet, starting after the preamble and start frame delimiter (SFD), plus sectionOffset octets into the frame if there is a sectionOffset field corresponding to this Information Element.

With sufficient length, this element also reports octets from the Ethernet payload. However, full packet capture of arbitrary packet streams is explicitly out of scope per the Security Considerations sections of [RFC5477] and [RFC2804].

When this Information Element has a fixed length, this MAY include padding octets that are used to fill out that fixed length.

When this information element has a variable length, the variable length MAY include up to 3 octets of padding, used to preserve 4-octet alignment of subsequent Information Elements or subsequent records within the same set.

In either case of fixed or variable length, the amount of populated octets MAY be specified in the sectionExportedOctets field

corresponding to this Information Element, in which case the remainder (if any) MUST be padding. If there is no sectionExportedOctets field corresponding to this Information Element, then all octets MUST be populated unless the total length of the Ethernet frame is less than the fixed length of this Information Element, in which case the remainder MUST be padding.

Abstract Data Type: octetArray

ElementId: TBD9

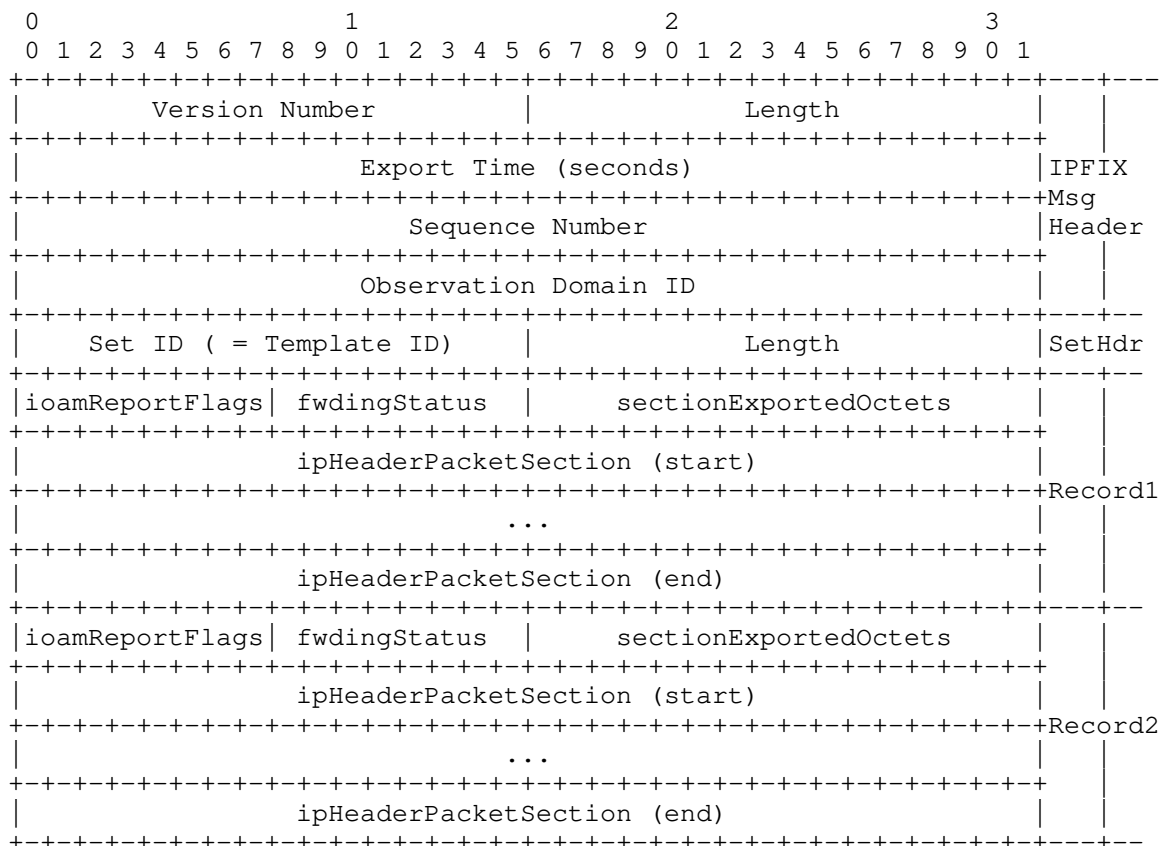
Status: current

4. Examples

This section shows a set of examples of how IOAM information along with other parts of the packet can be carried using IPFIX.

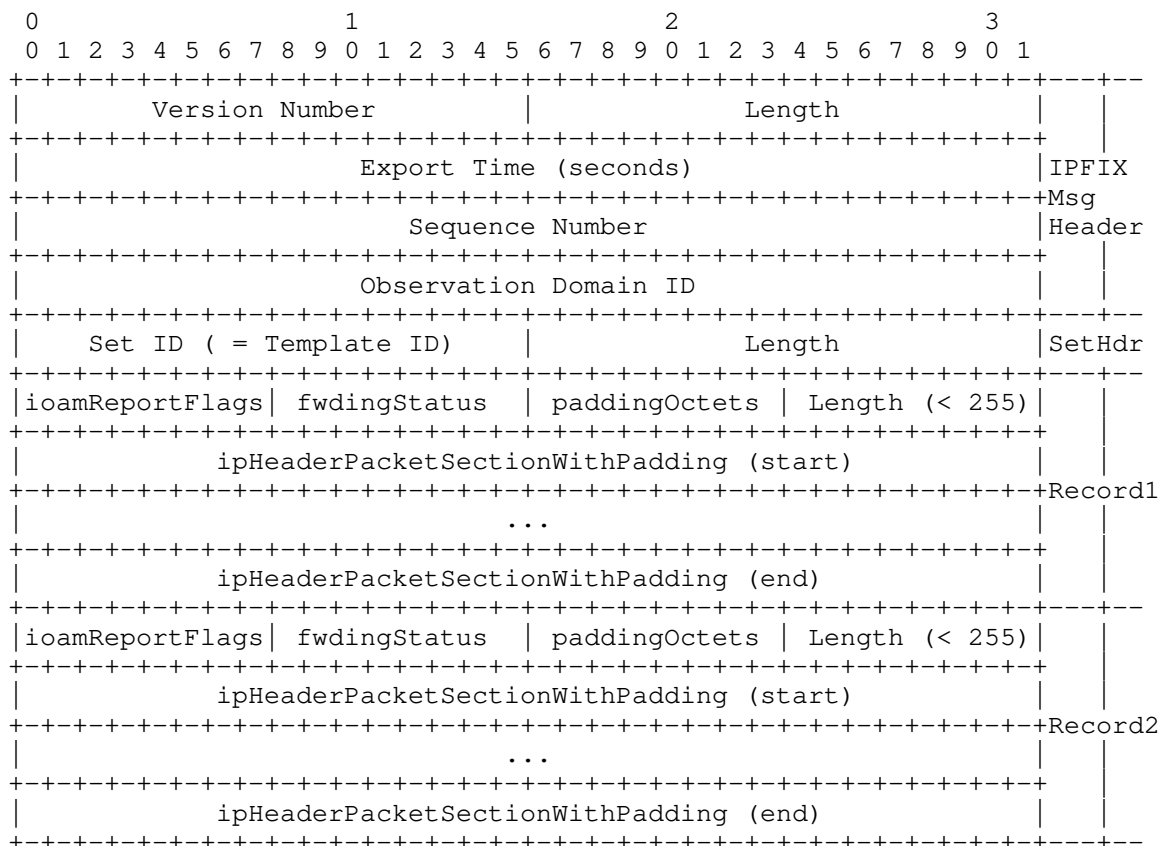
4.1. Fixed Length IP Packet

This example shows a fixed length IP packet. IOAM data is part of the ipHeaderPacketSection.



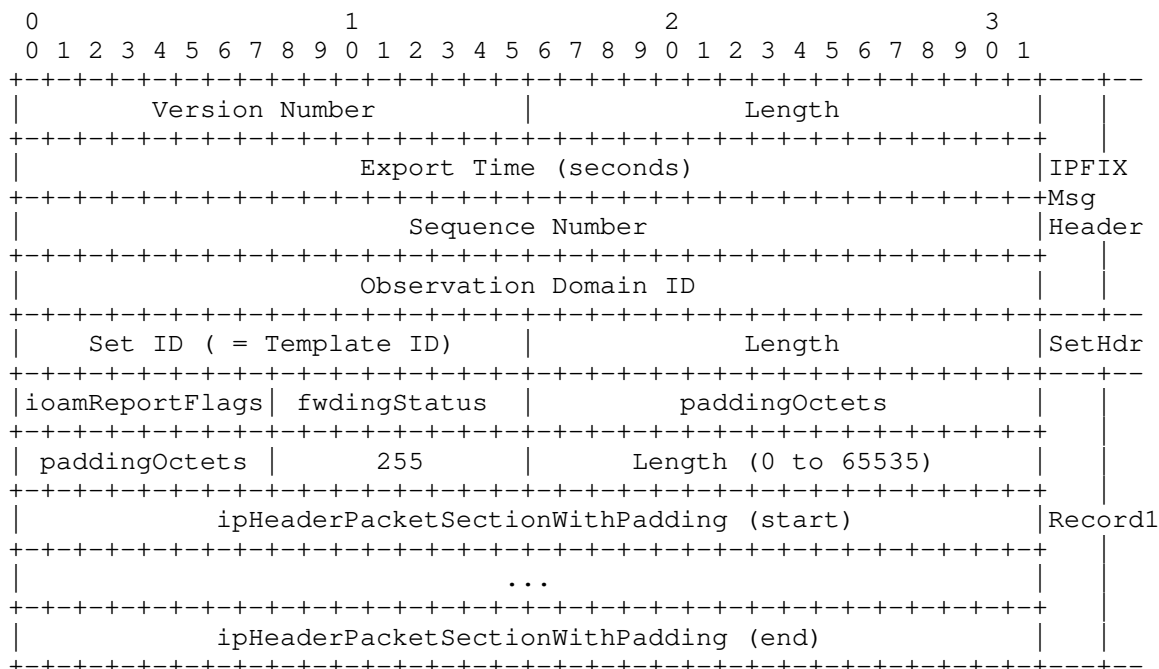
4.2. Variable Length IP Packet (length < 255)

This examples shows a variable length IP packet, with length < 255 bytes. IOAM data is part of the ipHeaderPacketSectionWithPadding.



