

Network Working Group
Internet-Draft
Updates: 2330 (if approved)
Intended status: Standards Track
Expires: April 29, 2018

J. Alvarez-Hamelin
Universidad de Buenos Aires
A. Morton
AT&T Labs
J. Fabini
TU Wien
October 26, 2017

Advanced Unidirectional Route Assessment
draft-amf-ippm-route-01

Abstract

This memo introduces an advanced unidirectional route assessment 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.

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 April 29, 2018.

Copyright Notice

Copyright (c) 2017 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 Route	3
2. Scope	4
3. Route Metric Terms and Definitions	5
3.1. Formal Name	5
3.2. Parameters	6
3.3. Metric Definitions	6
3.4. Related Round-Trip Delay and Loss Definitions	8
3.5. Discussion	8
3.6. Reporting the Metric	9
4. Route Assessment Methodologies	9
4.1. Active Methodologies	10
4.2. Hybrid Methodologies	11
4.3. Combining Different Methods	12
5. Background on Round-Trip Delay Measurement Goals	13
6. Tools to Measure Delays in the Internet	14
7. RTD Measurements Statistics	15
8. Conclusions	16
9. Security Considerations	17
10. IANA Considerations	17
11. Acknowledgements	17
12. References	17
12.1. Normative References	17
12.2. Informative References	20
Authors' Addresses	21

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 path components 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.

1.1. Issues with Earlier Work to define Route

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 [PT] rely on TTL 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: This a reality of Internet paths 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 TTL or 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 TTL or Hop Count, or cooperate with interrogation protocols. The presence of tunnels and nested tunnels further complicate assessment by hiding hops.

Hop: Although the [RFC2330] definition was a link-host pair, only hosts 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.

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 any two hosts.

<@@@ or only hosts communicating at the IP layer? We would have to re-define the Src and Dst Parameters and Host Identity if we generalize beyond IP. Should we include MPLS and the capabilities of [RFC8029], with explicit multipath identification (section 6.2.6)? >

Also, to specify a framework for active methods of measurement which use the techniques described in [PT] at a minimum, and 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] (intended only for single administrative domains), which do not rely on ICMP and 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 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 Terms and Definitions

This section sets requirements for the following components to support the Route Metric:

Note: the definitions concentrate on the IP-layer, but can be extended to other layers, and follow agreements on the scope.

Host Identity For hosts communicating at the IP-layer, the globally routable IP address(es) which the host uses when communicating with other hosts under normal or error conditions. The Host Identity revealed (and its connection to a Host Name through reverse DNS) determines whether interfaces to parallel links can be associated with a single host, or appear to be unique hosts.

Discoverable Host For hosts communicating at the IP-layer, 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 host identity) to the source. This requirement is also consistent with section 5.3.1 of [RFC1812] for routers.

Cooperating Host Hosts MUST respond to direct queries for their host identity as part of a previously agreed and established interrogation protocol. Hosts SHOULD also provide information such as arrival/departure interface identification, arrival timestamp, and any relevant information about the host or specific link which delivered the query to the host.

Hop A Hop MUST contain a Host Identity, and MAY contain arrival and/or departure interface identification.

3.1. Formal Name

Type-P-Route-Ensemble-Method-Variant, abbreviated as Route Ensemble.

Note that Type-P depends heavily on the chosen method and variant.

3.2. Parameters

This section lists the REQUIRED input factors to specify a Route metric.

- o Src, the IP address of a host
- o Dst, the IP address of a host
- o i, the TTL or Hop Limit of a packet sent from the host at Src to the host at Dst.
- o MaxHops, the maximum value of i used, (i=1,2,3,...MaxHops).
- o T0, a time (start of measurement interval)
- o Tf, a time (end of measurement interval)
- o T, the host time of a packet as measured at MP(Src), meaning Measurement Point at the Source.
- o Ta, the host time of a reply packet's *arrival* as measured at MP(Src), assigned to packets that arrive within a "reasonable" time (see parameter below).
- o Tmax, 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), thus the distribution of 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) results in identical treatment in a multi-path decision (such as that taken in load balancing).
- o Type-P, the complete description of the packets for which this assessment applies (including the flow-defining fields).

3.3. Metric Definitions

This section defines the REQUIRED measurement components of the Route metrics (unless otherwise indicated):

M, the total number of packets sent between T0 and Tf.

N, the smallest value of i needed for a packet to be received at Dst (sent between T_0 and T_f).

Nmax, the largest value of i needed for a packet to be received at Dst (sent between T_0 and T_f). Nmax may be equal to N.

Next, define a **singleton** definition for a Hop on the path, with sufficient indexes to identify all Hops identified in a measurement interval.

A Hop, designated $h(i,j)$, the IP address and/or identity of one of j Discoverable Hosts (or Cooperating Hosts) that are i hops away from the host with IP address = Src during the measurement interval, T_0 to T_f . As defined above, a Hop singleton measurement MUST contain a Host Identity, $hid(i,j)$, and MAY contain one or more of the following attributes:

- o $a(i,j)$ Arrival Interface ID
- o $d(i,j)$ Departure Interface ID
- o $t(i,j)$ Arrival Timestamp (where $t(i,j)$ is ideally supplied by the hop, or approximated from the sending time of the packet that revealed the hop)
- o Measurements of Round Trip Delay (for each packet that reveals the same Host Identity and attributes, but not timestamp of course, see next section)

Now that Host Identities and related information can be positioned according to their distance from the host with address Src in hops, we introduce two forms of Routes:

A Route Ensemble is defined as the combination of all routes traversed by different flows from the host at Src address to the host at Dst address. The route traversed by each flow (with addresses Src and Dst, and other fields which constitute flow criteria) is a member of the ensemble and called a Member Route.

Using $h(i,j)$ and components and parameters, further define:

A Member Route is an ordered graph $\{h(1,j), \dots, h(N_j, j)\}$ in the context of a single flow, where $h(i-1, j)$ and $h(i, j)$ are by 1 hop away from each other and $N_j = \text{Dst}$ is the minimum TTL value needed by the packet on Member Route j to reach Dst. Member Routes must be unique. This uniqueness 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 ($i=1..N_j$).

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 hosts with Src and Dst addresses. More formally, with the host 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 various routes may overlap).

3.4. Related Round-Trip Delay and Loss Definitions

$RTD(i,j,T)$ is defined as a singleton of the [RFC2681] Round-trip Delay between the host with IP address = Src and the host 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 host with IP address = Src and the host at Hop $h(i,j)$ at time T .

3.5. Discussion

Depending on the way that Host Identity is revealed, it may be difficult to determine parallel subpaths between the same pair of hosts (i.e. multiple parallel links). It is easier to detect parallel subpaths involving different hosts.

- o If a pair of discovered hosts identify two different IP addresses, then they will appear to be different hosts.

- o If a pair of discovered hosts identify two different IP addresses, and the IP addresses resolve to the same host name (in the DNS), then they will appear to be the same hosts.
- o If a discovered host always replies using the same IP address, regardless of the interface a packet arrives on, then multiple parallel links cannot be detected at the IP layer.
- o If parallel links between routers are aggregated below the IP layer, In other words, all links share the same pair of IP addresses, then the existence of these parallel links can't be detected at IP layer.

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). If this topic is investigated further, there may be some 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.

When a route assessment employs packets at the IP layer (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).

@@@ Editor's Note: There is an opportunity to investigate and discuss the RFC 2330 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. Knowledge of "class C" parameters on a path potentially reduces the number of flows required for a given method.

3.6. Reporting the Metric

@@@ to be provided

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 hosts on a path, and Hybrid active-passive methods that involve direct interrogation of cooperating hosts (usually within a single domain). Description of these methods follow.

@@@ Editor's Note: We need to incorporate description of Type-P packets (with the flow parameters) used in each method below.

4.1. Active Methodologies

We have chosen to describe the method based on that employed in current open source tools, thereby providing a practical framework for further advanced techniques to be included as method variants. This method is applicable to use across multiple administrative domains.

Paris-traceroute [PT] provides some measure of protection from path variation generated by ECMP load balancing, and it ensures traceroute packets will follow the same path in 98% of cases according to [SCAMPER]. If it is necessary to find every path possible between two hosts, Paris-traceroute provides "exhaustive" mode while scamper provides "tracelb" (stands for traceroute load balance).

The Type-P of packets used could be ICMP (as ones in the original traceroute), UDP and TCP. The later are used when a particular characteristic is needed to verify, such as filtering or traffic shaping on specific ports (i.e., services).

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. Since route assessment can be conducted using TCP, UDP or ICMP packets, this method **REQUIRES** the Diffserv field, the protocol number, IP source and destination addresses, and the port settings for TCP or UDP kept constant. For ICMP probes, the method additionally **REQUIRES** the type, code, and ICMP checksum constant; which take the same position in the header of an IP packet, e.g., bytes 20 to 23 when the header IP has no options.

Maintaining a constant checksum in ICMP is most challenging because the ICMP Sequence Number is part of the calculation. The advanced traceroute method requires calculations using the IP Sequence Number Field and the Identifier Field, yielding a constant ICMP checksum in successive packets. For an example of calculations to maintain a constant checksum, see Appendix A of [RFC7820], where revision of a timestamp field is complemented by modifying the 2 octet checksum complement field (these fields take the roles of the ICMP Sequence Number Identifier Fields, respectively).

For TCP and UDP packets, the checksum must also be kept constant. Therefore, the first four bytes of UDP (or TCP) data field are modified to compensate for fields that change from packet to packet.

Note: other variants of advanced traceroute are planned be described.

Finally, the return path is also important to check. Taking into account that it is an ICMP time exceeded (during transit) packet, the source and destination IP are constant for every reply. Then, we should consider the fields in the first 32 bits of the protocol on the top of IP: the type and code of ICMP packet, and its checksum. Again, to maintain the ICMP checksum constant for the returning packets, we need to consider the whole ICMP message. It contains the IP header of the discarded packet plus the first 8 bytes of the IP payload; that is some of the fields of TCP header, the UDP header plus four data bytes, the ICMP header plus four bytes. Therefore, for UDP case the data field is used to maintain the ICMP checksum constant in the returning packet. For the ICMP case, the identifier and sequence fields of the sent ICMP probe are manipulated to be constant. The TCP case presents no problem because its first eight bytes will be the same for every packet probe.

Formally, to maintain the same flow in the measurements to a certain hop, the Type-P-Route-Ensemble-Method-Variant packets should be[PT]:

- o TCP case: Fields Src, Dst, port-Src, port-Dst, and Diffserv Field should be the same.
- o UDP case: Fields Src, Dst, port-Src, port-Dst, and Diffserv Field should be the same, the UDP-checksum should change to maintain constant the IP checksum of the ICMP time exceeded reply. Then, the data length should be fixed, and the data field is used to fixing it (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).
- o ICMP case: The Data field should compensate variations on TTL, IP identification, and IP checksum for every packet.

Then, the way to identify different hops and attempts of the same 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.

4.2. 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:

1. The ingress node adds one or more fields to the measurement packets, and identifies to other nodes in the domain that a route assessment will be conducted using one or more specific packets. The packets typically originate from a host outside the domain, and constitute normal traffic on the domain.
2. Each node visited by the specific packet within in the domain identifies itself in a data field of the packet (the field has been added for this purpose).
3. When a measurement packet reaches the edge node of the domain, the edge node adds its identity to the list, removes all the identities from the packet, forwards the packet onward, and communicates the ordered list of node identities to the intended receiver.

In addition to node identity, nodes may also identify the ingress and egress interfaces utilized by the tracing packet, the time of day when the packet was processed, and other generic data (as described in section 4 of [I-D.ietf-ippm-ioam-data]).

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 hosts involved in the path that qualify as Cooperating Hosts, but not as Discoverable Hosts, 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 hosts that are part of a domain supporting an interrogation protocol have the following attributes:

1. Hosts at the ingress to the domain SHOULD be both Discoverable and Cooperating, and SHOULD reveal the same Host Identity in response to both active and hybrid methods.
2. Any Hosts within the domain that are both Discoverable and Cooperating SHOULD reveal the same Host Identity in response to both active and hybrid methods.

3. Hosts at the egress to the domain SHOULD be both Discoverable and Cooperating, and SHOULD reveal the same Host Identity in response to both active and hybrid methods.

When Hosts 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-hosts 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] [bdrmap][IDCong]).

The first two categories correspond to the physical distance dependency on Round Trip Delay (RTD) while the last one binds RTD with queueing delay on routers. Due to the significant contribution of propagation delay in long distance hops, RTD will be at least 100ms on transatlantic hops, depending on the geolocation of the vantage points. Moreover, RTD is typically greater 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 after sending several traceroute probes, it is possible to detect congestion observing different statistics parameters (e.g., see [IDCong]).

6. Tools to Measure Delays in the Internet

Internet routing is complex because it depends on the policies of thousands Autonomous Systems (AS). While most of the routers perform load balancing on flows using Equal Cost Multiple Path (ECMP), a few still divide the workload through packet-based techniques. The former scenario is defined according to [RFC2991] while the latter generates a round-robin scheme to deliver every new outgoing packet. ECMP keeps flow state in the router to ensure every packet of a flow is delivered by the same path, and this avoids increasing the packet delay variation and possibly producing overwhelming packet reordering in TCP flows.

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 well-known tool traceroute was developed to gather the IP addresses of each hop along a path using the ICMP protocol [RFC0792]. Besides, traceroute adds the measured RTD from each hop. However, the growing complexity of the Internet makes it more challenging to develop accurate traceroute implementation. For instance, the early traceroute tools would be inaccurate in the current network, mainly because they were not designed to retain 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.

Paris-traceroute-like tools operate in the following way: every packet should follow the same path because the sensitive fields of the header are controlled to appear as the same flow. This means that source and destination IP addresses, source and destination port numbers are the same in every packet. Additionally, Differentiated Services Code Point (DSCP), checksum and ICMP code should remain constant since they may affect the path selection.

Today's traceroute tools can send either UDP, TCP or ICMP packet probes. Since ICMP header does not include transport layer information, there are no fields for source and destination port numbers. For this reason, these tools keep constant ICMP type, code, and checksum fields to generate a kind of flow. However, the checksum may vary in every packet, therefore when probes use ICMP packets, ICMP Identifier and Sequence Number are manipulated to maintain constant checksum in every packet. On the other hand, when UDP probes are generated, the expected variation in the checksum of each packet is again compensated by manipulating the payload.