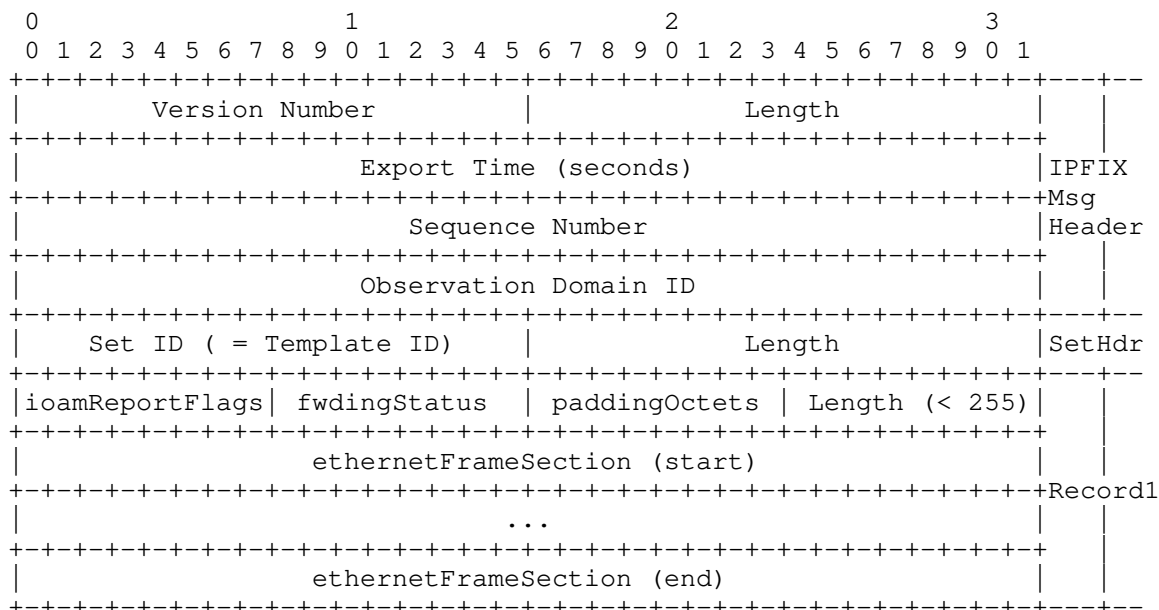
4.3. Variable Length IP Packet (length > 255)

This examples shows a variable length IP packet, with length > 255 bytes. IOAM data is part of the ipHeaderPacketSectionWithPadding.



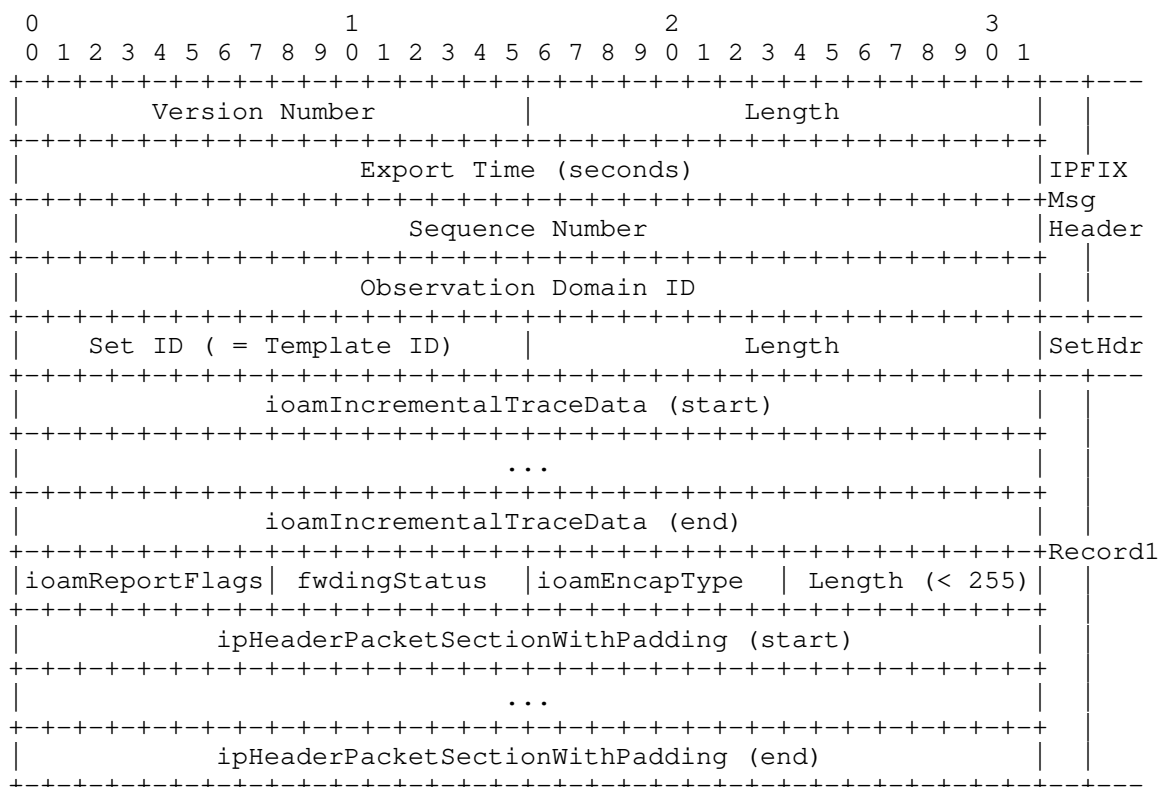
4.4. Variable Length ETHERNET Packet (length < 255)

This examples shows a variable length Ethernet packet, with length < 255 bytes. IOAM data is part of the ethernetFrameSection.



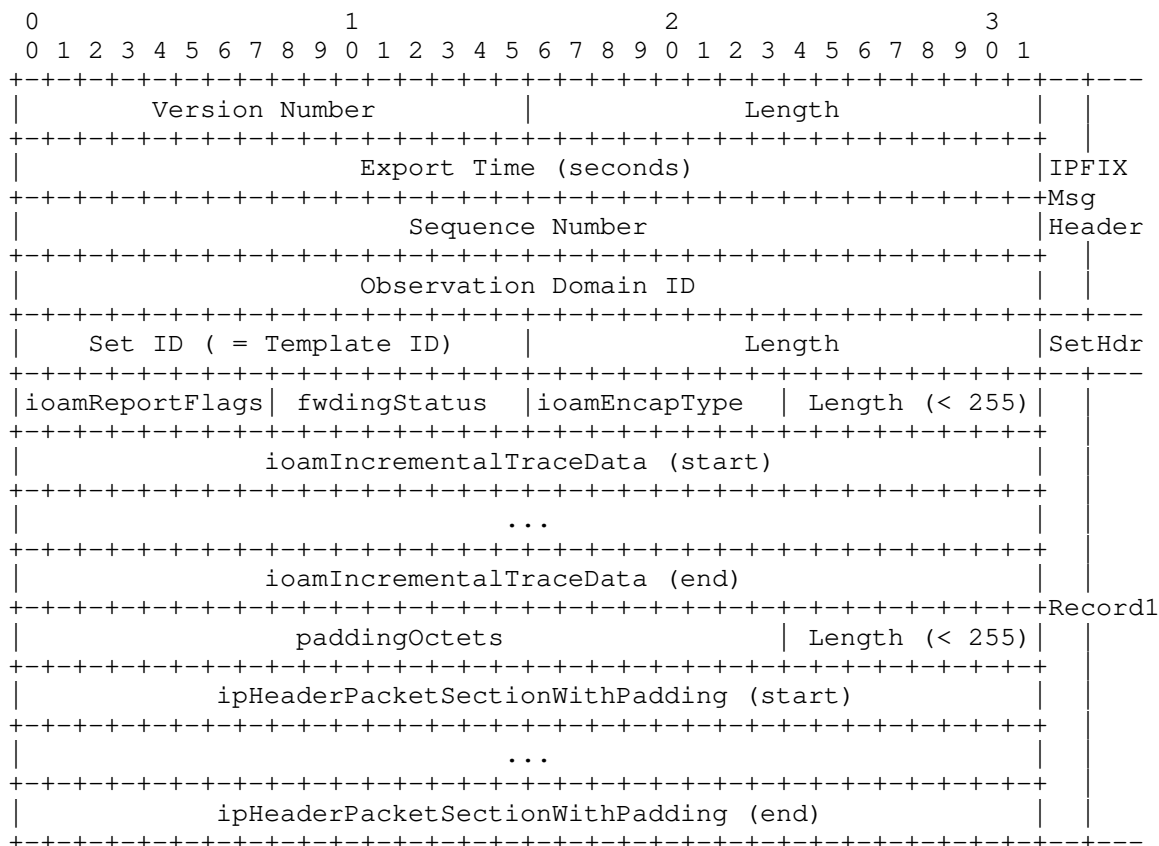
4.5. Variable Length IP Packet with Fixed Length IOAM Incremental Trace Data

This examples shows a variable length IP packet with length < 255 bytes and fixed length ioamIncrementalTraceData carried separately.



4.6. Variable Length IP Packet with Variable Length IOAM Incremental Trace Data

This examples shows a variable length IP packet with length < 255 bytes and variable length ioamIncrementalTraceData with length < 255 bytes carried separately.



5. IANA Considerations

IANA is requested to allocate code points for the following Information Elements in [IANA-IPFIX]:

TBD1 ioamReportFlags

TBD2 ioamEncapsulationType

TBD3 ioamPreallocatedTraceData

TBD4 ioamIncrementalTraceData

TBD5 ioamE2EData

TBD6 ioamPOTData

TBD7 ioamDirectExportData

TBD8 ipHeaderPacketSectionWithPadding

TBD9 ethernetFrameSection

See Section 3.2 for further details.

IANA is requested to create subregistries for ioamReportFlags defined in Section 3.2.1 and ioamEncapsulationType defined in Section 3.2.2.

6. Manageability Considerations

Manageability considerations will be addressed in a later version of this document.

7. Security Considerations

Security considerations will be addressed in a later version of this document.

8. Acknowledgements

The authors would like to thank Barak Gafni, Tal Mizrahi, Jennifer Lemon, and Aviv Kfir for their thoughts and comments on raw IOAM data export.

9. References

9.1. Normative References

- [I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-12 (work in progress), February 2021.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5476] Claise, B., Ed., Johnson, A., and J. Quittek, "Packet Sampling (PSAMP) Protocol Specifications", RFC 5476, DOI 10.17487/RFC5476, March 2009, <<https://www.rfc-editor.org/info/rfc5476>>.

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken,
"Specification of the IP Flow Information Export (IPFIX)
Protocol for the Exchange of Flow Information", STD 77,
RFC 7011, DOI 10.17487/RFC7011, September 2013,
<<https://www.rfc-editor.org/info/rfc7011>>.

9.2. Informative References

- [I-D.brockners-ippm-ioam-geneve]
Brockners, F., Bhandari, S., Govindan, V. P., Pignataro, C., Nainar, N. K., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Lapukhov, P., Gafni, B., Kfir, A., and M. Spiegel, "Geneve encapsulation for In-situ OAM Data", draft-brockners-ippm-ioam-geneve-05 (work in progress), November 2020.
- [I-D.brockners-ippm-ioam-vxlan-gpe]
Brockners, F., Bhandari, S., Govindan, V. P., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A., Gafni, B., Lapukhov, P., and M. Spiegel, "VXLAN-GPE Encapsulation for In-situ OAM Data", draft-brockners-ippm-ioam-vxlan-gpe-03 (work in progress), November 2019.
- [I-D.ietf-ippm-ioam-direct-export]
Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F., Bhandari, S., Sivakolundu, R., and T. Mizrahi, "In-situ OAM Direct Exporting", draft-ietf-ippm-ioam-direct-export-07 (work in progress), October 2021.
- [I-D.ietf-ippm-ioam-ipv6-options]
Bhandari, S. and F. Brockners, "In-situ OAM IPv6 Options", draft-ietf-ippm-ioam-ipv6-options-07 (work in progress), February 2022.
- [I-D.ietf-sfc-ioam-nsh]
Brockners, F. and S. Bhandari, "Network Service Header (NSH) Encapsulation for In-situ OAM (IOAM) Data", draft-ietf-sfc-ioam-nsh-07 (work in progress), January 2022.
- [I-D.weis-ippm-ioam-eth]
Weis, B., Brockners, F., Hill, C., Bhandari, S., Govindan, V. P., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A., Gafni, B., Lapukhov, P., and M. Spiegel, "EtherType Protocol Identification of In-situ OAM Data", draft-weis-ippm-ioam-eth-04 (work in progress), May 2020.

- [IANA-IPFIX] "IP Flow Information Export (IPFIX) Entities",
<<https://www.iana.org/assignments/ipfix/ipfix.xhtml>>.
- [RFC2804] IAB and IESG, "IETF Policy on Wiretapping", RFC 2804,
DOI 10.17487/RFC2804, May 2000,
<<https://www.rfc-editor.org/info/rfc2804>>.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an
IANA Considerations Section in RFCs", RFC 5226,
DOI 10.17487/RFC5226, May 2008,
<<https://www.rfc-editor.org/info/rfc5226>>.
- [RFC5477] Dietz, T., Claise, B., Aitken, P., Dressler, F., and G.
Carle, "Information Model for Packet Sampling Exports",
RFC 5477, DOI 10.17487/RFC5477, March 2009,
<<https://www.rfc-editor.org/info/rfc5477>>.

Authors' Addresses

Mickey Spiegel
Barefoot Networks, an Intel company
4750 Patrick Henry Drive
Santa Clara, CA 95054
US

Email: mickey.spiegel@intel.com

Frank Brockners
Cisco Systems, Inc.
Hansaallee 249, 3rd Floor
DUESSELDORF, NORDRHEIN-WESTFALEN 40549
Germany

Email: fbrockne@cisco.com

Shwetha Bhandari
Thoughtspot
3rd Floor, Indiqube Orion, 24th Main Rd, Garden Layout, HSR Layout
Bangalore, KARNATAKA 560 102
India

Email: shwetha.bhandari@thoughtspot.com

Ramesh Sivakolundu
Cisco Systems, Inc.
170 West Tasman Dr.
SAN JOSE, CA 95134
USA

Email: sramesh@cisco.com

ippm
Internet-Draft
Intended status: Standards Track
Expires: September 4, 2018

B. Weis
F. Brockners
C. Hill
S. Bhandari
V. Govindan
C. Pignataro
Cisco
H. Gredler
RtBrick Inc.
J. Leddy
Comcast
S. Youell
J MPC
T. Mizrahi
Marvell
A. Kfir
B. Gafni
Mellanox Technologies, Inc.
P. Lapukhov
Facebook
M. Spiegel
Barefoot Networks
March 03, 2018

GRE Encapsulation for In-situ OAM Data
draft-weis-ippm-ioam-gre-00

Abstract

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information in the packet while the packet traverses a path between two points in the network. This document outlines how IOAM data fields are encapsulated in GRE.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 4, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Conventions	3
2.1. Requirements Language	3
2.2. Abbreviations	3
3. In-situ OAM Metadata Transport in GRE	3
4. Security Considerations	5
5. IANA Considerations	5
6. References	5
6.1. Normative References	5
6.2. Informative References	6
Appendix A. Example GRE-IOAM Payloads	6
A.1. Example GRE-IOAM Tracing Payloads	6
Authors' Addresses	6

1. Introduction

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information in the packet while the packet traverses a path between two points in the network. This document outlines how IOAM data fields are encapsulated in the Generic Routing Encapsulation (GRE) [RFC2784].

2. Conventions

2.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.2. Abbreviations

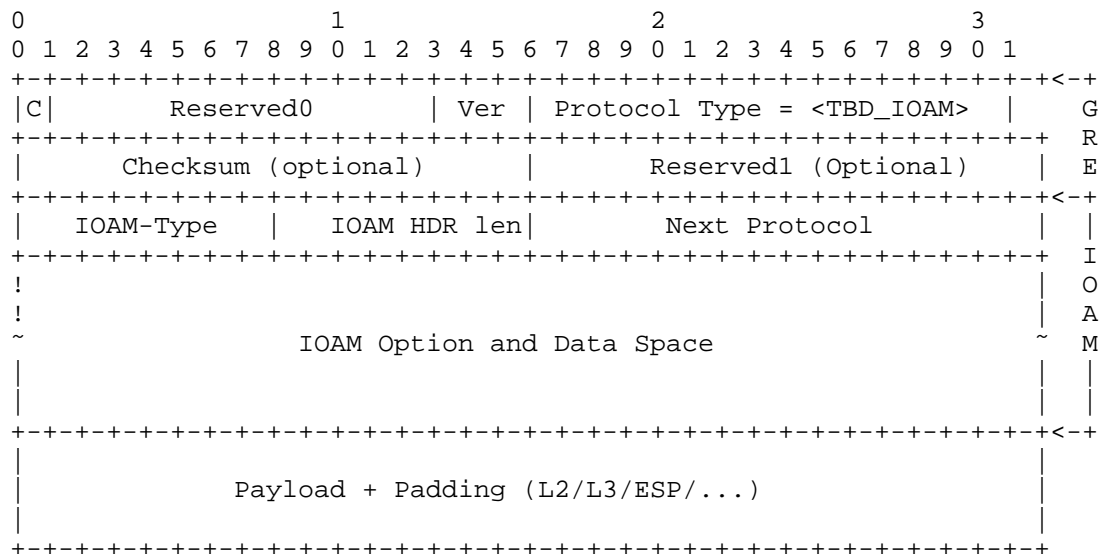
Abbreviations used in this document:

E2E:	Edge-to-Edge
GRE:	Generic Routing Encapsulation
IOAM:	In-situ Operations, Administration, and Maintenance
OAM:	Operations, Administration, and Maintenance
POT:	Proof of Transit

3. In-situ OAM Metadata Transport in GRE

GRE encapsulation is defined in [RFC2784]. IOAM encapsulation in GRE follows the GRE header.

IOAM data fields are carried in GRE using a Protocol Type value of TBD_IOAM. An IOAM header is added containing the different IOAM data fields defined in [I-D.ietf-ippm-ioam-data]. In an administrative domain where IOAM is used, insertion of the IOAM protocol header in GRE is enabled at the GRE tunnel endpoints, which also serve as IOAM encapsulating/decapsulating nodes by means of configuration.



The GRE header and fields are defined in [RFC2784]. The GRE Protocol Type value is TBD_IOAM.

The IOAM header is defined as follows.

IOAM Type: 8-bit field defining the IOAM Option type, as defined in Section 7.2 of [I-D.ietf-ippm-ioam-data].

IOAM HDR Len: 8 bits Length field contains the length of the variable IOAM data octets in 4-octet units.

Next Protocol: 16 bits Next Protocol Type field contains the protocol type of the packet following IOAM protocol header. When the most significant octet is 0x00, the Protocol Type is taken to be an IP Protocol Number as defined in [IP-PROT]. Otherwise, the Protocol Type is defined to be an EtherType value from [ETYPES]. An implementation receiving a packet containing a Protocol Type which is not listed in one of those registries SHOULD discard the packet.