Paris-traceroute allows its users to measure RTD in every hop 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 this method). This task is accomplished by repeating complete traceroute measurements with different flow parameters for each measurement. 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.

7. 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 generate 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 to compute quartile values "on the fly" using the algorithm presented in [P2].

This procedure allow 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 in performing traceroutes several times to obtain samples of the RTD in every hop

from a path, during a time window (W) and compute the quantiles for every hop. This could be done for a single path flow or for every detected path flow.

Even though a particular hop may be understood as the amount of hops away from the source, a more detailed classification could be used. For example, a possible classification may be identify ICMP Time Exceeded packets coming from the same routers to those who have the same hop distance, IP address of the router which is replying and TTL value of the received ICMP packet.

Thus, the proposed methodology is based on this algorithm:

```
=====
1  input:   W (window time of the measurement)
2           i_t (time between two measurements)
3           E (True: exhaustive, False: a single path)
4           Dst (destination IP address)
5  output:  Qs (quantiles for every hop and alt in the path(s) to Dst)
-----
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)
=====
```

In line 9 the advance-traceroute could be either Paris-traceroute or Scamper, which will use "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 in the path towards the Dst. Additionally, it could also return path variations using "alt" variable.

8. Conclusions

Combining the method proposed in Section 4 and statistics in Section 7, we can measure the performance of paths interconnecting two endpoints in Internet, and attempt the categorization of link types and congestion presence based on RTD.

9. 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).

@@@ add reference to security considerations from [I-D.ietf-ippm-ioam-data].

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.

10. IANA Considerations

This memo makes no requests of IANA.

11. Acknowledgements

The authors acknowledge Ruediger Geib, for his penetrating comments on the initial draft. 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. We thank them all!

12. References

12.1. Normative References

[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-00 (work in progress), September 2017.

- [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>>.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <<https://www.rfc-editor.org/info/rfc2460>>.
- [RFC2675] Borman, D., Deering, S., and R. Hinden, "IPv6 Jumbograms", RFC 2675, DOI 10.17487/RFC2675, August 1999, <<https://www.rfc-editor.org/info/rfc2675>>.
- [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>>.
- [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>>.
- [RFC4494] Song, JH., Poovendran, R., and J. Lee, "The AES-CMAC-96 Algorithm and Its Use with IPsec", RFC 4494, DOI 10.17487/RFC4494, June 2006, <<https://www.rfc-editor.org/info/rfc4494>>.

- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <<https://www.rfc-editor.org/info/rfc6282>>.
- [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>>.
- [RFC6564] Krishnan, S., Woodyatt, J., Kline, E., Hoagland, J., and M. Bhatia, "A Uniform Format for IPv6 Extension Headers", RFC 6564, DOI 10.17487/RFC6564, April 2012, <<https://www.rfc-editor.org/info/rfc6564>>.
- [RFC6673] Morton, A., "Round-Trip Packet Loss Metrics", RFC 6673, DOI 10.17487/RFC6673, August 2012, <<https://www.rfc-editor.org/info/rfc6673>>.
- [RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing of IPv6 Extension Headers", RFC 7045, DOI 10.17487/RFC7045, December 2013, <<https://www.rfc-editor.org/info/rfc7045>>.
- [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>>.

- [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>>.
- [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>>.

12.2. 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.
- [I-D.brockners-inband-oam-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-brockners-inband-oam-data-07 (work in progress), July 2017.
- [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.
- [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 P² algorithm for dynamic calculation of quantiles 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.
- [RFC7594] Eardley, P., Morton, A., Bagnulo, M., Burbidge, 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>>.
- [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

Jose 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: acmorton@att.com
URI: <http://home.comcast.net/~acmacm/>

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/>

IPPM Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 26, 2017

B. M Gaonkar
S. Jacob
Juniper
G. Fioccola
Telecom Italia
Q. Wu
Huawei
P. Ananthasankaran
Nokia
June 24, 2017

Performance Measurement Models
draft-bhaprasud-ippm-pm-03

Abstract

This document defines the performance measurement models for service level packets on the network which can be implemented in different kind of network scenarios. Based on the performance matrix, the analytics data can be pulled from a live network which is not possible at present. This can be used for self evolving networks.

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 December 26, 2017.

Copyright Notice

Copyright (c) 2017 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 used in this document	3
3. Traffic Management Architecture	5
3.1. Selection Process	5
3.2. Metering Process	6
4. Performance Measurement Models	6
4.1. Complete data measurement (Monitoring all the traffic) .	6
4.2. Color based data measurement	7
4.3. CoS based Data measurement	7
4.4. CoS and Color based Data measurement	8
5. Active and Passive performance measurements	8
6. Use Cases	8
7. Acknowledgements	9
8. Security Considerations	9
9. References	10
9.1. Normative References	10
9.2. Informative References	10
Authors' Addresses	10

1. Introduction

Today performance monitoring or tracking of the performance experienced by customer traffic is a key technology to strengthen service offering and verify service level agreement between customers and service providers, perform troubleshooting. The lack of adequate monitoring tools to detect an interesting subset of a packet stream, as identified by a particular packet attribute(e.g., commit rate or DSCP) and measure that packet loss drives an effort to design a new method for the performance monitoring of live traffic, possibly easy to implement and deploy. The draft aims to provide fine granularity loss, delay and delay variation measurement and define a performance measurement model on customer traffic based on a set of constraints that are associated with service level agreement such as cos attribute, color attribute. Each customer traffic is corresponding to an interesting subset of the same packet stream. The customer or a interesting packet stream can be identified by a list of source or destination prefixes, or by ingress or egress interfaces, combining with packet attributes such as DSCP or commit rate).Unlike Color and COS identification specified in MEF 23.1, this draft doesn't define

new Color and CoS identification mechanism, instead, it stick to color definition in [RFC2697] and [RFC2698] and COS definition in [RFC2474].

The network would be provisioned with multiple services(e.g., real time service, interactive service) having different network performance criteria(e.g., bandwidth constraint or packet loss constraint for the end to end path) based on the customers' requirement. This models aims at performing Loss, Delay and delay variation measurement for these services (belonging to the same customer)independently for each defined network performance criteria.

The class-of-service and packet color classification defined in the network is a key factor to classify network traffic and drive traffic management mechanism to achieve corresponding network performance criteria for each service. This draft uses the class-of-service model and color based model for any given network to define the performance measurement for various services with the different network performance criteria requirements.

The proposed models is suitable mainly for passive performance measurements but can be considered for active and hybrid performance measurements as well.

This solution models loss, delay an delay variation measurement in different kinds of network scenarios. The different models explained here will help to analyse performance pattern, analyze the network congestion in a better way and model the network in a better way. For instance, Loss measurement is carried out between 2 end points. The underlying technology could be an active loss measurement or a passive loss measurement.

Any loss measurement will require 2 counters:

- o Number of packets transmitted from one end point.
- o Number of packets received at the other end point.

This draft explains the different ways to model the above data and get meaningful result for the loss, delay and delay variation measurement. The underlying technology could be an MPLS performance measurement, or an IP based performance measurement.

2. Conventions used in this document

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 [RFC2119].

Observation Point An Observation Point is a location in the network where data packets can be observed. Examples include a line to which a probe is attached, a shared medium, such as an Ethernet-based LAN, a single port of a router, or a set of interfaces (physical or logical) of a router.

Persistence Data Store The persistence Data store is a scalable data store which collects time based data such as streaming data or time series data for network analytics.

Time Series Data Time Series Data is a sequence of data points with time stamps. The data points are limited to loss, delay and delay variation measurement results in this document.

Packet Stream A Packet Stream denotes a set of packets from the Observed Packet Stream that flows past some specified point within the Metering Process. An example of a Packet Stream is the output of the Selection Process.

Packet Content The Packet Content denotes the union of the packet header (which includes link layer, network layer, and other encapsulation headers) and the packet payload.

Color Identifier: It is used to identify the color that applies to the data packet. Color identifier can be assigned to service level packet based on commit rate and excess rate set for the traffic. For example, the service level packet will be set with "green" color if it is less than committed" rate; the Service Level packet will be set with "yellow" color if it is exceeding the "committed" rate but less than the "excess" rate. The service frame will be set with "red" color if it is exceeding both the "committed" and "excess" rates.

CoS Identifier: It is used to identify the CoS that applies to the data packet. CoS identifier can be assigned based on dot1p value in C-tag, or DSCP in IP header.

Complete data measurement: Complete data measurement is a data measurement method which monitors every packet and condense a large amount of information about packet arrivals into a small number of statistics. The aim of "monitoring every packet" is to ensure that the information reported is not dependent on the application.

Color based data measurement: Color based data measurement is a data measurement method which monitors the data packet with the same color identifier. Color identifier could be "green" color, "yellow" color and "red" color.

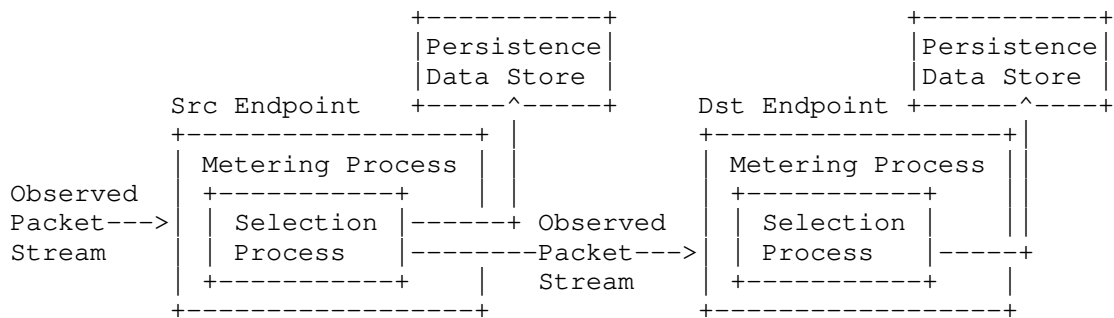
CoS based data measurement: Color based data measurement is a data measurement method which monitors the data packet with the same CoS identifier. COS identifier could be C-Tag Priority Code Point (PCP) or DSCP.

CoS and Color based Data measurement: CoS and Color based Data measurement is a data measurement method which monitors the data packet with the specific CoS Identifier and Specific Color Identifier as constraints. The measurement results with CoS Identifier and Color Identifier constraints constitute a Network Performance matrix.

3. Traffic Management Architecture

A stream of packets is observed at an Observation Point of the source endpoint and destination endpoints. Two observation points can also be placed at the same endpoint for node monitoring [I-D.ietf-ippm-alt-mark], i.e., one is at ingress interface of the endpoint and the other is at the egress interface of the endpoint. A Selection Process inspects each packet to determine whether or not it is to be selected for data analytics. The Selection Process is part of the Metering Process, which constructs a report stream on selected packets as output, using the Packet Content, and possibly other information such as the arrival timestamp. The report stream on selected packets will be stored in the persistence data store for real time data analysis or time sequence data analysis.

The following figure indicates the sequence of the three processes (Selection, Metering, and Storing).



3.1. Selection Process

This section defines the Selection Process and related objects.

Selection Process: A Selection Process takes the Observed Packet Stream as its input and selects a subset of that stream as its output.

Selection State: A Selection Process may maintain state information for use by the Selection Process. At a given time, the Selection State may depend on packets observed at and before that time, and other variables. Examples include sequence numbers of packets at the input of Selectors, a timestamp of observation of the packet at the Observation Point, indicators of whether the packet was selected by a given Selector.

Selector: A Selector defines the action of a Selection Process on a single packet of its input. If selected, the packet becomes an element of the output Packet Stream.

The Selector can make use of the following information in determining whether a packet is selected:

- * COS Identifier in the Packet Content;
- * Traffic attribute such as Color identifier;
- * Combination of CoS Identifier and Color Identifier

3.2. Metering Process

A Metering Process selects packets from the Observed Packet Stream using a Selection Process, and produces as output a Report Stream concerning the selected packets.

4. Performance Measurement Models

4.1. Complete data measurement (Monitoring all the traffic)

This model uses the complete data traffic between the 2 end-points to compute loss measurement, delay and delay variation. This will result in computation of loss, delay and delay variation measurement for the entire traffic in the network in one direction. This is primarily used in cases of backbone traffic where traffic from different services are aggregated and send into the core network. This will count all the packet, this gives the overall measurment between one endpoint to other.

4.2. Color based data measurement

This is same as the above section of "complete data measurement" with a minor difference, only monitoring the data packet with specific color identifier.

In this model the packets are counted in the following Way: Count specific data traffic with different color identifier between 2 end points for loss, delay and delay variation measurement. One example of Color based data measurement is to count two type of color based traffic:

- o Count all committed traffic between the 2 end-point for loss measurement.
- o Count all Excess traffic which is beyond the committed traffic for the specific network.
- o The probe carries the time stamps, which can later be used for calculating the service outage.
- o This method can be used for mapping the overall customer traffic along with EIR, based on the EIR provider can increase the bandwidth and charge him.

When both of these are combined then it becomes the model for complete traffic as mentioned in the above section.

In practice the Color of traffic can use any mechanism based on the network encapsulation. As long as the packets could be treated differently based on the underlying encapsulation this mechanism could be used.

This can be used for measuring the whole traffic of the customer who don't want cos level measurement. Ideally this can be used for provider who extend bandwidth for small providers, point to point services etc.

4.3. CoS based Data measurement

This model uses the data traffic in the network which is flowing in a specific CoS to measure the loss, delay and delay variation in the network. Based on the class of traffic in the network the transmitted and received packets are counted to calculate the packets transferred per service level. The time stamp will be captured along with the packet count to measure the service down time. This model measures the performance per service level. This data can be stored on the routers which can be used to plot the live analytics.

Primary use of this kind of measurement is to measure packet loss delay and delay variation for a specific service which needs to meet network performance requirements. The service could be a point-to-point layer2 service, an MPLS based service.

4.4. CoS and Color based Data measurement

This model uses a combination of both Color based data measurement and CoS based data measurement. Packets are counted for a specific CoS with a specific Color. This can count both in profile packet which are green and yellow which are out profile packets. This will not count the red packet which doesn't meet network performance requirements. The packets will be counted per service level with CIR and EIR along with time stamps to find the service outage and loss. The per service level counting for CoS and color will give more granular level data for plotting service graph and if some service is continuously exceeding the bandwidth this data can be used for charging the end customer for extra bandwidth usage or increase the bandwidth based on usage basis.

5. Active and Passive performance measurements

This model reinforces the use of well known methodologies for passive performance measurements. A very simple, flexible and straightforward mechanism is presented in [I-D.ietf-ippm-alt-mark]. The basic idea is to virtually split traffic flows into consecutive batches of packets: each block represents a measurable entity unambiguously recognizable thanks to the alternate marking. This approach, called Alternate Marking method, is efficient both for passive performance monitoring and for active performance monitoring. Most of the applications requires passive packet loss measurement for a better accuracy. Instead, in some cases, it is desirable to have only active delay measurements (e.g TWAMP or OWAMP), because it is enough.

6. Use Cases

Consider a provider running point to point service between router A and B for his customer "X". Customer "X" has voice traffic which requires special treatment, then he requires attention for database traffic. The customer "X" has SLA with the provider. Now the challenge faced by the provider is how to measure the traffic of customer "X" for each class and calculate the bandwidth, moreover the provider has to see whether the "X" is sending traffic which is exceeding the level so that he can make tariff accordingly. This problem is solved by the above models which can measure the packet for each class of traffic and tabulate the data. Later point of time this data can be pulled for evaluation.

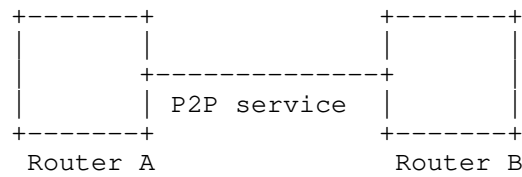


Figure 1: P2P

The same considerations can be applicable in a multipoint to multipoint scenario (e.g. VPN or Data Center interconnections). In this case Customer "X" has multiple ingress endpoints and multiple egress endpoints. The proposed matrix model is composed by the number of flows of "X" in the multipoint scenario and by class-of-service and color classification. So the SLA matrix is a reference for the analysis and evaluation phase.

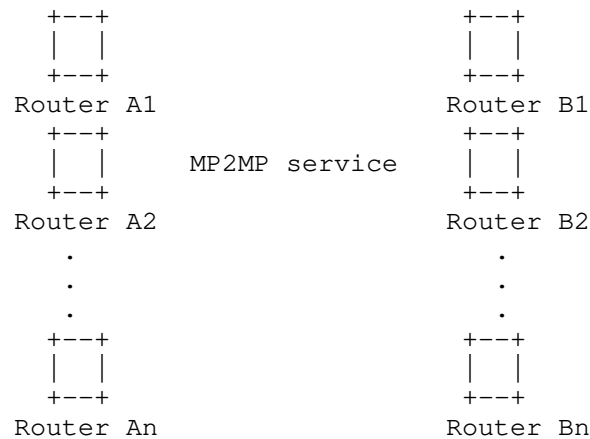


Figure 2: MP2MP

7. Acknowledgements

We would like to thank Brian Trammell for giving us the opportunity to present our draft. We would like to thank Greg Mirsky for the comments.

8. Security Considerations

This document does not introduce security issues beyond those discussed in [I-D.ietf-ippm-alt-mark].

9. References

9.1. Normative References

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

9.2. Informative References

[I-D.ietf-ippm-alt-mark]
Fioccola, G., Capello, A., Cociglio, M., Castaldelli, L.,
Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi,
"Alternate Marking method for passive performance
monitoring", draft-ietf-ippm-alt-mark-04 (work in
progress), March 2017.

Authors' Addresses

Bharat M Gaonkar
Juniper Networks
1133 Innovation Way
Sunnyvale, California 94089
USA

Email: gbharat@juniper.net

Sudhin Jacob
Juniper Networks
1133 Innovation Way
Sunnyvale, California 94089
USA

Email: gbharat@juniper.net

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

Email: giuseppe.fioccola@telecomitalia.it

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: bill.wu@huawei.com

Praveen Ananthasankaran
Nokia
Manyata Embassy Tech Park, Silver Oak (Wing A),
Outer Ring Road, Nagawara
Bangalore 560045
India

Email: praveen.ananthasankaran@nokia.com

ippm
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2018

F. Brockners
S. Bhandari
C. Pignataro
Cisco
H. Gredler
RtBrick Inc.
J. Leddy
Comcast
S. Youell
JPMC
T. Mizrahi
Marvell
D. Mozes
Mellanox Technologies Ltd.
P. Lapukhov
Facebook
R. Chang
Barefoot Networks
D. Bernier
Bell Canada
July 2, 2017

Data Fields for In-situ OAM
draft-brockners-inband-oam-data-07

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 the data fields and associated data types for in-situ OAM. In-situ OAM data fields can be embedded into a variety of transports such as NSH, Segment Routing, Geneve, native IPv6 (via extension header), or IPv4. 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 <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 January 3, 2018.

Copyright Notice

Copyright (c) 2017 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	3
3. Scope, Applicability, and Assumptions	4
4. IOAM Data Types and Formats	5
4.1. IOAM Tracing Options	6
4.1.1. Pre-allocated Trace Option	8
4.1.2. Incremental Trace Option	11
4.1.3. IOAM node data fields and associated formats	14
4.1.4. Examples of IOAM node data	19
4.2. IOAM Proof of Transit Option	21
4.3. IOAM Edge-to-Edge Option	23
5. IOAM Data Export	23
6. IANA Considerations	24
6.1. Creation of a New In-Situ OAM (IOAM) Protocol Parameters IANA registry	24
6.2. IOAM Trace Type Registry	24
6.3. IOAM Trace Flags Registry	24
6.4. IOAM POT Type Registry	25
6.5. IOAM E2E Type Registry	25
7. Manageability Considerations	25
8. Security Considerations	25
9. Acknowledgements	25
10. References	25
10.1. Normative References	25

10.2. Informative References	26
Authors' Addresses	27

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 is being sent within packets specifically dedicated to OAM. A discussion of the motivation and requirements for in-situ OAM can be found in [I-D.brockners-inband-oam-requirements]. IOAM is to complement mechanisms such as Ping or Traceroute, or more recent active probing mechanisms as described in [I-D.lapukhov-dataplane-probe]. In terms of "active" or "passive" OAM, "in-situ" OAM can be considered a hybrid OAM type. While no extra packets are sent, IOAM adds information to the packets therefore cannot be considered passive. In terms of the classification given in [RFC7799] IOAM could be portrayed as Hybrid Type 1. "In-situ" mechanisms do not require extra packets to be sent and hence don't change the packet traffic mix within the network. 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 live 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.

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
Geneve:	Generic Network Virtualization Encapsulation [I-D.ietf-nvo3-geneve]
IOAM:	In-situ Operations, Administration, and Maintenance
MTU:	Maximum Transmit Unit
NSH:	Network Service Header [I-D.ietf-sfc-nsh]

OAM: Operations, Administration, and Maintenance

POT: Proof of Transit

SFC: Service Function Chain

SID: Segment Identifier

SR: Segment Routing

VXLAN-GPE: Virtual eXtensible Local Area Network, Generic Protocol Extension [I-D.ietf-nvo3-vxlan-gpe]

3. Scope, Applicability, and Assumptions

IOAM deployment assumes a set of constraints, requirements, and guiding principles which are described in this section.

Scope: This document defines the data fields and associated data types for in-situ OAM. The in-situ OAM data field can be transported by a variety of transport protocols, including NSH, Segment Routing, Geneve, IPv6, or IPv4. Specification details for these different transport protocols are outside the scope of this document.

Deployment domain (or scope) of in-situ OAM deployment: IOAM is a network domain focused feature, with "network domain" being a set of network devices or entities within a single administration. For example, a network domain can include an enterprise campus using physical connections between devices or an overlay network using virtual connections / tunnels for connectivity between said devices. A network domain is defined by its perimeter or edge. Designers of carrier protocols for IOAM must 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, e.g. using for example packet filtering methods. The operator should consider 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 a native IPv6 transport, IOAM support for ICMPv6 Echo Request/Reply could be 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 live user traffic by the devices which form the edge of a domain.

Devices within an IOAM domain can update and/or add IOAM data-fields. Domain edge devices 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 live user traffic. It SHOULD be possible to enable 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.) The selected set of traffic can also be all traffic.

Encapsulation independence: Data formats for IOAM SHOULD be defined in a transport-independent manner. IOAM applies to a variety of encapsulating protocols. A definition of how IOAM data fields are carried by different transport protocols is outside the scope of this document.

Layering: If several encapsulation protocols (e.g., in case of tunneling) are stacked on top of each other, IOAM data-records could be present at every layer. The behavior follows the ships-in-the-night model.

Combination with active OAM mechanisms: IOAM should be usable for active network probing, enabling for example a customized version of traceroute. Decapsulating IOAM nodes may have an ability to send the IOAM information retrieved from the packet back to the source address of the packet or to the encapsulating node.

IOAM implementation: The IOAM data-field definitions take the specifics of devices with hardware data-plane and software data-plane into account.

4. IOAM Data Types and Formats

This section defines IOAM data types and data fields and associated data types required for IOAM. The different uses of IOAM require the definition of different types of data. The IOAM data fields for the data being carried corresponds to the three main categories of IOAM data defined in [I-D.brockners-inband-oam-requirements], which are: edge-to-edge, per node, and for selected nodes only.

Transport options for IOAM data are outside the scope of this memo, and are discussed in [I-D.brockners-inband-oam-transport]. IOAM data fields are fixed length data fields. A bit field determines the set of OAM data fields embedded in a packet. Depending on the type of the encapsulation, a counter field indicates how many data fields are included in a particular packet.

IOAM is expected to be deployed in a specific domain rather than on the overall Internet. The part of the network which employs IOAM is referred to as the "IOAM-domain". IOAM data is added to a packet upon entering the IOAM-domain and is removed from the packet when exiting the domain. Within the IOAM-domain, the IOAM data may be updated by network nodes that the packet traverses. The device which adds an IOAM data container to the packet to capture IOAM data is called the "IOAM encapsulating node", whereas the device which removes the IOAM data container is referred to as the "IOAM decapsulating node". Nodes within the domain which are aware of IOAM data and read and/or write or process the IOAM data are called "IOAM transit nodes". IOAM nodes which add or remove the IOAM data container can also update the IOAM data fields at the same time. Or in other words, IOAM encapsulation 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 Segment Routing deployment might require the segment routing path to be verified. In that case, only the SR nodes would also be IOAM transit nodes rather than all nodes.

4.1. IOAM Tracing Options

"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, i.e., in a typical deployment all nodes in an in-situ OAM-domain would participate in IOAM and thus be IOAM transit nodes, IOAM encapsulating or IOAM decapsulating nodes. If not all nodes within a domain are IOAM capable, IOAM tracing information will only be collected on those nodes which are IOAM capable. Nodes which are not IOAM capable 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.

To optimize hardware and software implementations tracing is defined as two separate options. Any deployment MAY choose to configure and support one or both of the following options. An implementation of the transport protocol that carries these in-situ OAM data MAY choose to support only one of the options. In the event that both options are utilized at the same time, the Incremental Trace Option MUST be placed before the Pre-allocated Trace Option. Given that the operator knows which equipment is deployed in a particular IOAM, the operator will decide by means of configuration which type(s) of trace options will be enabled for a particular domain.

Pre-allocated Trace Option: This trace option is defined as a container of node data fields with pre-allocated space for each node to populate its information. This option is useful for

software implementations where it is efficient to allocate the space once and index into the array to populate the data during transit. The IOAM encapsulating node allocates the option header and sets the fields in the option header. The in situ OAM 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. A pointer which is part of the IOAM trace data points to the next empty slot in the array, which is where the next IOAM transit node fills in its data.

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. The maximum length of the node data list is written into the option header. This type of trace recording is useful for some of the hardware implementations as this 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 in-situ OAM encapsulating node allocates the option header. The in-situ OAM encapsulating node based on operational state and configuration sets the fields in the header to control how large the node data list can grow. IOAM transit nodes push their node data to the node data list and increment the number of node data fields in the header.

Every node data entry is to hold information for a particular IOAM transit node that is traversed by a packet. The in-situ OAM decapsulating node removes the IOAM data and processes and/or exports the metadata. IOAM data uses its own name-space for information such as node identifier or interface identifier. This allows for a domain-specific definition and interpretation. For example: In one case an interface-id could 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).

The following IOAM data is defined for IOAM tracing:

- 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. Different definitions of processing time are feasible and expected, though it is important that all devices of an in-situ OAM 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 deployment, all IOAM nodes should 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 A mechanism to detect whether IOAM trace data was added at every hop or whether certain hops in the domain weren't in-situ OAM transit nodes.

The "node data list" array 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.

4.1.1. Pre-allocated Trace Option

In-situ OAM pre-allocated trace option:

Pre-allocated trace 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
+-----+-----+-----+-----+-----+-----+-----+-----+
|          IOAM-Trace-Type          |NodeLen|  Flags  | Octets-left |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Pre-allocated Trace Option Data MUST be 4-octet aligned:

```

+-----+-----+-----+-----+-----+-----+-----+-----+<--+
|                                     |                                     |
|                               node data list [0]                               |
|                                     |                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+  D
|                               node data list [1]                               |  a
|                                     |                                     |  t
+-----+-----+-----+-----+-----+-----+-----+-----+  a
|                               ...                               |
+-----+-----+-----+-----+-----+-----+-----+-----+  S
|                               node data list [n-1]                               |  p
|                                     |                                     |  a
+-----+-----+-----+-----+-----+-----+-----+-----+  c
|                               node data list [n]                               |  e
|                                     |                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+<--+

```

IOAM-Trace-Type: A 16-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 bit fields are defined in this document, with details on each field described in the Section 4.1.3. 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 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 nanoseconds in the node data.
- Bit 4 When set indicates presence of transit delay in the node data.
- Bit 5 When set indicates presence of app_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 variable length Opaque State Snapshot field.
- 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 app_data wide in the node data.
- Bit 11 When set indicates presence of the Checksum Complement node data.
- Bit 12-15 Undefined in this draft.