IOAM Option and Data Space: IOAM option header and data is present as specified by the IOAM-Type field, and is defined in Section 4 of [I-D.ietf-ippm-ioam-data].

Multiple IOAM options MAY be included within the GRE encapsulation. For example, if a GRE encapsulation contains two IOAM options before a data packet, the Next Protocol field of the first IOAM option will contain the value of TBD IOAM, while the Next Protocol field of the

second IOAM option will contain the Ethertype or IP protocol Number indicating the type of the data packet.

4. Security Considerations

This document describes the encapsulation of IOAM data fields in GRE. Security considerations of the specific IOAM data fields for each case (i.e., Trace, Proof of Transit, and E2E) are described in defined in [I-D.ietf-ippm-ioam-data].

As this document describes new protocol fields within the existing GRE encapsulation, these are similar to the security considerations of [RFC2784].

IOAM data transported in an OAM E2E header SHOULD be integrity protected (e.g., with IPsec ESP [RFC4303]) to detect changes made by a device between the sending and receiving OAM endpoints.

5. IANA Considerations

A new EtherType value is requested to be added to the [ETYPES] IANA registry. The description should be "In-situ OAM (IOAM)".

6. References

6.1. Normative References

- [ETYPES] "IANA Ethernet Numbers",
<<https://www.iana.org/assignments/ieee-802-numbers/ieee-802-numbers.xhtml>>.
- [I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., Chang, R., and d. daniel.bernier@bell.ca, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-01 (work in progress), October 2017.
- [IP-PROT] "IANA Protocol Numbers",
<<https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

- [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <<https://www.rfc-editor.org/info/rfc2784>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

6.2. Informative References

- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<https://www.rfc-editor.org/info/rfc4303>>.

Appendix A. Example GRE-IOAM Payloads

A.1. Example GRE-IOAM Tracing Payloads

TBD

Authors' Addresses

Brian Weis
Cisco Systems, Inc.
170 W. Tasman Drive
San Jose, California 95134-1706
USA

Phone: +1-408-526-4796
Email: bew@cisco.com

Frank Brockners
Cisco Systems, Inc.
Hansaallee 249, 3rd Floor
DUESSELDORF, NORDRHEIN-WESTFALEN 40549
Germany

Email: fbrockne@cisco.com

Craig Hill
Cisco Systems, Inc.
13600 Dulles Technology Drive
Herndon, Virginia 20171
United States

Email: crhill@cisco.com

Shwetha Bhandari
Cisco Systems, Inc.
Cessna Business Park, Sarjapura Marathalli Outer Ring Road
Bangalore, KARNATAKA 560 087
India

Email: shwethab@cisco.com

Vengada Prasad Govindan
Cisco Systems, Inc.

Email: venggovi@cisco.com

Carlos Pignataro
Cisco Systems, Inc.
7200-11 Kit Creek Road
Research Triangle Park, NC 27709
United States

Email: cpignata@cisco.com

Hannes Gredler
RtBrick Inc.

Email: hannes@rtbrick.com

John Leddy
Comcast

Email: John_Leddy@cable.comcast.com

Stephen Youell
JP Morgan Chase
25 Bank Street
London E14 5JP
United Kingdom

Email: stephen.youell@jpmorgan.com

Tal Mizrahi
Marvell
6 Hamada St.
Yokneam 20692
Israel

Email: talmi@marvell.com

Aviv Kfir
Mellanox Technologies, Inc.
350 Oakmead Parkway, Suite 100
Sunnyvale, CA 94085
U.S.A.

Email: avivk@mellanox.com

Barak Gafni
Mellanox Technologies, Inc.
350 Oakmead Parkway, Suite 100
Sunnyvale, CA 94085
U.S.A.

Email: gbarak@mellanox.com

Petr Lapukhov
Facebook
1 Hacker Way
Menlo Park, CA 94025
US

Email: petr@fb.com

Mickey Spiegel
Barefoot Networks
2185 Park Boulevard
Palo Alto, CA 94306
US

Email: mspiegel@barefootnetworks.com

IPPM
Internet-Draft
Intended status: Standards Track
Expires: October 10, 2018

T. Zhou, Ed.
J. Guichard
Huawei
F. Brockners
S. Raghavan
Cisco Systems
April 8, 2018

A YANG Data Model for In-Situ OAM
draft-zhou-ippm-ioam-yang-01

Abstract

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information in user packets while the packets traverse a path between two points in the network. This document defines a YANG module for the IOAM function.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 10, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Tree Diagrams	3
2. Design of the IOAM YANG Data Model	3
2.1. Profiles	3
2.2. Preallocated Tracing Profile	4
2.3. Incremental Tracing Profile	5
2.4. Proof of Transit Profile	5
2.5. Edge to Edge Profile	5
3. IOAM YANG Module	6
4. IANA Considerations	16
5. Security Considerations	16
6. Acknowledgements	16
7. References	17
7.1. Normative References	17
7.2. Informative References	17
Appendix A. Examples	18
Authors' Addresses	18

1. Introduction

In-situ Operations, Administration, and Maintenance (IOAM) [I-D.ietf-ippm-ioam-data] records OAM information within user packets while the packets traverse a network. The data types and data formats for IOAM data records have been defined in [I-D.ietf-ippm-ioam-data]. The IOAM data can be embedded in many protocol encapsulations such as Network Services Header, Segment Routing, and IPv6 [I-D.brockners-inband-oam-transport].

This document defines a data model for IOAM capabilities using the YANG data modeling language [RFC7950]. This YANG model supports all the three categories of IOAM data, which are Tracing Option, Proof of Transit Option, and Edge-to-Edge Option.

1.1. Tree Diagrams

The meaning of the symbols in these diagrams is as follows:

- o Brackets "[" and "]" enclose list keys.
- o Curly braces "{" and "}" contain names of optional features that make the corresponding node conditional.
- o Abbreviations before data node names: "rw" means configuration (read-write), "ro" state data (read-only).
- o Symbols after data node names: "?" means an optional node, "!" a container with presence, and "*" denotes a "list" or "leaf-list".
- o Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- o Ellipsis ("...") stands for contents of subtrees that are not shown.

2. Design of the IOAM YANG Data Model

2.1. Profiles

The IOAM model is organized as list of profiles as shown in the following figure. Each profile associates with one flow and the corresponding IOAM information.

```

+--rw ioam
  +--rw ioam-profiles
    +--rw enabled?          boolean
    +--rw ioam-profile* [profile-name]
      +--rw profile-name          string
      +--rw filter
        | +--rw filter-type?    ioam-filter-type
        | +--rw acl-name?       -> /acl:acls/acl/name
      +--rw protocol-type?      ioam-protocol-type
      +--rw incremental-tracing-profile {incremental-trace}?
        | ...
      +--rw preallocated-tracing-profile {preallocated-trace}?
        | ...
      +--rw pot-profile {proof-of-transit}?
        | ...
      +--rw e2e-profile {edge-to-edge}?
        ...

```

The "enabled" is an administrative configuration. When it is set to true, IOAM configuration is enabled for the system. Meanwhile, the IOAM data-plane functionality is enabled.

The "filter" is used to identify a flow, where the IOAM profile can apply. There may be multiple filter types. ACL is the default one.

The IOAM data can be encapsulated into multiple protocols, e.g., IPv6 [RFC8200], Geneve [I-D.ietf-nvo3-geneve], VxLAN-GPE [I-D.ietf-nvo3-vxlan-gpe]. The "protocol-type" is used to indicate where the IOAM is applied. For example, if the "protocol-type" is IPv6, the IOAM ingress node will encapsulate the associated flow with the IPv6-IOAM [I-D.brockners-inband-oam-transport] format.

IOAM data includes three usage options with four encapsulation types, i.e., incremental tracing data, preallocated tracing data, prove of transit data and end to end data. In practice, multiple IOAM data types can be encapsulated into the same IOAM header. The "ioam-profile" contains a set of sub-profiles, each of which relates to one encapsulation type. The configured object may not support all the sub-profiles. The supported sub-profiles are indicated by 4 defined features, i.e., "incremental-trace", "preallocated-trace", "proof-of-transit", "edge-to-edge".

2.2. Preallocated Tracing Profile

The IOAM tracing data is expected to be collected at every node that a packet traverses to ensure visibility into the entire path a packet takes within an IOAM domain. The preallocated tracing option will create pre-allocated space for each node to populate its information. The "preallocated-tracing-profile" contains the detailed information for the preallocated tracing data. The information includes:

- o enabled: indicates whether the preallocated tracing profile is enabled.
- o node-action: indicates the operation (e.g., encapsulate IOAM header, transit the IOAM data, or decapsulate IOAM header) applied to the dedicated flow.
- o trace-type: indicates the per-hop data to be captured by the IOAM enabled nodes and included in the node data list.
- o Loopback mode is used to send a copy of a packet back towards the source.

```
+--rw preallocated-tracing-profile {preallocated-trace}?
  +--rw enabled?                boolean
  +--rw node-action?            ioam-node-action
  +--rw trace-type?            ioam-trace-types
  +--rw enable-loopback-mode?   boolean
```

2.3. Incremental Tracing Profile

The incremental tracing option contains a variable node data fields where each node allocates and pushes its node data immediately following the option header. The "incremental-tracing-profile" contains the detailed information for the incremental tracing data. The detailed information is the same as the Preallocated Tracing Profile, but with one more variable, "max-length", which restricts the length of the IOAM header.

```
+--rw incremental-tracing-profile {incremental-trace}?
  +--rw enabled?                boolean
  +--rw node-action?            ioam-node-action
  +--rw trace-type?            ioam-trace-types
  +--rw enable-loopback-mode?   boolean
  +--rw max-length?            uint32
```

2.4. Proof of Transit Profile

The IOAM Proof of Transit data is to support the path or service function chain verification use cases. The "pot-profile" contains the detailed information for the prove of transit data. The detailed information are described in [I-D.brockners-proof-of-transit].

```
+--rw pot-profile {proof-of-transit}?
  +--rw enabled?                boolean
  +--rw active-profile-index?    pot:profile-index-range
  +--rw pot-profile-list* [pot-profile-index]
  +--rw pot-profile-index       profile-index-range
  +--rw prime-number            uint64
  +--rw secret-share            uint64
  +--rw public-polynomial       uint64
  +--rw lpc                    uint64
  +--rw validator?              boolean
  +--rw validator-key?          uint64
  +--rw bitmask?                uint64
```

2.5. Edge to Edge Profile

The IOAM edge to edge option is to carry data that is added by the IOAM encapsulating node and interpreted by IOAM decapsulating node.