Section 4.1.3 describes the IOAM data types and their formats. Within an in-situ OAM domain possible combinations of these bits making the IOAM-Trace-Type can be restricted by configuration knobs.

Node Data Length: 4-bit unsigned integer. This field specifies the length of data added by each node in multiples of 4-octets. For example, if 3 IOAM-Trace-Type bits are set and none of them is wide, then the Node Data Length would be 3. If 3 IOAM-Trace-Type bits are set and 2 of them are wide, then the Node Data Length would be 5.

Flags 5-bit field. Following flags are defined:

- Bit 0 "Overflow" (O-bit) (most significant bit). This bit is set by the network element if there is not enough number of octets

left to record node data, no field is added and the overflow "O-bit" must be set to "1" in the header. This is useful for transit nodes to ignore further processing of the option.

Bit 1 "Loopback" (L-bit). Loopback mode is used to send a copy of a packet back towards the source. Loopback mode assumes that a return path from transit nodes and destination nodes towards the source exists. The encapsulating node decides (e.g. using a filter) which packets loopback mode is enabled for by setting the loopback bit. The encapsulating node also needs to ensure that sufficient space is available in the IOAM header for loopback operation. The loopback bit when set indicates to the transit nodes processing this option to create a copy of the packet received and send this copy of the packet back to the source of the packet while it continues to forward the original packet towards the destination. The source address of the original packet is used as destination address in the copied packet. The address of the node performing the copy operation is used as the source address. The L-bit MUST be cleared in the copy of the packet a nodes sends it back towards the source. On its way back towards the source, the packet is processed like a regular packet with IOAM information. Once the return packet reaches the IOAM domain boundary IOAM decapsulation occurs as with any other packet containing IOAM information.

Bit 2-4 Reserved: Must be zero.

Octets-left: 7-bit unsigned integer. It is the data space in multiples of 4-octets remaining for recording the node data. This is used as an offset in data space to record the node data element.

Node data List [n]: Variable-length field. The type of which is determined by the IOAM-Trace-Type representing the n-th node data in the node data list. The node data list is encoded starting from the last node data of the path. The first element of the node data list (node data list [0]) contains the last node of the path while the last node data of the node data list (node data list[n]) contains the first node data of the path traced. The index contained in "Octets-left" identifies the offset for current active node data to be populated.

4.1.2. Incremental Trace Option

In-situ OAM incremental trace option:

In-situ OAM incremental trace 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	2	3	4	5	6	7	8	9																								
IOAM-Trace-Type																NodeLen																Flags																Max Length															

IOAM Incremental Trace Option Data MUST be 4-octet aligned:

node data list [0]																																																															
node data list [1]																																																															
...																																																															
node data list [n-1]																																																															
node data list [n]																																																															

IOAM-trace-type: A 16-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 bit fields are defined in this document, with details on each field described in the Section 4.1.3. 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 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 nanoseconds in the node data.
- Bit 4 When set indicates presence of transit delay in the node data.
- Bit 5 When set indicates presence of app_data in the node data.
- Bit 6 When set indicates presence of queue depth in the node data.
- Bit 7 When set indicates presence of variable length Opaque State Snapshot field.
- Bit 8 When set indicates presence of Hop_Lim and node_id wide 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 app_data wide in the node data.
- Bit 11 When set indicates presence of the Checksum Complement node data.
- Bit 12-15 Undefined in this draft.

Section 4.1.3 describes the IOAM data types and their formats.

Node Data Length: 4-bit unsigned integer. This field specifies the length of data added by each node in multiples of 4-octets. For example, if 3 IOAM-Trace-Type bits are set and none of them is wide, then the Node Data Length would be 3. If 3 IOAM-Trace-Type bits are set and 2 of them are wide, then the Node Data Length would be 5.

Flags 5-bit field. Following flags are defined:

- Bit 0 "Overflow" (O-bit) (least significant bit). This bit is set by the network element if there is not enough number of octets left to record node data, no field is added and the overflow "O-bit" must be set to "1" in the header. This is useful for transit nodes to ignore further processing of the option.

Bit 1 "Loopback" (L-bit). This bit when set indicates to the transit nodes processing this option to send a copy of the packet back to the source of the packet while it continues to forward the original packet towards the destination. The L-bit MUST be cleared in the copy of the packet before sending it.

Bit 2-4 Reserved. Must be zero.

Maximum Length: 7-bit unsigned integer. This field specifies the maximum length of the node data list in multiples of 4-octets. Given that the sender knows the minimum path MTU, the sender can set the maximum length according to the number of node data bytes allowed before exceeding the MTU. Thus, a simple comparison between "Opt data Len" and "Max Length" allows to decide whether or not data could be added.

Node data List [n]: Variable-length field. The type of which is determined by the OAM Type representing the n-th node data in the node data list. The node data list is encoded starting from the last node data of the path. The first element of the node data list (node data list [0]) contains the last node of the path while the last node data of the node data list (node data list[n]) contains the first node data of the path traced.

4.1.3. IOAM node data fields and associated formats

All the data fields MUST be 4-octet aligned. The IOAM encapsulating node MUST initialize data fields that it adds to the packet to zero. 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 left unaltered except when explicitly specified in the field description below. In the description of data below if zero is valid value then a non-zero value to mean not populated is specified.

Data field and associated data type for each of the data field is shown below:

Hop_Lim and node_id: 4-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 | |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Hop_Lim: 1-octet unsigned integer. It is set to the Hop Limit value in the packet at 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 over, and therefore its specific semantics are outside the scope of this memo.

node_id: 3-octet unsigned integer. Node identifier field to uniquely identify a node within in-situ OAM domain. The procedure to allocate, manage and map the node_ids is beyond the scope of this document.

ingress_if_id and egress_if_id: 4-octet field defined as follows:
When this field is part of the data field but a node populating the field is not able to fill it, the 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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           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.

timestamp seconds: 4-octet unsigned integer. Absolute timestamp in seconds that specifies the time at which the packet was received by the node. The structure of this field is identical to the most significant 32 bits of the 64 least significant bits of the [IEEE1588v2] timestamp. This truncated field consists of a 32-bit seconds field. As defined in [IEEE1588v2], the timestamp specifies the number of seconds elapsed since 1 January 1970 00:00:00 according to the International Atomic Time (TAI).

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           timestamp seconds           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

timestamp nanoseconds: 4-octet unsigned integer in the range 0 to 10^9-1 . This timestamp specifies the fractional part of the wall clock time at which the packet was received by the node in units of nanoseconds. This field is identical to the 32 least significant bits of the [IEEE1588v2] timestamp. This fields

allows for delay computation between any two nodes in the network when the nodes are time synchronized. 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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     timestamp nanoseconds                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

transit delay: 4-octet unsigned integer in the range 0 to $2^{30}-1$.

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 $2^{30}-1$ nanoseconds then the top bit 'O' is set to indicate overflow. 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                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

app_data: 4-octet placeholder which can be used by the node to add application specific data. App_data represents a "free-format" 4-octet bit field with its semantics defined by a specific deployment.

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     app_data                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

queue depth: 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 number of memory buffers used by the queue (a packet may consume one or more memory buffers, depending on its size). 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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|               queue depth               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Opaque State Snapshot: Variable length field. It allows the network element to store an arbitrary state in the node data field, without a pre-defined schema. 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. The 24-bit "Schema Id" field in the field indicates which particular schema is used, and should be configured on the network element by the operator.

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Length   |           Schema ID           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|
|               Opaque data
|
~
.
.
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Length: 1-octet unsigned integer. It is the length in octets of the Opaque data field that follows Schema Id. It MUST always be a multiple of 4.

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.

Hop_Lim and node_id wide: 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. It is set to the Hop Limit value in the packet at 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 for e.g. TTL value in IPv4 header or hop limit field from IPv6 header of the packet. The semantics of the Hop_Lim field depend on the lower layer protocol that IOAM is encapsulated over, and therefore its specific semantics are outside the scope of this memo.

node_id: 7-octet unsigned integer. Node identifier field to uniquely identify a node within in-situ OAM domain. The procedure to allocate, manage and map the node_ids is beyond the scope of this document.

ingress_if_id and egress_if_id wide: 8-octet field defined as follows: 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 0xFFFFFFFFFFFFFFFF 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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     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.

app_data wide: 8-octet placeholder which can be used by the node to add application specific data. App data represents a "free-format" 8-octet bit field with its semantics defined by a specific deployment.

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     app data                                     ~
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
~                                     app data (contd)                             |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Checksum Complement: 4-octet node data which contains a two-octet Checksum Complement field, and a 2-octet reserved field. The Checksum Complement can be used when IOAM is transported over encapsulations that make use of a UDP transport, such as VXLAN-GPE

or Geneve. In this case, incorporating the IOAM node data requires the UDP Checksum field to be updated. Rather than to recompute the Checksum field, a node can 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. 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          |          Reserved          |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

4.1.4. Examples of IOAM node data

An entry in the "node data list" array can have different formats, following the needs of the deployment. Some deployments might only be interested in recording the node identifiers, whereas others might be interested in recording node identifier and timestamp. The section defines different types that an entry in "node data list" can take.

0x002B: IOAM-Trace-Type is 0x2B then the format of node data is:

```

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          |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ingress_if_id |          egress_if_id          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          timestamp nanoseconds          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          app_data          |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

0x0003: IOAM-Trace-Type is 0x0003 then the format is:

```

 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 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| ingress_if_id | egress_if_id |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

0x0009: IOAM-Trace-Type is 0x0009 then the format is:

```

 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 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| timestamp nanoseconds |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

0x0021: IOAM-Trace-Type is 0x0021 then the format is:

```

 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 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| app_data |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

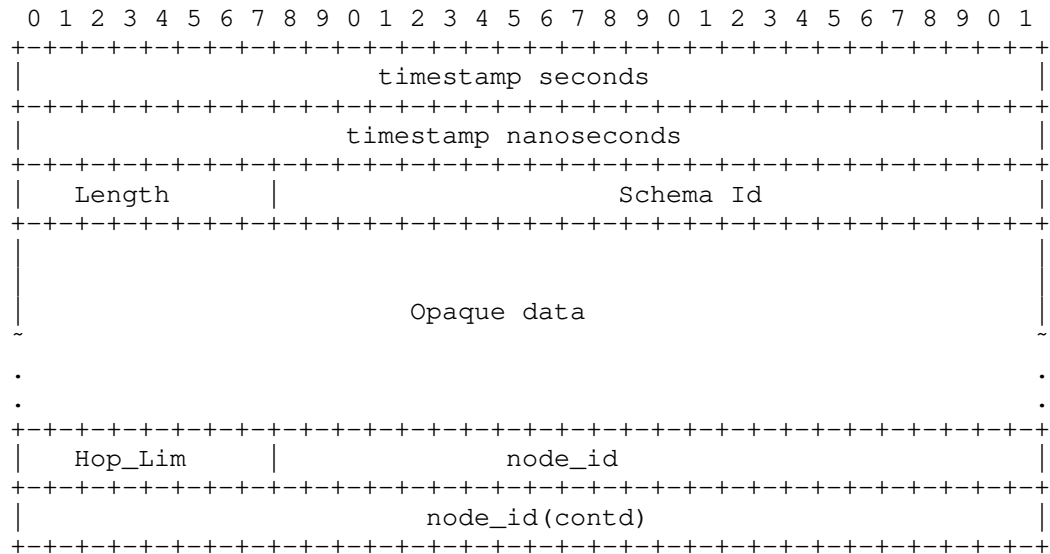
0x0029: IOAM-Trace-Type is 0x0029 then the format is:

```

 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 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| timestamp nanoseconds |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| app_data |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

0x018C: IOAM-Trace-Type is 0x104D then the format is:



4.2. IOAM Proof of Transit Option

IOAM Proof of Transit data is to support the path or service function chain [RFC7665] verification use cases. Proof-of-transit uses methods like nested hashing or nested encryption of the IOAM data or mechanisms such as Shamir's Secret Sharing Schema (SSSS). 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, all of these 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 to the packet:

- o Random: Unique identifier for the packet (e.g., 64-bits allow for the unique identification of 2^{64} packets).
- o Cumulative: Information which is handed from node to node and updated by every node according to a verification algorithm.

IOAM proof of transit option:

IOAM proof of transit option header:

```

 0 1 2 3 4 5 6 7
+-----+
|IOAM POT Type|P|
+-----+
```

IOAM proof of transit option data 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
+-----+-----+-----+-----+-----+-----+-----+-----+<+
|                                     Random                                     | |
+-----+-----+-----+-----+-----+-----+-----+-----+ P
|                                     Random(contd)                             | O
+-----+-----+-----+-----+-----+-----+-----+-----+ T
|                                     Cumulative                                 | |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Cumulative (contd)                         | |
+-----+-----+-----+-----+-----+-----+-----+-----+<+

```

IOAM POT Type: 7-bit identifier of a particular POT variant that dictates the POT data that is included. This document defines POT Type 0:

0: POT data is a 16 Octet field as described below.

Profile to use (P): 1-bit. Indicates which POT-profile is used to generate the Cumulative. Any node participating in POT will have a maximum of 2 profiles configured that drive the computation of cumulative. The two profiles are numbered 0, 1. This bit conveys whether profile 0 or profile 1 is used to compute the Cumulative.

Random: 64-bit Per packet Random number.

Cumulative: 64-bit Cumulative that is updated at specific nodes by processing per packet Random number field and configured parameters.

Note: Larger or smaller sizes of "Random" and "Cumulative" data are feasible and could be required for certain deployments (e.g. in case of space constraints in the transport protocol used). Future versions of this document will address different sizes of data for "proof of transit".

4.3. IOAM Edge-to-Edge Option

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 IOAM transit nodes MAY process the data without modifying it.

Currently only sequence numbers use the IOAM edge-to-edge option. In order to detect packet loss, packet reordering, or packet duplication in an in-situ OAM-domain, sequence numbers can be added to packets of a particular tube (see [I-D.hildebrand-spud-prototype]). Each tube leverages a dedicated namespace for its sequence numbers.

IOAM edge-to-edge option:

IOAM edge-to-edge option header:

```

  0 1 2 3 4 5 6 7
+-----+
| IOAM-E2E-Type |
+-----+
```

IOAM edge-to-edge option data 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           |
+-----+-----+-----+-----+
```

IOAM-E2E-Type: 8-bit identifier of a particular in situ OAM E2E variant.

0: E2E option data is a 64-bit sequence number added to a specific tube which is used to identify packet loss and reordering for that tube.

5. 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 discussion of IOAM data processing and export is left for a future version of this document.

6. IANA Considerations

This document requests the following IANA Actions.

6.1. Creation of a New In-Situ OAM (IOAM) Protocol Parameters IANA registry

IANA is requested to create a new protocol registry for "In-Situ OAM (IOAM) Protocol Parameters". This is the common registry that will include registrations for all IOAM namespaces. Each Registry, whose names are listed below:

IOAM Trace Type

IOAM Trace flags

IOAM POT Type

IOAM E2E Type

will contain the current set of possibilities defined in this document. New registries in this name space are created via RFC Required process as per [RFC8126].

The subsequent sub-sections detail the registries herein contained.

6.2. IOAM Trace Type Registry

This registry defines code point for each bit in the 16-bit IOAM-Trace-Type field for Pre-allocated trace option and Incremental trace option defined in Section 4.1. The meaning of Bit 0 - 11 for trace type are defined in this document in Paragraph 1 of (Section 4.1.1). The meaning for Bit 12 - 15 are available for assignment via RFC Required process as per [RFC8126].

6.3. IOAM Trace Flags Registry

This registry defines code point for each bit in the 5 bit flags for Pre-allocated trace option and Incremental trace option defined in Section 4.1. The meaning of Bit 0 - 1 for trace flags are defined in this document in Paragraph 5 of Section 4.1.1. The meaning for Bit 2 - 4 are available for assignment via RFC Required process as per [RFC8126].

6.4. IOAM POT Type Registry

This registry defines 128 code points to define IOAM POT Type for IOAM proof of transit option Section 4.2. The code point value 0 is defined in this document, 1 - 127 are available for assignment via RFC Required process as per [RFC8126].

6.5. IOAM E2E Type Registry

This registry defines 256 code points to define IOAM-E2E-Type for IOAM E2E option Section 4.3. The code point value 0 is defined in this document, 1 - 255 are available for assignments via RFC Required process as per [RFC8126].

7. Manageability Considerations

Manageability considerations will be addressed in a later version of this document..

8. Security Considerations

Security considerations will be addressed in a later version of this document. For a discussion of security requirements of in-situ OAM, please refer to [I-D.brockners-inband-oam-requirements].

9. 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, and Andrew Yourtchenko 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, and Mickey Spiegel.

10. References

10.1. Normative References

- [IEEE1588v2]
Institute of Electrical and Electronics Engineers,
"1588-2008 - IEEE Standard for a Precision Clock
Synchronization Protocol for Networked Measurement and
Control Systems", IEEE Std 1588-2008, 2008,
<[http://standards.ieee.org/findstds/
standard/1588-2008.html](http://standards.ieee.org/findstds/standard/1588-2008.html)>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,
<<http://www.rfc-editor.org/info/rfc2119>>.
- [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,
<<http://www.rfc-editor.org/info/rfc8126>>.

10.2. 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.
- [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. <>, "Encapsulations for In-situ
OAM Data", draft-brockners-inband-oam-transport-03 (work
in progress), March 2017.
- [I-D.hildebrand-spud-prototype]
Hildebrand, J. and B. Trammell, "Substrate Protocol for
User Datagrams (SPUD) Prototype", draft-hildebrand-spud-
prototype-03 (work in progress), March 2015.
- [I-D.ietf-nvo3-geneve]
Gross, J., Ganga, I., and T. Sridhar, "Geneve: Generic
Network Virtualization Encapsulation", draft-ietf-
nvo3-geneve-04 (work in progress), March 2017.
- [I-D.ietf-nvo3-vxlan-gpe]
Maino, F., Kreeger, L., and U. Elzur, "Generic Protocol
Extension for VXLAN", draft-ietf-nvo3-vxlan-gpe-04 (work
in progress), April 2017.

- [I-D.ietf-sfc-nsh]
Quinn, P. and U. Elzur, "Network Service Header", draft-ietf-sfc-nsh-13 (work in progress), June 2017.
- [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.lapukhov-dataplane-probe]
Lapukhov, P. and r. remy@barefootnetworks.com, "Data-plane probe for in-band telemetry collection", draft-lapukhov-dataplane-probe-01 (work in progress), June 2016.
- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<http://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, <<http://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, <<http://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, <<http://www.rfc-editor.org/info/rfc7821>>.

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

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 2066721
Israel

Email: talmi@marvell.com

David Mozes
Mellanox Technologies Ltd.

Email: davidm@mellanox.com

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

Email: petr@fb.com

Remy Chang
Barefoot Networks
2185 Park Boulevard
Palo Alto, CA 94306
US

Daniel
Bell Canada

Email: daniel.bernier@bell.ca

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 4, 2018

X. Ding
Q. Wu
Huawei
R. Gu
China Mobile
July 3, 2017

An Enhanced Media Delivery Index (eMDI) based on TCP
draft-ding-tcp-emdi-00

Abstract

This document introduces an Enhanced Media Delivery Index (eMDI) that can be used as a diagnostic tool or a quality indicator for monitoring a network intended to deliver streaming media over TCP transport. It aims to address the problems that RFC4445 has when measuring in environments where TCP traffic is dominated as a transport for streaming media.

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 January 4, 2018.

Copyright Notice

Copyright (c) 2017 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. Terminologies	3
3. Measurement Setup	3
4. Measurement Method	4
5. Use Examples	5
5.1. Network Troubleshooting in VoD scenario	5
5.2. WiFi Anomaly Analysis in the Home Network	6
6. Security Considerations	6
7. Normative References	6
Authors' Addresses	7

1. Introduction

TCP is one major transport protocol in use in most IP networks, and supports the transfer of over 80 percent of all traffic (e.g., OTT traffic, IPTV VOD traffic) across the public Internet today. Packet loss ratio and latency are two major characteristics in the network to affect the behavior of TCP. The bad TCP performance might also indicate the unacceptable end-user-perceived quality level.

Media Delivery Index (MDI) [RFC4445] is a method widely used in the network as a diagnostic tool to measure both the instantaneous and longer-term behavior of networks carrying streaming media in the media layer. However the limitation of MDI measurement is mostly applicable to streaming media and protocol over UDP, it falls short when monitoring a network intended to deliver multimedia applications over TCP Transport, i.e., the traditional MDI metrics especially Media Loss Rate (MLR) deployed in the network devices is difficult to infer the packet loss if the missing packets were retransmitted when the packet loss was detected by the TCP sender. On the other hand, TCP sender will adjust the sending data rate to reduce the probability of further packet loss, which means throughput is declining when extra delay is incurred by retransmitting lost packets. Therefore, throughput can be regarded as a quality indication for network monitoring and diagnosis.

This document introduces a new measurement method and associated metrics, i.e., downstream/upstream/end to end throughput, to complement methods defined in [RFC4445]. This new method can quickly identify the root cause of the QoS related problem, improve efficiency of network monitoring and troubleshooting.

2. Terminologies

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].

This document uses the following terms:

Measurement point (MP): A measurement point is the logical or physical location defined in the TCP that acts as a source of information gathered for monitoring purposes.

Upstream packet lost ratio (UPLR): UPLR is the ratio of the number of packets lost to the total number of packets sent from server to measurement point during predefined measurement interval.

Downstream packet lost ratio (DPLR): DPLR is the ratio of the number of packets lost to the total number of packets sent from measurement point to client during a predefined measurement interval.

Upstream average RTT (URTT): URTT is the average RTT at the path from server to measurement point during a predefined measurement interval.

Downstream average RTT (DRTT): DRTT is the average RTT at the path from measurement point to client during a predefined measurement interval.

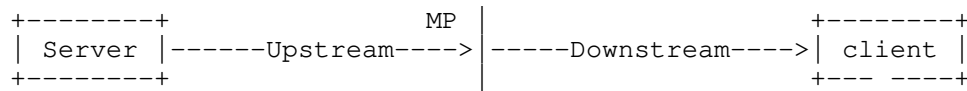
End to end Throughput (E2ET): E2ET is the rate of successful packet delivery over an end-end network path during a predefined measurement interval.

Downstream throughput (DT): DT is measured by the number of packets received per second at the downstream of measurement point during a predefined measurement interval.

Upstream throughput (UT): UT is measured by the number of packets received per second at the upstream of measurement point during a predefined measurement interval.

3. Measurement Setup

A stream of packets sent by streaming media Server passes through MP (MP can be bridge, router or gateway), and finally reach the client (destination endpoint). If a node A is placed between the server and MP in the network, then A is upstream node of MP. Otherwise, A is downstream node of MP.



4. Measurement Method

The rationale of the measurement is to compare DT/UT/E2ET with data packet rate. If DT is less than data packet rate and UT is greater than data packet rate, there is something wrong with the downstream network. Otherwise, the upstream network has some problems.

When the packet loss occurs in the network, an additional limit (i.e., packet loss probability) is imposed on the throughput besides TCP receive window. In case of light or moderate packet loss when the TCP rate is adjusted by the congestion avoidance algorithm, DT can be calculated according to the following formula:

$$DT = MSS / (DRTT + URTT) (DPLR) (1/2);$$

Where MSS is the maximum segment size. Assuming the number of lost packets at the downstream during a predefined measurement interval is a , and the number of total packets sent by MP is x , then DPLR is then calculated as following:

$$DPLR = a/x.$$

Average RTT of some packets ($d1..dm$) at the downstream direction are used to compute DRTT:

$$DRTT = \sum (RTT_{di}) / m, i = 1..m$$

Where RTT_{di} indicates the RTT of packet d_i at downstream.

Similarly, average RTT of some packets ($u1..un$) at the upstream direction are used to compute URTT:

$$URTT = \sum (RTT_{ui}) / n, i = 1..n$$

Where RTT_{ui} indicates the RTT of packet u_i at the upstream.

And, UT can be calculated according to the formula:

$$UT = MSS / (DRTT + URTT) (UPLR) (1/2);$$

Assuming the number of lost packets at the upstream during a predefined measurement interval is b , and the number of total packets sent by Server is y , then UPLR is then calculated as following:

$$UPLR = b/y.$$

And E2ET can be calculated according to the formula:

$$E2ET = MSS / (DRTT + URTT) (UPLR + DPLR) (1/2) .$$

5. Use Examples

5.1. Network Troubleshooting in VoD scenario

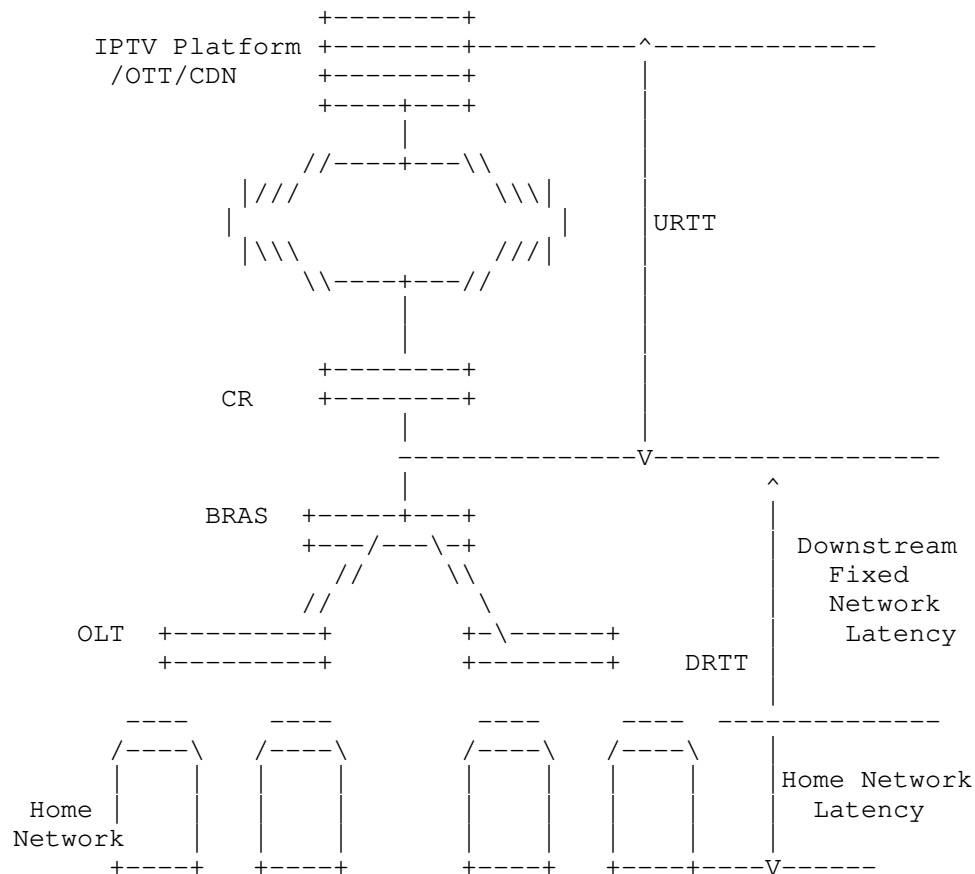


Figure 1: Figure 1

The proposed measurement method can be applied when VoD streaming media running over TCP is delivered as unicast stream from VoD server in the operator network to end users in home network. In some cases, the fault occurs in the home network which cause user experience downgrading, in some other cases, fault occurs in the operator network.

To pinpoint the location of the fault, MP can be deployed on ONT device of the home network. The home network is refer to the downstream of the MP and the operator network is refer to the upstream of the MP. Suppose the rate of the media rate is v , we can compare $DT/UT/E2ET$ with v . If $DT < v$ and $UT > v$, the home network is the root cause for streaming media quality downgrading. If $DT > v$ and $UT < v$, the operator network is the root cause. If $DT > v$, $UT > v$, and $E2E < v$, both home network and operator network should be responsible for streaming media quality downgrading.

5.2. WiFi Anomaly Analysis in the Home Network

WiFi latency is a key factor impacting the user experience of home network application. [WIFI] shows WiFi latency follows a long tail distribution: its 50th, 90th and 99th percentile are around 3ms, 20ms and 250ms. If the WiFi network get congested, the quality degrades proportionally with WiFi latency. To analyse WiFi Anomaly degree in the home network, See figure 1, we can calculate cumulative distribution of WiFi latency based on measured values:

WiFi Latency = DRTT - Downstream Fixed Network Latency

and determine threshold value for WiFi Latency based on periodically collected dataset, e.g.,

Threshold = $UBV + coef * (UBV - LBV)$

Where UBV is the 75th percentile value, LBV is the 25th Percentile value, $coef$ is coefficient value which can be set to 1.5.

By Comparing WiFi latency measured value with the threshold value, we can decide if WiFi Anomaly is the root cause of network quality degrading.

6. Security Considerations

This document does not introduce security issues beyond those discussed in [RFC4445].

7. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", March 1997.
- [RFC4445] Welch, J. and J. Clark, "A Proposed Media Delivery Index (MDI)", RFC 4445, DOI 10.17487/RFC4445, April 2006, <<http://www.rfc-editor.org/info/rfc4445>>.

[WIFI] MobiSys'16, 2016, Singapore, ACM ISBN
 978-1-4503-4269-8/16/06, "Characterizing and Improving
 WiFi Latency in Large-Scale Operational Networks", 2016.

Authors' Addresses

Xiaojian Ding
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: dingxiaojian1@huawei.com

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: bill.wu@huawei.com

Rong Gu
China Mobile
32 Xuanwumen West Ave, Xicheng District
Beijing 100053
China

Email: gurong_cmcc@outlook.com

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.

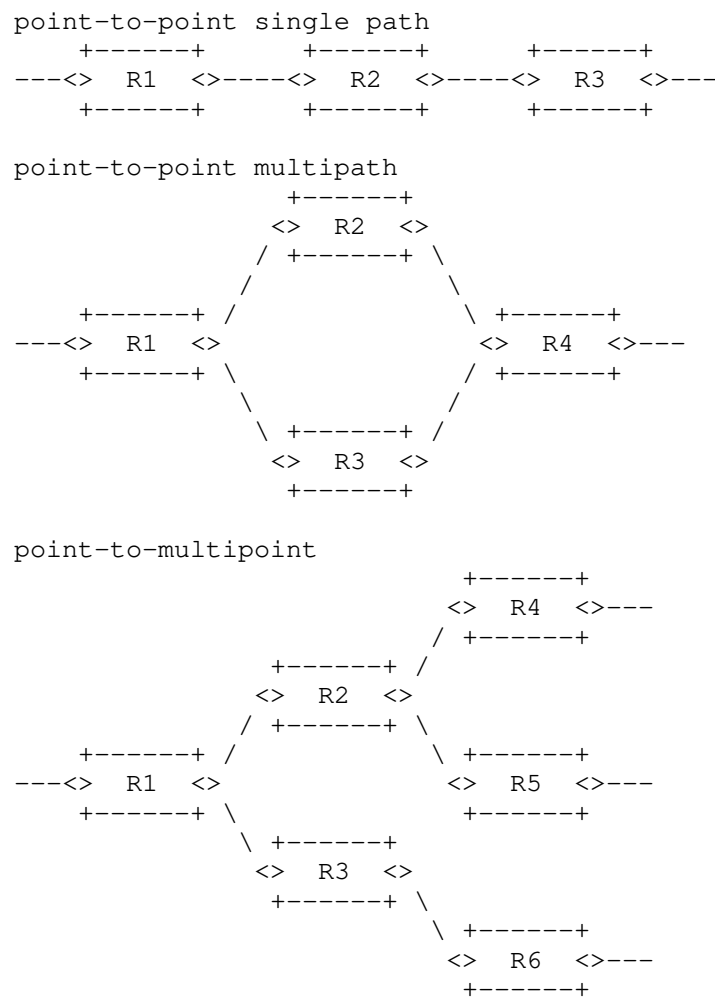
The choice of the identification fields directly affects the type of paths that the flow would follow in the network. In fact, it is possible to relate a set of identification fields with the pattern of the resulting graphs, as listed in Figure 1.

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.



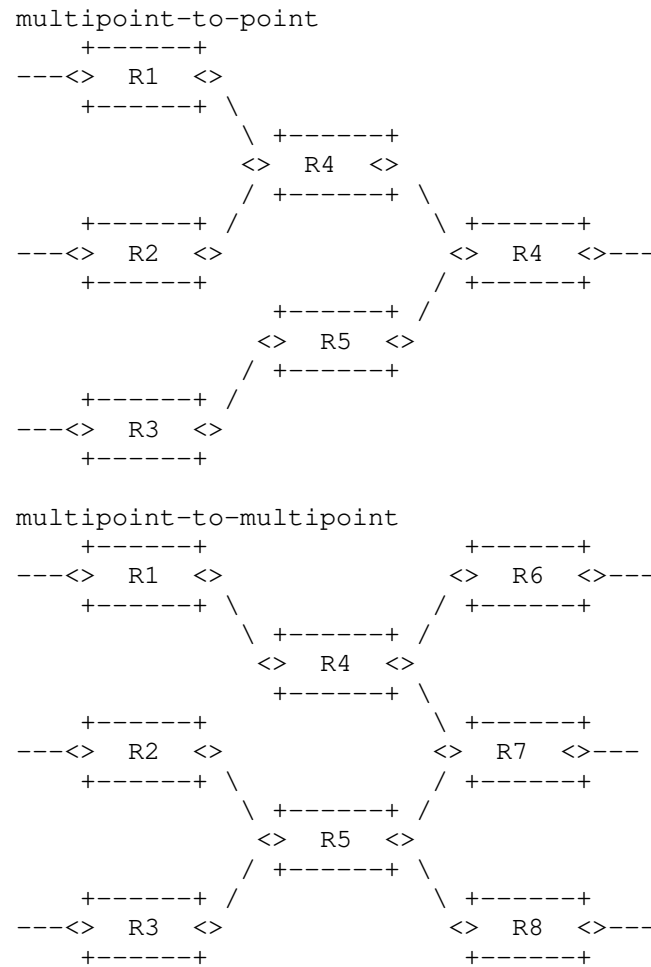


Figure 1: Flow classification

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:

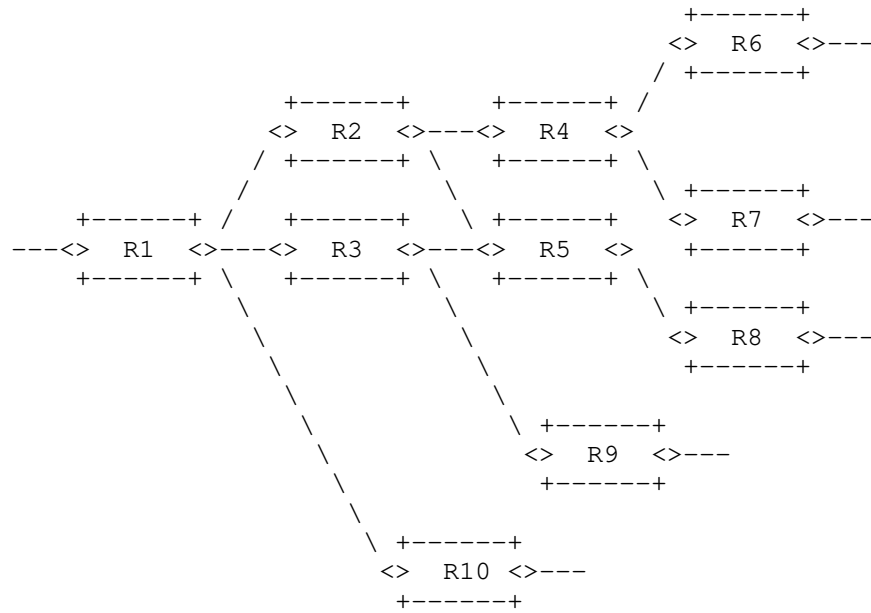