The "e2e-profile" contains the detailed information for the edge to edge data. The detailed information includes:

- o enabled: indicates whether the edge to edge profile is enabled.
- o node-action is the same semantic as in Section 2.2.
- o e2e-type indicates data to be carried from the ingress IOAM node to the egress IOAM node.

```
+++rw e2e-profile {edge-to-edge}?  
    +-rw enabled?                boolean  
    +-rw node-action?            ioam-node-action  
    +-rw e2e-type?               ioam-e2e-types
```

3. IOAM YANG Module

```
<CODE BEGINS> file "ietf-ioam@2018-04-08.yang"  
module ietf-ioam {  
  yang-version 1.1;  
  namespace "urn:ietf:params:xml:ns:yang:ietf-ioam";  
  prefix "ioam";  
  
  import ietf-pot-profile {  
    prefix "pot";  
  }  
  
  import ietf-access-control-list {  
    prefix "acl";  
  }  
  
  organization  
    "IETF IPPM (IP Performance Metrics) Working Group";  
  
  contact  
    "WG Web: <http://tools.ietf.org/wg/ippm>  
    WG List: <ippm@ietf.org>  
    Editor: zhoutianran@huawei.com";  
  
  description  
    "This YANG module specifies a vendor-independent data  
    model for the in Situ OAM (iOAM).";  
  
  revision 2018-04-08 {  
    description "Initial revision.";  
    reference "draft-zhou-ippm-ioam-yang";  
  }
```

```
/*
 * FEATURES
 */

feature incremental-trace
{
  description
    "This feature indicated that the incremental tracing mode is
    supported";
}

feature preallocated-trace
{
  description
    "This feature indicated that the preallocated tracing mode is
    supported";
}

feature proof-of-transit
{
  description
    "This feature indicated that the proof of transit mode is
    supported";
}

feature edge-to-edge
{
  description
    "This feature indicated that the edge to edge mode is
    supported";
}

/*
 * IDENTITIES
 */
identity base-filter {
  description
    "Base identity to represent a filter. A filter is used to
    specify the flow to apply the iOAM profile. ";
}

identity acl-filter {
  base base-filter;
  description
    "Apply ACL rule to specify the flow.";
}

identity base-protocol {
```



```
    description
      "Base identity to represent the carrier protocol. It's used to
       indicate what layer and protocol the iOAM data is embedded.";
  }

  identity ipv6-protocol {
    base base-protocol;
    description
      "The described iOAM data is embedded in ipv6 protocol.";
  }

  identity base-node-action {
    description
      "Base identity to represent the node actions. It's used to
       indicate what action the node will take.";
  }

  identity encapsulate {
    base base-node-action;
    description
      "indicate the node is to encapsulate the iOAM packet";
  }

  identity transit {
    base base-node-action;
    description
      "indicate the node is to transit the iOAM packet";
  }

  identity decapsulate {
    base base-node-action;
    description
      "indicate the node is to decapsulate the iOAM packet";
  }

/*
 * TYPE DEFINITIONS
 */

typedef ioam-filter-type {
  type identityref {
    base base-filter;
  }
  description
    "Specifies a known type of filter.";
}

typedef ioam-protocol-type {
```

```
    type identityref {
      base base-protocol;
    }
    description
      "Specifies a known type of carrier protocol for the iOAM data.";
  }

  typedef ioam-node-action {
    type identityref {
      base base-node-action;
    }
    description
      "Specifies a known type of node action.";
  }

  typedef ioam-trace-types {
    type bits {
      bit ioam-hop-lim-node-id {
        position 0;
        description
          "When set indicates presence of Hop_Lim and node_id in the
           node data.";
      }
      bit ioam-if-id {
        position 1;
        description
          "When set indicates presence of ingress_if_id and
           egress_if_id in the node data.";
      }
      bit ioam-timestamp-seconds {
        position 2;
        description
          "When set indicates presence of time stamp seconds in the
           node data.";
      }
      bit ioam-timestamp-nanoseconds {
        position 3;
        description
          "When set indicates presence of time stamp nanoseconds in
           the node data.";
      }
      bit ioam-transit-delay {
        position 4;
        description
          "When set indicates presence of transit delay in the node
           data.";
      }
      bit ioam-app-data {
```

```
        position 5;
        description
            "When set indicates presence of app_data in the node data.";
    }
    bit ioam-queue-depth {
        position 6;
        description
            "When set indicates presence of queue depth in the node
            data.";
    }
    bit ioam-opaque-state-snapshot {
        position 7;
        description
            "When set indicates presence of variable length Opaque
            State Snapshot field.";
    }
    bit ioam-hop-lim-node-id-wide {
        position 8;
        description
            "When set indicates presence of Hop_Lim and node_id wide
            in the node data.";
    }
    bit ioam-if-id-wide {
        position 9;
        description
            "When set indicates presence of ingress_if_id and
            egress_if_id wide in the node data.";
    }
    bit app-data-wide {
        position 10;
        description
            "When set indicates presence of app_data wide in the node
            data.";
    }
}
description
    "A 16-bit identifier which specifies which data types are used
    in this node data list.";
}

typedef ioam-pot-types {
    type bits {
        bit ioam-bytes-16 {
            position 0;
            description
                "POT data is a 16 Octet field";
        }
    }
}
```

```
    description
      "7-bit identifier of a particular POT variant that dictates
       the POT data that is included.";
  }

  typedef ioam-e2e-types {
    type bits {
      bit ioam-seq-num {
        position 0;
        description
          "A 64-bit sequence number added to a specific tube which is
           used to identify packet loss and reordering for that
           tube.";
      }
    }
    description
      "8-bit identifier of a particular in situ OAM E2E variant.";
  }

/*
 * GROUP DEFINITIONS
 */

  grouping ioam-filter {
    description "A grouping for iOAM filter definition";

    leaf filter-type {
      type ioam-filter-type;
      description "filter type";
    }

    leaf acl-name {
      when "../filter-type = 'acl-filter'";
      type leafref {
        path "/acl:acls/acl:acl/acl:name";
      }
      description "Access Control List name.";
    }
  }

  grouping ioam-incremental-tracing-profile {
    description
      "A grouping for incremental tracing profile.";

    leaf node-action {
      type ioam-node-action;
      description "node action";
    }
  }
```

```
leaf trace-type {
  when "../node-action = 'encapsulate'";
  type ioam-trace-types;
  description
    "The trace type is only defined at the encapsulation node.";
}

leaf enable-loopback-mode {
  when "../node-action = 'encapsulate'";
  type boolean;
  default false;
  description
    "Loopback mode is used to send a copy of a packet back towards
    the source. The loopback mode is only defined at the
    encapsulation node.";
}

leaf max-length {
  when "../node-action = 'encapsulate'";
  type uint32;
  description
    "This field specifies the maximum length of the node data list
    in octets. The max-length is only defined at the
    encapsulation node. And it's only used for the incremental
    tracing mode.";
}
}

grouping ioam-preallocated-tracing-profile {
  description
    "A grouping for incremental tracing profile.";

  leaf node-action {
    type ioam-node-action;
    description "node action";
  }

  leaf trace-type {
    when "../node-action = 'encapsulate'";
    type ioam-trace-types;
    description
      "The trace type is only defined at the encapsulation node.";
  }

  leaf enable-loopback-mode {
    when "../node-action = 'encapsulate'";
    type boolean;
    default false;
  }
}
```

```
        description
            "Loopback mode is used to send a copy of a packet back towards
            the source. The loopback mode is only defined at the
            encapsulation node.";
    }
}

grouping ioam-e2e-profile {
    description
        "A grouping for tracing profile.";

    leaf node-action {
        type ioam-node-action;
        description
            "indicate how the node act for this profile";
    }

    leaf e2e-type {
        when "../node-action = 'encapsulate'";
        type ioam-e2e-types;
        description
            "The e2e type is only defined at the encapsulation node.";
    }
}

grouping ioam-admin-config {
    description
        "IOAM top-level administrative configuration.";

    leaf enabled {
        type boolean;
        default false;
        description
            "When true, IOAM configuration is enabled for the system.
            Meanwhile, the IOAM data-plane functionality is enabled.";
    }
}

/*
 * DATA NODES
 */

container ioam {
    description "iOAM top level container";

    container ioam-profiles {
        description
            "Contains a list of iOAM profiles.";
    }
}
```

```
uses ioam-admin-config;

list ioam-profile {
  key "profile-name";
  ordered-by user;
  description
    "A list of iOAM profiles that configured on the node.";

  leaf profile-name {
    type string;
    description
      "Unique identifier for each iOAM profile";
  }

  container filter {
    uses ioam-filter;
    description
      "The filter which is used to indicate the flow to apply
      iOAM.";
  }

  leaf protocol-type {
    type ioam-protocol-type;
    description
      "This item is used to indicate the carrier protocol where
      the iOAM is applied.";
  }

  container incremental-tracing-profile {
    if-feature incremental-trace;
    description
      "describe the profile for incremental tracing option";

    leaf enabled {
      type boolean;
      default false;
      description
        "When true, apply incremental tracing option to the
        specified flow identified by the filter.";
    }

    uses ioam-incremental-tracing-profile;
  }

  container preallocated-tracing-profile {
    if-feature preallocated-trace;
    description
      "describe the profile for preallocated tracing option";
  }
}
```

```
    leaf enabled {
      type boolean;
      default false;
      description
        "When true, apply preallocated tracing option to the
        specified flow identified by the following filter.";
    }

    uses ioam-preallocated-tracing-profile;
  }

  container pot-profile {
    if-feature proof-of-transit;
    description
      "describe the profile for pot option";

    leaf enabled {
      type boolean;
      default false;
      description
        "When true, apply Proof of Transit option to the
        specified flow identified by the following filter.";
    }

    leaf active-profile-index {
      type pot:profile-index-range;
      description
        "Proof of transit profile index that is currently
        active. Will be set in the first hop of the path
        or chain. Other nodes will not use this field.";
    }

    uses pot:pot-profile;
  }

  container e2e-profile {
    if-feature edge-to-edge;
    description
      "describe the profile for e2e option";

    leaf enabled {
      type boolean;
      default false;
      description
        "When true, apply End to end option to the
        specified flow identified by the following filter.";
    }
  }
```



```
        uses ioam-e2e-profile;
    }
}
}
}
<CODE ENDS>
```

4. IANA Considerations

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number (and remove this note).