Figure 2: Monitoring Network Graph

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:

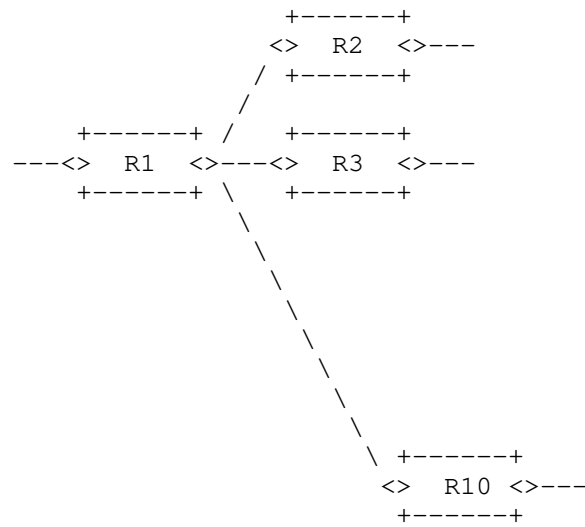
1. Group 1: (R1-R2), (R1-R3), (R1-R10)
2. Group 2: (R2-R4), (R2-R5)
3. Group 3: (R3-R5), (R3-R9)
4. Group 4: (R4-R6), (R4-R7)
5. Group 5: (R5-R8)

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):

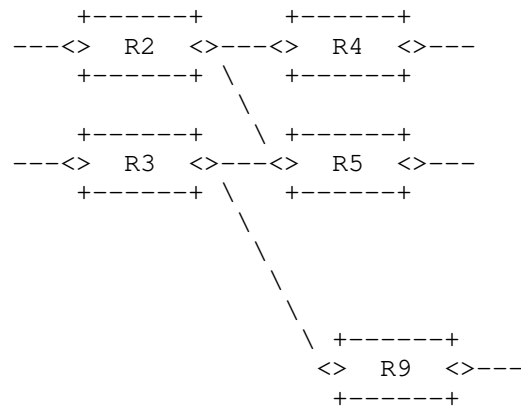
1. Cluster 1: (R1-R2), (R1-R3), (R1-R10)
2. Cluster 2: (R2-R4), (R2-R5), (R3-R5), (R3-R9)
3. Cluster 3: (R4-R6), (R4-R7)
4. Cluster 4: (R5-R8)

In the end the following 4 Clusters are obtained:

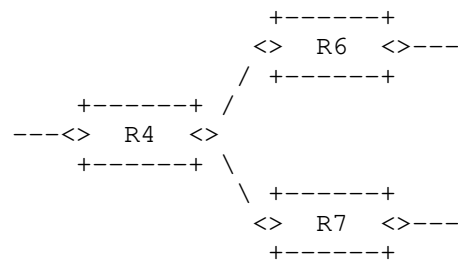
Cluster 1



Cluster 2



Cluster 3



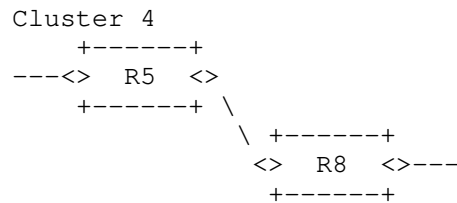


Figure 3: Clusters example

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.

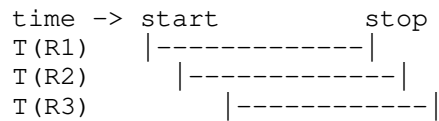


Figure 4: Measurement Interval

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.

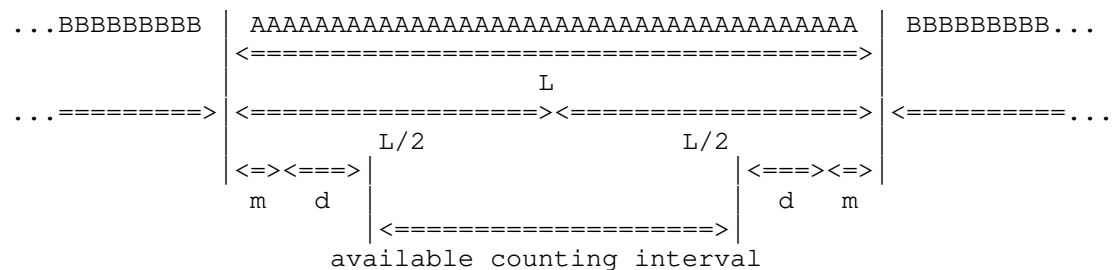


Figure 5: Timing Aspects for Multipoint paths

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

Network Working Group
Internet-Draft
Updates: 2330 (if approved)
Intended status: Informational
Expires: January 1, 2019

A. Morton
AT&T Labs
J. Fabini
TU Wien
N. Elkins
Inside Products, Inc.
M. Ackermann
Blue Cross Blue Shield of Michigan
V. Hegde
Consultant
June 30, 2018

IPv6, IPv4 and Coexistence Updates for IPPM's Active Metric Framework
draft-ietf-ippm-2330-ipv6-06

Abstract

This memo updates the IP Performance Metrics (IPPM) Framework RFC 2330 with new considerations for measurement methodology and testing. It updates the definition of standard-formed packets in RFC 2330 to include IPv6 packets, deprecates the definition of minimal IP packet, and augments distinguishing aspects of packets, referred to as Type-P for test packets in RFC 2330. This memo identifies that IPv4-IPv6 co-existence can challenge measurements within the scope of the IPPM Framework. Example use cases include, but are not limited to IPv4-IPv6 translation, NAT, or protocol encapsulation. IPv6 header compression and use of IPv6 over Low-Power Wireless Area Networks (6LoWPAN) are considered and excluded from the standard-formed packet evaluation.

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 January 1, 2019.

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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	3
2. Scope	3
3. Packets of Type-P	3
4. Standard-Formed Packets	5
5. NAT, IPv4-IPv6 Transition and Compression Techniques	8
6. Security Considerations	10
7. IANA Considerations	10
8. Acknowledgements	10
9. References	10
9.1. Normative References	10
9.2. Informative References	13
Authors' Addresses	14

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 IPPM framework [RFC2330] recognized (in section 13) that many aspects of IP packets can influence its processing during transfer across the network.

In Section 15 of [RFC2330], the notion of a "standard-formed" packet is defined. However, the definition was never updated to include IPv6, as the original authors originally desired to do.

In particular, IPv6 Extension Headers and protocols which use IPv6 header compression are growing in use. This memo seeks to provide the needed updates.

2. Scope

The purpose of this memo is to expand the coverage of IPPM metrics to include IPv6, and to highlight additional aspects of test packets and make them part of the IPPM performance metric framework.

The scope is to update key sections of [RFC2330], adding considerations that will aid the development of new measurement methodologies intended for today's IP networks. Specifically, this memo expands the Type-P examples in section 13 of [RFC2330] and expands the definition (in section 15 of [RFC2330]) of a standard-formed packet to include IPv6 header aspects and other features.

Other topics in [RFC2330] which might be updated or augmented are deferred to future work. This includes the topics of passive and various forms of hybrid active/passive measurements.

3. Packets of Type-P

A fundamental property of many Internet metrics is that the measured value of the metric depends on characteristics of the IP packet(s) used to make the measurement. Potential influencing factors include IP header fields and their values, but also higher-layer protocol headers and their values. Consider an IP-connectivity metric: one obtains different results depending on whether one is interested in connectivity for packets destined for well-known TCP ports or unreserved UDP ports, or those with invalid IPv4 checksums, or those

with TTL or Hop Limit of 16, for example. In some circumstances these distinctions will result in special treatment of packets in intermediate nodes and end systems (for example, if Diffserv [RFC2474], ECN [RFC3168], Router Alert [RFC6398], Hop-by-hop extensions [RFC7045], or Flow Labels [RFC6437] are used, or in the presence of firewalls or RSVP reservations).

Because of this distinction, we introduce the generic notion of a "packet of Type-P", where in some contexts P will be explicitly defined (i.e., exactly what type of packet we mean), partially defined (e.g., "with a payload of B octets"), or left generic. Thus we may talk about generic IP-Type-P-connectivity or more specific IP-port-HTTP-connectivity. Some metrics and methodologies may be fruitfully defined using generic Type-P definitions which are then made specific when performing actual measurements.

Whenever a metric's value depends on the type of the packets involved in the metric, the metric's name will include either a specific type or a phrase such as "Type-P". Thus we will not define an "IP-connectivity" metric but instead an "IP-Type-P-connectivity" metric and/or perhaps an "IP-port-HTTP-connectivity" metric. This naming convention serves as an important reminder that one must be conscious of the exact type of traffic being measured.

If the information constituting Type-P at the Source is found to have changed at the Destination (or at a measurement point between the Source and Destination, as in [RFC5644]), then the modified values MUST be noted and reported with the results. Some modifications occur according to the conditions encountered in transit (such as congestion notification) or due to the requirements of segments of the Source to Destination path. For example, the packet length will change if IP headers are converted to the alternate version/address family, or if optional Extension Headers are added or removed. Even header fields like TTL/Hop Limit that typically change in transit may be relevant to specific tests. For example Neighbor Discovery Protocol (NDP) [RFC4861] packets are transmitted with Hop Limit value set to 255, and the validity test specifies that the Hop Limit MUST have a value of 255 at the receiver, too. So, while other tests may intentionally exclude the TTL/Hop Limit value from their Type-P definition, for this particular test the correct Hop Limit value is of high relevance and MUST be part of the Type-P definition.

Local policies in intermediate nodes based on examination of IPv6 Extension Headers may affect measurement repeatability. If intermediate nodes follow the recommendations of [RFC7045], repeatability may be improved to some degree.

A closely related note: it would be very useful to know if a given Internet component (like host, link, or path) treats equally a class C of different types of packets. If so, then any one of those types of packets can be used for subsequent measurement of the component. This suggests we devise a metric or suite of metrics that attempt to determine class C (a designation which has no relationship to address assignments, of course).

Load balancing over parallel paths is one particular example where such a class C would be more complex to determine in IPPM measurements. Load balancers and routers often use flow identifiers, computed as hashes of (specific parts of) the packet header, for deciding among the available parallel paths a packet will traverse. Packets with identical hashes are assigned to the same flow and forwarded to the same resource in the load balancer's (or router's) pool. The presence of a load balancer on the measurement path, as well as the specific headers and fields that are used for the forwarding decision, are not known when measuring the path as a black-box. Potential assessment scenarios include the measurement of one of the parallel paths, and the measurement of all available parallel paths that the load balancer can use. Knowledge of a load balancer's flow definition (alternatively: its class C specific treatment in terms of header fields in scope of hash operations) is therefore a prerequisite for repeatable measurements. A path may have more than one stage of load balancing, adding to class C definition complexity.

4. Standard-Formed Packets

Unless otherwise stated, all metric definitions that concern IP packets include an implicit assumption that the packet is **standard-formed**. A packet is standard-formed if it meets all of the following REQUIRED criteria:

- + It includes a valid IP header: see below for version-specific criteria.
- + It is not an IP fragment.
- + The Source and Destination addresses correspond to the intended Source and Destination, including Multicast Destination addresses.
- + If a transport header is present, it contains a valid checksum and other valid fields.

For an IPv4 ([RFC0791] and updates) packet to be standard-formed, the following additional criteria are REQUIRED:

- o The version field is 4
- o The Internet Header Length (IHL) value is ≥ 5 ; the checksum is correct.
- o Its total length as given in the IPv4 header corresponds to the size of the IPv4 header plus the size of the payload.
- o Either the packet possesses sufficient TTL to travel from the Source to the Destination if the TTL is decremented by one at each hop, or it possesses the maximum TTL of 255.
- o It does not contain IP options unless explicitly noted.

For an IPv6 ([RFC8200] and updates) packet to be standard-formed, the following criteria are REQUIRED:

- o The version field is 6.
- o Its total length corresponds to the size of the IPv6 header (40 octets) plus the length of the payload as given in the IPv6 header.
- o The payload length value for this packet (including Extension Headers) conforms to the IPv6 specifications.
- o Either the packet possesses sufficient Hop Limit to travel from the Source to the Destination if the Hop Limit is decremented by one at each hop, or it possesses the maximum Hop Limit of 255.
- o Either the packet does not contain IP Extension Headers, or it contains the correct number and type of headers as specified in the packet, and the headers appear in the standard-conforming order (Next Header).
- o All parameters used in the header and Extension Headers are found in the IANA Registry of Internet Protocol Version 6 (IPv6) Parameters, specified in [IANA-6P].

Two mechanisms require some discussion in the context of standard-formed packets, namely IPv6 over Low-Power Wireless Area Networks (6LowPAN, [RFC4944]) and Robust Header Compression (ROHC, [RFC3095]). IPv6 over Low-Power Wireless Area Networks (6LowPAN), as defined in [RFC4944] and updated by [RFC6282] with header compression and [RFC6775] with neighbor discovery optimizations, proposes solutions for using IPv6 in resource-constrained environments. An adaptation layer enables the transfer of IPv6 packets over networks having a MTU smaller than the minimum IPv6 MTU. Fragmentation and re-assembly of

IPv6 packets, as well as the resulting state that would be stored in intermediate nodes, poses substantial challenges to measurements. Likewise, ROHC operates statefully in compressing headers on subpaths, storing state in intermediate hosts. The modification of measurement packets' Type-P by ROHC and 6LowPAN, as well as requirements with respect to the concept of standard-formed packets for these two protocols requires substantial work. Because of these reasons we consider ROHC and 6LowPAN packets to be out of the scope for the standard-formed packet evaluation.

The topic of IPv6 Extension Headers brings current controversies into focus as noted by [RFC6564] and [RFC7045]. However, measurement use cases in the context of the IPPM framework like in-situ OAM [I-D.ietf-ippm-ioam-data] in enterprise environments can benefit from inspection, modification, addition or deletion of IPv6 extension headers in hosts along the measurement path.

[RFC8250] endorses the use of IPv6 Destination Option for measurement purposes, consistent with other approved IETF specifications.

The following additional considerations apply when IPv6 Extension Headers are present:

- o Extension Header inspection: Some intermediate nodes may inspect Extension Headers or the entire IPv6 packet while in transit. In exceptional cases, they may drop the packet or route via a sub-optimal path, and measurements may be unreliable or unrepeatable. The packet (if it arrives) may be standard-formed, with a corresponding Type-P.
- o Extension Header modification: In Hop-by-Hop headers, some TLV encoded options may be permitted to change at intermediate nodes while in transit. The resulting packet may be standard-formed, with a corresponding Type-P.
- o Extension Header insertion or deletion: Although such behavior is not endorsed by current standards, it is possible that Extension Headers could be added to, or removed from the header chain. The resulting packet may be standard-formed, with a corresponding Type-P. This point simply encourages measurement system designers to be prepared for the unexpected, and to notify users when such events occur. There are issues with Extension Header insertion and deletion of course, such as exceeding the path MTU due to insertion, etc.
- o A change in packet length (from the corresponding packet observed at the Source) or header modification is a significant factor in

Internet measurement, and REQUIRES a new Type-P to be reported with the test results.

It is further REQUIRED that if a packet is described as having a "length of B octets", then $0 \leq B \leq 65535$; and if B is the payload length in octets, then $B \leq (65535 - \text{IP header size in octets, including any Extension Headers})$. The jumbograms defined in [RFC2675] are not covered by the above length analysis, but if the IPv6 Jumbogram Payload Hop-by-Hop Option Header is present, then a packet with corresponding length MUST be considered standard-formed. In practice, the path MTU will restrict the length of standard-formed packets that can successfully traverse the path. Path MTU Discovery for IP version 6 (PMTUD, [RFC8201]) or Packetization Layer Path MTU Discovery (PLPMTUD, [RFC4821]) is recommended to prevent fragmentation.

So, for example, one might imagine defining an IP connectivity metric as "IP-type-P-connectivity for standard-formed packets with the IP Diffserv field set to 0", or, more succinctly, "IP-type-P-connectivity with the IP Diffserv Field set to 0", since standard-formed is already implied by convention. Changing the contents of a field, such as the Diffserv Code Point, ECN bits, or Flow Label may have a profound affect on packet handling during transit, but does not affect a packet's status as standard-formed. Likewise, the addition, modification, or deletion of extension headers may change the handling of packets in transit hosts.

[RFC2330] defines the "minimal IP packet from A to B" as a particular type of standard-formed packet often useful to consider. When defining IP metrics no packet smaller or simpler than this can be transmitted over a correctly operating IP network. However, the concept of the minimal IP packet has not been employed (since typical active measurement systems employ a transport layer and a payload) and its practical use is limited. Therefore, this memo deprecates the concept of the "minimal IP packet from A to B".

5. NAT, IPv4-IPv6 Transition and Compression Techniques

This memo adds the key considerations for utilizing IPv6 in two critical conventions of the IPPM Framework, namely packets of Type-P and standard-formed packets. The need for co-existence of IPv4 and IPv6 has originated transitioning standards like the Framework for IPv4/IPv6 Translation in [RFC6144] or IP/ICMP Translation Algorithms in [RFC7915] and [RFC7757].

The definition and execution of measurements within the context of the IPPM Framework is challenged whenever such translation mechanisms are present along the measurement path. In particular use cases like

IPv4-IPv6 translation, NAT, protocol encapsulation, or IPv6 header compression may result in modification of the measurement packet's Type-P along the path. All these changes **MUST** be reported. Example consequences include, but are not limited to:

- o Modification or addition of headers or header field values in intermediate nodes. IPv4-IPv6 transitioning or IPv6 header compression mechanisms may result in changes of the measurement packets' Type-P, too. Consequently, hosts along the measurement path may treat packets differently because of the Type-P modification. Measurements at observation points along the path may also need extra context to uniquely identify a packet.
- o Network Address Translators (NAT) on the path can have unpredictable impact on latency measurement (in terms of the amount of additional time added), and possibly other types of measurements. It is not usually possible to control this impact (as testers may not have any control of the underlying network or middleboxes). There is a possibility that stateful NAT will lead to unstable performance for a flow with specific Type-P, since state needs to be created for the first packet of a flow, and state may be lost later if the NAT runs out of resources. However, this scenario does not invalidate the Type-P for testing - for example the purpose of a test might be exactly to quantify the NAT's impact on delay variation. The presence of NAT may mean that the measured performance of Type-P will change between the source and the destination. This can cause an issue when attempting to correlate measurements conducted on segments of the path that include or exclude the NAT. Thus, it is a factor to be aware of when conducting measurements.
- o Variable delay due to internal state. One side effect of changes due to IPv4-IPv6 transitioning mechanisms is the variable delay that intermediate nodes spend for header modifications. Similar to NAT the allocation of internal state and establishment of context within intermediate nodes may cause variable delays, depending on the measurement stream pattern and position of a packet within the stream. For example the first packet in a stream will typically trigger allocation of internal state in an intermediate IPv4-IPv6 transition host. Subsequent packets can benefit from lower processing delay due to the existing internal state. However, large inter-packet delays in the measurement stream may result in the intermediate host deleting the associated state and needing to re-establish it on arrival of another stream packet. It is worth noting that this variable delay due to internal state allocation in intermediate nodes can be an explicit use case for measurements.

- o Variable delay due to packet length. IPv4-IPv6 transitioning or header compression mechanisms modify the length of measurement packets. The modification of the packet size may or may not change the way how the measurement path treats the packets.

6. Security Considerations

The security considerations that apply to any active measurement of live paths are relevant here as well. See [RFC4656] and [RFC5357].

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.

7. IANA Considerations

This memo makes no requests of IANA.

8. Acknowledgements

The authors thank Brian Carpenter for identifying the lack of IPv6 coverage in IPPM's Framework, and for listing additional distinguishing factors for packets of Type-P. Both Brian and Fred Baker discussed many of the interesting aspects of IPv6 with the co-authors, leading to a more solid first draft: thank you both. Thanks to Bill Jouris for an editorial pass through the pre-00 text. As we completed our journey, Nevil Brownlee, Mike Heard, Spencer Dawkins, Warren Kumari, and Suresh Krishnan all contributed useful suggestions.

9. References

9.1. Normative References

- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/info/rfc791>>.
- [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>>.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <<https://www.rfc-editor.org/info/rfc2474>>.
- [RFC2675] Borman, D., Deering, S., and R. Hinden, "IPv6 Jumbograms", RFC 2675, DOI 10.17487/RFC2675, August 1999, <<https://www.rfc-editor.org/info/rfc2675>>.
- [RFC3095] Bormann, C., Burmeister, C., Degermark, M., Fukushima, H., Hannu, H., Jonsson, L-E., Hakenberg, R., Koren, T., Le, K., Liu, Z., Martensson, A., Miyazaki, A., Svanbro, K., Wiebke, T., Yoshimura, T., and H. Zheng, "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed", RFC 3095, DOI 10.17487/RFC3095, July 2001, <<https://www.rfc-editor.org/info/rfc3095>>.
- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <<https://www.rfc-editor.org/info/rfc3168>>.
- [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>>.
- [RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, DOI 10.17487/RFC4821, March 2007, <<https://www.rfc-editor.org/info/rfc4821>>.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <<https://www.rfc-editor.org/info/rfc4861>>.
- [RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <<https://www.rfc-editor.org/info/rfc4944>>.

- [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>>.
- [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>>.
- [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>>.
- [RFC6144] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", RFC 6144, DOI 10.17487/RFC6144, April 2011, <<https://www.rfc-editor.org/info/rfc6144>>.
- [RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <<https://www.rfc-editor.org/info/rfc6282>>.
- [RFC6398] Le Faucheur, F., Ed., "IP Router Alert Considerations and Usage", BCP 168, RFC 6398, DOI 10.17487/RFC6398, October 2011, <<https://www.rfc-editor.org/info/rfc6398>>.
- [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>>.
- [RFC6564] Krishnan, S., Woodyatt, J., Kline, E., Hoagland, J., and M. Bhatia, "A Uniform Format for IPv6 Extension Headers", RFC 6564, DOI 10.17487/RFC6564, April 2012, <<https://www.rfc-editor.org/info/rfc6564>>.
- [RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <<https://www.rfc-editor.org/info/rfc6775>>.
- [RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing of IPv6 Extension Headers", RFC 7045, DOI 10.17487/RFC7045, December 2013, <<https://www.rfc-editor.org/info/rfc7045>>.

- [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>>.
- [RFC7757] Anderson, T. and A. Leiva Popper, "Explicit Address Mappings for Stateless IP/ICMP Translation", RFC 7757, DOI 10.17487/RFC7757, February 2016, <<https://www.rfc-editor.org/info/rfc7757>>.
- [RFC7915] Bao, C., Li, X., Baker, F., Anderson, T., and F. Gont, "IP/ICMP Translation Algorithm", RFC 7915, DOI 10.17487/RFC7915, June 2016, <<https://www.rfc-editor.org/info/rfc7915>>.
- [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>>.
- [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>>.
- [RFC8201] McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed., "Path MTU Discovery for IP version 6", STD 87, RFC 8201, DOI 10.17487/RFC8201, July 2017, <<https://www.rfc-editor.org/info/rfc8201>>.
- [RFC8250] Elkins, N., Hamilton, R., and M. Ackermann, "IPv6 Performance and Diagnostic Metrics (PDM) Destination Option", RFC 8250, DOI 10.17487/RFC8250, September 2017, <<https://www.rfc-editor.org/info/rfc8250>>.

9.2. Informative References

- [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-03 (work in progress), June 2018.
- [IANA-6P] IANA, "IANA Internet Protocol Version 6 (IPv6) Parameters", Internet Assigned Numbers Authority <https://www.iana.org/assignments/ipv6-parameters>, January 2018.

[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
URI: <http://home.comcast.net/~acmacm/>

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/>

Nalini Elkins
Inside Products, Inc.
Carmel Valley, CA 93924
USA

Email: nalini.elkins@insidethestack.com

Michael S. Ackermann
Blue Cross Blue Shield of Michigan

Email: mackermann@bcbsm.com

Vinayak Hegde
Consultant
Brahma Sun City, Wadgaon-Sheri
Pune, Maharashtra 411014
INDIA

Phone: +91 9449834401
Email: vinayakh@gmail.com
URI: <http://www.vinayakhegde.com>

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 15, 2017

G. Mirsky
X. Min
ZTE Corp.
A. Pan
W. Luo
Ericsson
June 13, 2017

Two-Way Active Measurement Protocol (TWAMP) Light Data Model
draft-mirsky-ippm-twamp-light-yang-09

Abstract

This document specifies the data model for implementations of Session-Sender and Session-Reflector for Two-Way Active Measurement Protocol (TWAMP) Light mode using YANG.

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 December 15, 2017.

Copyright Notice

Copyright (c) 2017 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
1.1. Conventions used in this document	2
1.1.1. Requirements Language	2
2. Scope, Model, and Applicability	3
2.1. Data Model Parameters	3
2.1.1. Session-Sender	3
2.1.2. Session-Reflector	4
3. Data Model	4
3.1. Tree Diagram	5
3.2. YANG Module	9
4. IANA Considerations	27
5. Security Considerations	28
6. References	28
6.1. Normative References	28
6.2. Informative References	29
Appendix A. Acknowledgements	29
Authors' Addresses	29

1. Introduction

The Two-Way Active Measurement Protocol (TWAMP) [RFC5357] can be used to measure performance parameters of IP networks such as latency, jitter, and packet loss by sending test packets and monitoring their experience in the network. The [RFC5357] defines two protocols, TWAMP Control and TWAMP Test, and a profile of TWAMP Test, TWAMP Light. The TWAMP Light is known to have many implementations though no common management framework being defined, thus leaving some aspects of test packet processing to interpretation. The goal of this document is to collect analyze these variations; describe common model while allowing for extensions in the future. This document defines such a TWAMP data model and specifies it formally using the YANG data modeling language [RFC6020].

1.1. Conventions used in this document

1.1.1. 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. Scope, Model, and Applicability

The scope of this document includes model of the TWAMP Light as defined in Appendix I of [RFC5357]. This mode of TWAMP Light will be referred in this document as Stateless. Another mode, where the Session-Reflector is aware of the state of the TWAMP test session and thus can independently count reflected test packets, referred as Stateful. This document benefits from earlier attempt to define TWAMP MIB in [I-D.elteto-ippm-twamp-mib] and from TWAMP YANG model defined in [I-D.ietf-ippm-twamp-yang].

Figure 1 updates TWAMP-Light reference model presented in Appendix I [RFC5357] for the scenario when instantiation of a TWAMP-Test session between Session-Sender and Session-Reflector controlled by communication between a Configuration Client as a manager and Configuration Servers as agents of the configuration session.

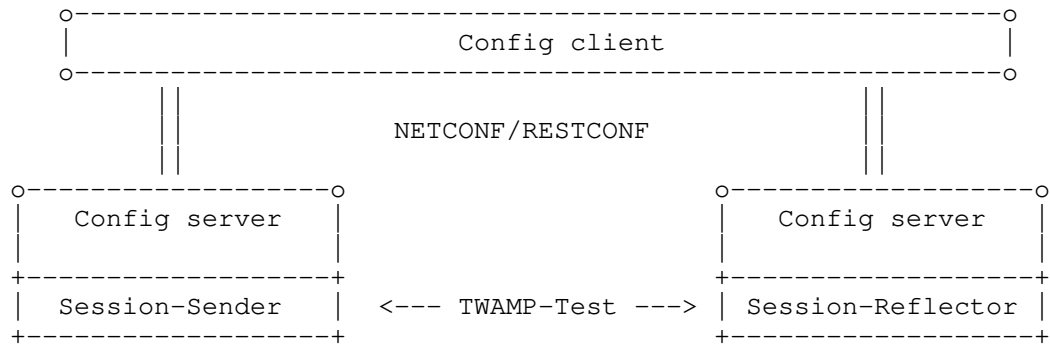


Figure 1: TWAMP Light Reference Model

2.1. Data Model Parameters

This section describes all the parameters of the the TWAMP-Light data model.

2.1.1. Session-Sender

The `twamp-light-session-sender` container holds items that are related to the configuration of the TWAMP-Light Session-Sender logical entity.

The `twamp-light-session-sender-state` container holds information about the state of the particular TWAMP-Light test session.

RPCs `twamp-sender-start` and `twamp-sender-stop` respectively start and stop the referenced by `session-id` TWAMP-Light test session.

2.1.1.1. Controls for Test Session and Performance Metric Calculation

The data model supports several scenarios for Session-Sender to execute test sessions and calculate performance metrics:

The test mode in which the test packets are sent unbound in time at defined by the parameter `'interval'` in the `twamp-light-session-sender` container frequency is referred as continuous mode. Performance metrics in the continuous mode are calculated at period defined by the parameter `'measurement-interval'`.

The test mode that has specific number of the test packets configured for the test session in the `'number-of-packets'` parameter is referred as periodic mode. The test session may be repeated by the Session-Sender with the same parameters. The `'repeat'` parameter defines number of tests and the `'repeat-interval'` - the interval between the consecutive tests. The performance metrics are calculated after each test session when the interval defined by the `'session-timeout'` expires.

2.1.2. Session-Reflector

The `twamp-light-session-reflector` container holds items that are related to the configuration of the TWAMP-Light Session-Reflector logical entity.

The `twamp-light-session-refl-state` container holds Session-Reflector state data for the particular TWAMP-Light test session.

3. Data Model

Creating TWAMP-Light data model presents number of challenges and among them is identification of a test-session at Session-Reflector. A Session-Reflector MAY require only as little as its IP and UDP port number in received TWAMP-Test packet to spawn new test session. More so, to test processing of Class-of-Service along the same route in Equal Cost Multi-Path environment Session-Sender may run TWAMP test sessions concurrently using the same source IP address, source UDP port number, destination IP address, and destination UDP port number. Thus the only parameter that can be used to differentiate these test sessions would be DSCP value. The DSCP field may get re-marked along the path and without use of [RFC7750] that will go undetected, but by using five-tuple instead of four-tuple as a key we can ensure that TWAMP test packets that are considered as different test sessions follow the same path even in ECMP environments.

3.1. Tree Diagram