This document requests IANA to register the following URI in the "IETF XML Registry" [RFC3688]:

```
URI: urn:ietf:params:xml:ns:yang:ietf-ioam
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.
```

This document requests IANA to register the following YANG module in the "YANG Module Names" registry [RFC7950].

```
name: ietf-ioam
namespace: urn:ietf:params:xml:ns:yang:ietf-ioam
prefix: nat
reference: RFC XXXX
```

5. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the support of SSH is mandatory to implement secure transport [RFC6242]. The NETCONF access control model [RFC6536] provides means to restrict access by some users to a pre-configured subset of all available NETCONF protocol operations and data. All data nodes defined in the YANG module which can be created, modified and deleted (i.e., config true, which is the default). These data nodes are considered sensitive. Write operations (e.g., edit-config) applied to these data nodes without proper protection can negatively affect network operations.

6. Acknowledgements

For their valuable comments, discussions, and feedback, we wish to acknowledge Greg Mirsky and Reshad Rahman.

7. References

7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <<https://www.rfc-editor.org/info/rfc6536>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.

7.2. Informative References

- [I-D.brockners-inband-oam-transport] Brockners, F., Bhandari, S., Govindan, V., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., and R. Chang, "Encapsulations for In-situ OAM Data", draft-brockners-inband-oam-transport-05 (work in progress), July 2017.

[I-D.brockners-proof-of-transit]

Brockners, F., Bhandari, S., Dara, S., Pignataro, C., Leddy, J., Youell, S., Mozes, D., and T. Mizrahi, "Proof of Transit", draft-brockners-proof-of-transit-04 (work in progress), October 2017.

[I-D.ietf-ippm-ioam-data]

Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., Chang, R., daniel.bernier@bell.ca, d., and J. Lemon, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-02 (work in progress), March 2018.

[I-D.ietf-nvo3-geneve]

Gross, J., Ganga, I., and T. Sridhar, "Geneve: Generic Network Virtualization Encapsulation", draft-ietf-nvo3-geneve-06 (work in progress), March 2018.

[I-D.ietf-nvo3-vxlan-gpe]

Maino, F., Kreeger, L., and U. Elzur, "Generic Protocol Extension for VXLAN", draft-ietf-nvo3-vxlan-gpe-05 (work in progress), October 2017.

Appendix A. Examples

TBD

Authors' Addresses

Tianran Zhou
Huawei
156 Beiqing Rd.
Beijing 100095
China

Phone: 15810236238
Email: zhoutianran@huawei.com

Jim Guichard
Huawei
USA

Email: james.n.guichard@huawei.com

Frank Brockners
Cisco Systems
Hansaallee 249, 3rd Floor
DUESSELDORF, NORDRHEIN-WESTFALEN 40549
Germany

Email: fbrockne@cisco.com

Srihari Raghavan
Cisco Systems
Tril Infopark Sez, Ramanujan IT City
Neville Block, 2nd floor, Old Mahabalipuram Road
Chennai, Tamil Nadu 600113
India

Email: srihari@cisco.com

IPPM
Internet-Draft
Intended status: Standards Track
Expires: January 31, 2021

T. Zhou, Ed.
Huawei
J. Guichard
Futurewei
F. Brockners
S. Raghavan
Cisco Systems
July 30, 2020

A YANG Data Model for In-Situ OAM
draft-zhou-ippm-ioam-yang-08

Abstract

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information in user packets while the packets traverse a path between two points in the network. This document defines a YANG module for the IOAM function.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 31, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Conventions used in this document	2
2.1. Tree Diagrams	3
3. Design of the IOAM YANG Data Model	3
3.1. Profiles	3
3.2. Preallocated Tracing Profile	5
3.3. Incremental Tracing Profile	5
3.4. Direct Export Profile	6
3.5. Proof of Transit Profile	6
3.6. Edge to Edge Profile	7
4. IOAM YANG Module	7
5. Security Considerations	21
6. IANA Considerations	22
7. Acknowledgements	22
8. References	22
8.1. Normative References	23
8.2. Informative References	24
Authors' Addresses	24

1. Introduction

In-situ Operations, Administration, and Maintenance (IOAM) [I-D.ietf-ippm-ioam-data] records OAM information within user packets while the packets traverse a network. The data types and data formats for IOAM data records have been defined in [I-D.ietf-ippm-ioam-data]. The IOAM data can be embedded in many protocol encapsulations such as Network Services Header (NSH) and IPv6.

This document defines a data model for IOAM capabilities using the YANG data modeling language [RFC7950]. This YANG model supports all the five IOAM options, which are Incremental Tracing Option, Pre-allocated Tracing Option, Direct Export Option [I-D.ietf-ippm-ioam-direct-export], Proof of Transit (PoT) Option, and Edge-to-Edge Option.

2. Conventions used in this document

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in

BCP14, [RFC2119], [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following terms are defined in [RFC7950] and are used in this specification:

- o augment
- o data model
- o data node

The terminology for describing YANG data models is found in [RFC7950].

2.1. Tree Diagrams

Tree diagrams used in this document follow the notation defined in [RFC8340].

3. Design of the IOAM YANG Data Model

3.1. Profiles

The IOAM model is organized as list of profiles as shown in the following figure. Each profile associates with one flow and the corresponding IOAM information.

```

module: ietf-ioam
+--rw ioam
  +--rw ioam-profiles
    +--rw admin-config
      | +--rw enabled?    boolean
    +--rw ioam-profile* [profile-name]
      +--rw profile-name          string
      +--rw filter
        | +--rw filter-type?    ioam-filter-type
        | +--rw acl-name?      -> /acl:acls/acl/name
      +--rw protocol-type?      ioam-protocol-type
      +--rw incremental-tracing-profile {incremental-trace}?
        | ...
      +--rw preallocated-tracing-profile {preallocated-trace}?
        | ...
      +--rw direct-export-profile {direct-export}?
        | ...
      +--rw pot-profile {proof-of-transit}?
        | ...
      +--rw e2e-profile {edge-to-edge}?
        | ...

```

The "enabled" is an administrative configuration. When it is set to true, IOAM configuration is enabled for the system. Meanwhile, the IOAM data-plane functionality is enabled.

The "filter" is used to identify a flow, where the IOAM profile can apply. There may be multiple filter types. ACL [RFC8519] is the default one.

The IOAM data can be encapsulated into multiple protocols, e.g., IPv6 [I-D.ietf-ippm-ioam-ipv6-options] and NSH [I-D.ietf-sfc-ioam-nsh]. The "protocol-type" is used to indicate where the IOAM is applied. For example, if the "protocol-type" is IPv6, the IOAM ingress node will encapsulate the associated flow with the IPv6-IOAM [I-D.ietf-ippm-ioam-ipv6-options] format.

IOAM data includes five encapsulation types, i.e., incremental tracing data, preallocated tracing data, direct export data, prove of transit data and end to end data. In practice, multiple IOAM data types can be encapsulated into the same IOAM header. The "ioam-profile" contains a set of sub-profiles, each of which relates to one encapsulation type. The configured object may not support all the sub-profiles. The supported sub-profiles are indicated by 5 defined features, i.e., "incremental-trace", "preallocated-trace", "direct export", "proof-of-transit", "edge-to-edge".

3.2. Preallocated Tracing Profile

The IOAM tracing data is expected to be collected at every node that a packet traverses to ensure visibility into the entire path a packet takes within an IOAM domain. The preallocated tracing option will create pre-allocated space for each node to populate its information. The "preallocated-tracing-profile" contains the detailed information for the preallocated tracing data. The information includes:

- o enabled: indicates whether the preallocated tracing profile is enabled.
- o node-action: indicates the operation (e.g., encapsulate IOAM header, transit the IOAM data, or decapsulate IOAM header) applied to the dedicated flow.
- o use-namespace: indicate the namespace used for the trace types.
- o trace-type: indicates the per-hop data to be captured by the IOAM enabled nodes and included in the node data list.
- o Loopback mode is used to send a copy of a packet back towards the source.
- o Active mode indicates that a packet is used for active measurement.

```

+--rw preallocated-tracing-profile {preallocated-trace}?
  +--rw enabled?                boolean
  +--rw node-action?            ioam-node-action
  +--rw trace-types
    | +--rw use-namespace?      ioam-namespace
    | +--rw trace-type*         ioam-trace-type
  +--rw enable-loopback-mode?    boolean
  +--rw enable-active-mode?      boolean

```

3.3. Incremental Tracing Profile

The incremental tracing option contains a variable node data fields where each node allocates and pushes its node data immediately following the option header. The "incremental-tracing-profile" contains the detailed information for the incremental tracing data. The detailed information is the same as the Preallocated Tracing Profile, but with one more variable, "max-length", which restricts the length of the IOAM header.

```
+--rw incremental-tracing-profile {incremental-trace}?
  +--rw enabled?                boolean
  +--rw node-action?            ioam-node-action
  +--rw trace-types
    | +--rw use-namespace?      ioam-namespace
    | +--rw trace-type*         ioam-trace-type
  +--rw enable-loopback-mode?    boolean
  +--rw enable-active-mode?      boolean
  +--rw max-length?              uint32
```

3.4. Direct Export Profile

The direct export option is used as a trigger for IOAM nodes to export IOAM data to a receiving entity (or entities). The "direct-export-profile" contains the detailed information for the direct export data. The detailed information is the same as the Preallocated Tracing Profile, but with one more optional variable, "flow-id", which is used to correlate the exported data of the same flow from multiple nodes and from multiple packets.

```
+--rw direct-export-profile {direct-export}?
  +--rw enabled?                boolean
  +--rw node-action?            ioam-node-action
  +--rw trace-types
    | +--rw use-namespace?      ioam-namespace
    | +--rw trace-type*         ioam-trace-type
  +--rw enable-loopback-mode?    boolean
  +--rw enable-active-mode?      boolean
  +--rw flow-id?                 uint32
```

3.5. Proof of Transit Profile

The IOAM Proof of Transit data is to support the path or service function chain verification use cases. The "pot-profile" contains the detailed information for the prove of transit data. The detailed information are described in [I-D.ietf-sfc-proof-of-transit].

```

+--rw pot-profile {proof-of-transit}?
  +--rw enabled?                boolean
  +--rw active-profile-index?    pot:profile-index-range
  +--rw pot-profile-list* [pot-profile-index]
    +--rw pot-profile-index      profile-index-range
    +--rw prime-number           uint64
    +--rw secret-share           uint64
    +--rw public-polynomial       uint64
    +--rw lpc                    uint64
    +--rw validator?             boolean
    +--rw validator-key?         uint64
    +--rw bitmask?              uint64
      +--rw opot-masks
      +--rw downstream-mask*     uint64
      +--rw upstream-mask*       uint64

```

3.6. Edge to Edge Profile

The IOAM edge to edge option is to carry data that is added by the IOAM encapsulating node and interpreted by IOAM decapsulating node. The "e2e-profile" contains the detailed information for the edge to edge data. The detailed information includes:

- o enabled: indicates whether the edge to edge profile is enabled.
- o node-action is the same semantic as in Section 2.2.
- o use-namespace: indicate the namespace used for the edge to edge types.
- o e2e-type indicates data to be carried from the ingress IOAM node to the egress IOAM node.

```

+--rw e2e-profile {edge-to-edge}?
  +--rw enabled?                boolean
  +--rw node-action?            ioam-node-action
  +--rw e2e-types
    +--rw use-namespace?        ioam-namespace
    +--rw e2e-type*              ioam-e2e-type

```

4. IOAM YANG Module

```

<CODE BEGINS> file "ietf-ioam@2020-07-13.yang"
module ietf-ioam {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-ioam";
  prefix "ioam";

```

```
import ietf-pot-profile {
  prefix "pot";
  reference "draft-ietf-sfc-proof-of-transit";
}

import ietf-access-control-list {
  prefix "acl";
  reference
    "RFC 8519: YANG Data Model for Network Access Control
     Lists (ACLs)";
}

organization
  "IETF IPPM (IP Performance Metrics) Working Group";

contact
  "WG Web: <http://tools.ietf.org/wg/ippm>
  WG List: <ippm@ietf.org>
  Editor: zhoutianran@huawei.com
  Editor: james.n.guichard@futurewei.com
  Editor: fbrockne@cisco.com
  Editor: srihari@cisco.com";

description
  "This YANG module specifies a vendor-independent data
  model for the In Situ OAM (IOAM).

  Copyright (c) 2020 IETF Trust and the persons identified as
  authors of the code. All rights reserved.

  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject
  to the license terms contained in, the Simplified BSD License
  set forth in Section 4.c of the IETF Trust's Legal Provisions
  Relating to IETF Documents
  (http://trustee.ietf.org/license-info).

  This version of this YANG module is part of RFC XXXX; see the
  RFC itself for full legal notices."

revision 2020-07-13 {
  description "Initial revision.";
  reference "draft-zhou-ippm-ioam-yang";
}

/*
 * FEATURES
 */
```

```
feature incremental-trace
{
  description
    "This feature indicated that the incremental tracing option is
    supported";
  reference "draft-ietf-ippm-ioam-data";
}

feature preallocated-trace
{
  description
    "This feature indicated that the preallocated tracing option is
    supported";
  reference "draft-ietf-ippm-ioam-data";
}

feature direct-export
{
  description
    "This feature indicated that the direct export option is
    supported";
  reference "ietf-ippm-ioam-direct-export";
}

feature proof-of-transit
{
  description
    "This feature indicated that the proof of transit option is
    supported";
  reference "draft-ietf-ippm-ioam-data";
}

feature edge-to-edge
{
  description
    "This feature indicated that the edge to edge option is
    supported";
  reference "draft-ietf-ippm-ioam-data";
}

/*
 * IDENTITIES
 */
identity base-filter {
  description
    "Base identity to represent a filter. A filter is used to
    specify the flow to apply the IOAM profile. ";
}
```

```
identity acl-filter {
  base base-filter;
  description
    "Apply ACL rules to specify the flow.";
}

identity base-protocol {
  description
    "Base identity to represent the carrier protocol. It's used to
    indicate what layer and protocol the IOAM data is embedded.";
}

identity ipv6-protocol {
  base base-protocol;
  description
    "The described IOAM data is embedded in IPv6 protocol.";
  reference "ietf-ippm-ioam-ipv6-options";
}

identity nsh-protocol {
  base base-protocol;
  description
    "The described IOAM data is embedded in NSH.";
  reference "ietf-sfc-ioam-nsh";
}

identity base-node-action {
  description
    "Base identity to represent the node actions. It's used to
    indicate what action the node will take.";
}

identity action-encapsulate {
  base base-node-action;
  description
    "indicate the node is to encapsulate the IOAM packet";
}

identity action-transit {
  base base-node-action;
  description
    "indicate the node is to transit the IOAM packet";
}

identity action-decapsulate {
  base base-node-action;
  description
    "indicate the node is to decapsulate the IOAM packet";
}
```

```
}

identity base-trace-type {
  description
    "Base identity to represent trace types";
}

identity trace-hop-lim-node-id {
  base base-trace-type;
  description
    "indicates presence of Hop_Lim and node_id in the
    node data.";
}

identity trace-if-id {
  base base-trace-type;
  description
    "indicates presence of ingress_if_id and egress_if_id in the
    node data.";
}

identity trace-timestamp-seconds {
  base base-trace-type;
  description
    "indicates presence of time stamp seconds in the node data.";
}

identity trace-timestamp-nanoseconds {
  base base-trace-type;
  description
    "indicates presence of time stamp nanoseconds in the node data.";
}

identity trace-transit-delay {
  base base-trace-type;
  description
    "indicates presence of transit delay in the node data.";
}

identity trace-namespace-data {
  base base-trace-type;
  description
    "indicates presence of namespace specific data (short format)
    in the node data.";
}

identity trace-queue-depth {
  base base-trace-type;
```

```
    description
      "indicates presence of queue depth in the node data.";
  }

  identity trace-opaque-state-snapshot {
    base base-trace-type;
    description
      "indicates presence of variable length Opaque State Snapshot
      field.";
  }

  identity trace-hop-lim-node-id-wide {
    base base-trace-type;
    description
      "indicates presence of Hop_Lim and node_id wide in the
      node data.";
  }

  identity trace-if-id-wide {
    base base-trace-type;
    description
      "indicates presence of ingress_if_id and egress_if_id wide in
      the node data.";
  }

  identity trace-namespace-data-wide {
    base base-trace-type;
    description
      "indicates presence of namespace specific data in wide format
      in the node data.";
  }

  identity trace-buffer-occupancy {
    base base-trace-type;
    description
      "indicates presence of buffer occupancy in the node data.";
  }

  identity trace-checksum-complement {
    base base-trace-type;
    description
      "indicates presence of the Checksum Complement node data.";
  }

  identity base-pot-type {
    description
      "Base identity to represent Proof of Transit(PoT) types";
  }
```



```
identity pot-bytes-16 {
  base base-pot-type;
  description
    "POT data is a 16 Octet field.";
}

identity base-e2e-type {
  description
    "Base identity to represent e2e types";
}

identity e2e-seq-num-64 {
  base base-e2e-type;
  description
    "indicates presence of a 64-bit sequence number";
}

identity e2e-seq-num-32 {
  base base-e2e-type;
  description
    "indicates presence of a 32-bit sequence number";
}

identity e2e-timestamp-seconds {
  base base-e2e-type;
  description
    "indicates presence of timestamp seconds for the
    transmission of the frame";
}

identity e2e-timestamp-subseconds {
  base base-e2e-type;
  description
    "indicates presence of timestamp subseconds for the
    transmission of the frame";
}

identity base-namespace {
  description
    "Base identity to represent the namespace";
}

identity namespace-ietf {
  base base-namespace;
  description
    "namespace that specified in IETF.";
}
```

```
/*
 * TYPE DEFINITIONS
 */

typedef ioam-filter-type {
  type identityref {
    base base-filter;
  }
  description
    "Specifies a known type of filter.";
}

typedef ioam-protocol-type {
  type identityref {
    base base-protocol;
  }
  description
    "Specifies a known type of carrier protocol for the IOAM data.";
}

typedef ioam-node-action {
  type identityref {
    base base-node-action;
  }
  description
    "Specifies a known type of node action.";
}

typedef ioam-trace-type {
  type identityref {
    base base-trace-type;
  }
  description
    "Specifies a known trace type.";
}

typedef ioam-pot-type {
  type identityref {
    base base-pot-type;
  }
  description
    "Specifies a known pot type.";
}

typedef ioam-e2e-type {
  type identityref {
    base base-e2e-type;
  }
}
```

```
    description
      "Specifies a known e2e type.";
  }

  typedef ioam-namespace {
    type identityref {
      base base-namespace;
    }
    description
      "Specifies the supported namespace.";
  }

/*
 * GROUP DEFINITIONS
 */

  grouping ioam-filter {
    description "A grouping for IOAM filter definition";

    leaf filter-type {
      type ioam-filter-type;
      description "filter type";
    }

    leaf acl-name {
      when "../filter-type = 'ioam:acl-filter'";
      type leafref {
        path "/acl:acls/acl:acl/acl:name";
      }
      description "Access Control List name.";
    }
  }

  grouping encap-tracing {
    description
      "A grouping for the generic configuration for
      tracing profile.";

    container trace-types {
      description
        "the list of trace types for encapsulate";

      leaf use-namespace {
        type ioam-namespace;
        description
          "the namespace used for the encapsulation";
      }
    }
  }
```

```
    leaf-list trace-type {
      type ioam-trace-type;
      description
        "The trace type is only defined at the encapsulation node.";
    }
  }

  leaf enable-loopback-mode {
    type boolean;
    default false;
    description
      "Loopback mode is used to send a copy of a packet back towards
       the source. The loopback mode is only defined at the
       encapsulation node.";
  }

  leaf enable-active-mode {
    type boolean;
    default false;
    description
      "Active mode indicates that a packet is used for active
       measurement. An IOAM decapsulating node that receives a
       packet with the Active flag set in one of its Trace options
       must terminate the packet.";
  }
}

grouping ioam-incremental-tracing-profile {
  description
    "A grouping for incremental tracing profile.";

  leaf node-action {
    type ioam-node-action;
    description "node action";
  }

  uses encap-tracing {
    when "node-action = 'ioam:action-encapsulate'";
  }

  leaf max-length {
    when "../node-action = 'ioam:action-encapsulate'";
    type uint32;
    description
      "This field specifies the maximum length of the node data list
       in octets. The max-length is only defined at the
       encapsulation node. And it's only used for the incremental
       tracing mode.";
  }
}
```

```
    }  
  }  
  
  grouping ioam-preallocated-tracing-profile {  
    description  
      "A grouping for incremental tracing profile.";  
  
    leaf node-action {  
      type ioam-node-action;  
      description "node action";  
    }  
  
    uses encap-tracing {  
      when "node-action = 'ioam:action-encapsulate'";  
    }  
  }  
  
  grouping ioam-direct-export-profile {  
    description  
      "A grouping for direct export profile.";  
  
    leaf node-action {  
      type ioam-node-action;  
      description "node action";  
    }  
  
    uses encap-tracing {  
      when "node-action = 'ioam:action-encapsulate'";  
    }  
  
    leaf flow-id {  
      when "../node-action = 'ioam:action-encapsulate'";  
      type uint32;  
      description  
        "flow-id is used to correlate the exported data of the same  
        flow from multiple nodes and from multiple packets.";  
    }  
  }  
  
  grouping ioam-e2e-profile {  
    description  
      "A grouping for end to end profile.";  
  
    leaf node-action {  
      type ioam-node-action;  
      description  
        "indicate how the node act for this profile";  
    }  
  }  
}
```

```
    }

    container e2e-types {
      when "../node-action = 'ioam:action-encapsulate'";
      description
        "the list of e2e types for encapsulate";

      leaf use-namespace {
        type ioam-namespace;
        description
          "the namespace used for the encapsulation";
      }

      leaf-list e2e-type {
        type ioam-e2e-type;
        description
          "The e2e type is only defined at the encapsulation node.";
      }
    }
  }

  grouping ioam-admin-config {
    description
      "IOAM top-level administrative configuration.";

    leaf enabled {
      type boolean;
      default false;
      description
        "When true, IOAM configuration is enabled for the system.
        Meanwhile, the IOAM data-plane functionality is enabled.";
    }
  }
}

/*
 * DATA NODES
 */

container ioam {
  description "IOAM top level container";

  container ioam-profiles {
    description
      "Contains a list of IOAM profiles.";

    container admin-config {
      description
        "Contains all the administrative configurations related to
```

```
        the IOAM functionalities and all the IOAM profiles.";

    uses ioam-admin-config;
}

list ioam-profile {
    key "profile-name";
    ordered-by user;
    description
        "A list of IOAM profiles that configured on the node.";

    leaf profile-name {
        type string;
        mandatory true;
        description
            "Unique identifier for each IOAM profile";
    }

    container filter {
        uses ioam-filter;
        description
            "The filter which is used to indicate the flow to apply
            IOAM.";
    }

    leaf protocol-type {
        type ioam-protocol-type;
        description
            "This item is used to indicate the carrier protocol where
            the IOAM is applied.";
    }

    container incremental-tracing-profile {
        if-feature incremental-trace;
        description
            "describe the profile for incremental tracing option";

        leaf enabled {
            type boolean;
            default false;
            description
                "When true, apply incremental tracing option to the
                specified flow identified by the filter.";
        }

        uses ioam-incremental-tracing-profile;
    }
}
```

```
container preallocated-tracing-profile {
  if-feature preallocated-trace;
  description
    "describe the profile for preallocated tracing option";

  leaf enabled {
    type boolean;
    default false;
    description
      "When true, apply preallocated tracing option to the
       specified flow identified by the following filter.";
  }

  uses ioam-preallocated-tracing-profile;
}

container direct-export-profile {
  if-feature direct-export;
  description
    "describe the profile for direct-export option";

  leaf enabled {
    type boolean;
    default false;
    description
      "When true, apply direct-export option to the
       specified flow identified by the following filter.";
  }

  uses ioam-direct-export-profile;
}

container pot-profile {
  if-feature proof-of-transit;
  description
    "describe the profile for PoT option";

  leaf enabled {
    type boolean;
    default false;
    description
      "When true, apply Proof of Transit option to the
       specified flow identified by the following filter.";
  }

  leaf active-profile-index {
    type pot:profile-index-range;
    description
```



```

        "Proof of transit profile index that is currently
        active. Will be set in the first hop of the path
        or chain. Other nodes will not use this field.";
    }

    uses pot:pot-profile;
}

container e2e-profile {
    if-feature edge-to-edge;
    description
        "describe the profile for e2e option";

    leaf enabled {
        type boolean;
        default false;
        description
            "When true, apply End to end option to the
            specified flow identified by the following filter.";
    }

    uses ioam-e2e-profile;
}
}
}
}
<CODE ENDS>
```

5. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC5246].

The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config)

to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

- o /ioam/ioam-profiles/admin-config

The items in the container above include the top level administrative configurations related to the IOAM functionalities and all the IOAM profiles. Unexpected changes to these items could lead to the IOAM function disruption and/ or misbehavior of all the IOAM profiles.

- o /ioam/ioam-profiles/ioam-profile

The entries in the list above include the whole IOAM profile configurations which indirectly create or modify the device configurations. Unexpected changes to these entries could lead to the mistake of the IOAM behavior for the corresponding flows.

6. IANA Considerations

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number (and remove this note).

IANA is requested to assign a new URI from the IETF XML Registry [RFC3688]. The following URI is suggested:

URI: urn:ietf:params:xml:ns:yang:ietf-ioam
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.

This document also requests a new YANG module name in the YANG Module Names registry [RFC7950] with the following suggestion:

name: ietf-ioam
namespace: urn:ietf:params:xml:ns:yang:ietf-ioam
prefix: ioam
reference: RFC XXXX

7. Acknowledgements

For their valuable comments, discussions, and feedback, we wish to acknowledge Greg Mirsky, Reshad Rahman and Tom Petch.

8. References

8.1. Normative References

- [I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-10 (work in progress), July 2020.
- [I-D.ietf-ippm-ioam-direct-export]
Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F., Bhandari, S., Sivakolundu, R., and T. Mizrahi, "In-situ OAM Direct Exporting", draft-ietf-ippm-ioam-direct-export-00 (work in progress), February 2020.
- [I-D.ietf-sfc-proof-of-transit]
Brockners, F., Bhandari, S., Mizrahi, T., Dara, S., and S. Youell, "Proof of Transit", draft-ietf-sfc-proof-of-transit-06 (work in progress), June 2020.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/info/rfc5246>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <<https://www.rfc-editor.org/info/rfc6536>>.

- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8519] Jethanandani, M., Agarwal, S., Huang, L., and D. Blair, "YANG Data Model for Network Access Control Lists (ACLs)", RFC 8519, DOI 10.17487/RFC8519, March 2019, <<https://www.rfc-editor.org/info/rfc8519>>.

8.2. Informative References

- [I-D.ietf-ippm-ioam-ipv6-options]
Bhandari, S., Brockners, F., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A., Gafni, B., Lapukhov, P., Spiegel, M., Krishnan, S., and R. Asati, "In-situ OAM IPv6 Options", draft-ietf-ippm-ioam-ipv6-options-02 (work in progress), July 2020.
- [I-D.ietf-sfc-ioam-nsh]
Brockners, F. and S. Bhandari, "Network Service Header (NSH) Encapsulation for In-situ OAM (IOAM) Data", draft-ietf-sfc-ioam-nsh-04 (work in progress), June 2020.

Authors' Addresses

Tianran Zhou
Huawei
156 Beiqing Rd.
Beijing 100095
China

Email: zhoutianran@huawei.com

Jim Guichard
Futurewei
United States of America

Email: james.n.guichard@futurewei.com

Frank Brockners
Cisco Systems
Hansaallee 249, 3rd Floor
Duesseldorf, Nordrhein-Westfalen 40549
Germany

Email: fbrockne@cisco.com

Srihari Raghavan
Cisco Systems
Tril Infopark Sez, Ramanujan IT City
Neville Block, 2nd floor, Old Mahabalipuram Road
Chennai, Tamil Nadu 600113
India

Email: srihari@cisco.com