```

module: ietf-twamp-light
  +--rw twamp-light
    |
    |   +--rw twamp-light-session-sender {session-sender-light}?
    |   |
    |   |   +--rw sender-light-enable?    enable
    |   |   +--rw test-session* [session-id]
    |   |   |
    |   |   |   +--rw session-id                uint32
    |   |   |   +--rw test-session-enable?      enable
    |   |   |   +--rw number-of-packets?        union
    |   |   |   +--rw packet-padding-size?      uint32
    |   |   |   +--rw interval?                 uint32
    |   |   |   +--rw session-timeout?          uint32
    |   |   |   +--rw measurement-interval?     uint32
    |   |   |   +--rw repeat?                   union
    |   |   |   +--rw repeat-interval?          uint32
    |   |   |   +--rw dscp-value?               inet:dscp
    |   |   |   +--rw test-session-reflector-mode? session-reflector-mode
    |   |   |   +--rw sender-ip                 inet:ip-address
    |   |   |   +--rw sender-udp-port           inet:port-number
    |   |   |   +--rw reflector-ip              inet:ip-address
    |   |   |   +--rw reflector-udp-port        inet:port-number
    |   |   |   +--rw authentication-params! {twamp-light-authentication}?
    |   |   |   |   +--rw key-chain?    kc:key-chain-ref
    |   |   |   +--rw first-percentile?    percentile
    |   |   |   +--rw second-percentile?   percentile
    |   |   |   +--rw third-percentile?    percentile
    |   |   +--rw twamp-light-session-reflector {session-reflector-light}?
    |   |   |
    |   |   |   +--rw reflector-light-enable?    enable
    |   |   |   +--rw ref-wait?                  uint32
    |   |   |   +--rw reflector-light-mode-state? session-reflector-mode
    |   |   |   +--rw test-session* [session-id]
    |   |   |   |
    |   |   |   |   +--rw session-id                uint32
    |   |   |   |   +--rw dscp-handling-mode?        session-dscp-mode
    |   |   |   |   +--rw dscp-value?                inet:dscp
    |   |   |   |   +--rw sender-ip                  inet:ip-address
    |   |   |   |   +--rw sender-udp-port            inet:port-number
    |   |   |   |   +--rw reflector-ip                inet:ip-address
    |   |   |   |   +--rw reflector-udp-port          inet:port-number
    |   |   |   |   +--rw authentication-params! {twamp-light-authentication}?
    |   |   |   |   |   +--rw key-chain?    kc:key-chain-ref
    |   |   +--ro twamp-light-state
    |   |   |
    |   |   |   +--ro twamp-light-session-sender-state {session-sender-light}?
    |   |   |   |
    |   |   |   |   +--ro test-session-state* [session-id]
    |   |   |   |   |
    |   |   |   |   |   +--ro session-id                uint32
    |   |   |   |   |   +--ro sender-session-state?    enumeration
    |   |   |   |   +--ro current-stats

```

	+	ro	start-time		yang:date-and-time
	+	ro	packet-padding-size?		uint32
	+	ro	interval?		uint32
	+	ro	duplicate-packets?		uint32
	+	ro	reordered-packets?		uint32
	+	ro	sender-ip		inet:ip-address
	+	ro	sender-udp-port		inet:port-number
	+	ro	reflector-ip		inet:ip-address
	+	ro	reflector-udp-port		inet:port-number
	+	ro	dscp?		inet:dscp
	+	ro	sent-packets?		uint32
	+	ro	rcv-packets?		uint32
	+	ro	sent-packets-error?		uint32
	+	ro	rcv-packets-error?		uint32
	+	ro	last-sent-seq?		uint32
	+	ro	last-rcv-seq?		uint32
	+	ro	two-way-delay		
		+	ro	delay	
			+	ro	min? yang:gauge32
			+	ro	max? yang:gauge32
			+	ro	avg? yang:gauge32
		+	ro	delay-variation	
			+	ro	min? uint32
			+	ro	max? uint32
			+	ro	avg? uint32
	+	ro	one-way-delay-far-end		
		+	ro	delay	
			+	ro	min? yang:gauge32
			+	ro	max? yang:gauge32
			+	ro	avg? yang:gauge32
		+	ro	delay-variation	
			+	ro	min? uint32
			+	ro	max? uint32
			+	ro	avg? uint32
	+	ro	one-way-delay-near-end		
		+	ro	delay	
			+	ro	min? yang:gauge32
			+	ro	max? yang:gauge32
			+	ro	avg? yang:gauge32
		+	ro	delay-variation	
			+	ro	min? uint32
			+	ro	max? uint32
			+	ro	avg? uint32
	+	ro	low-percentile		
		+	ro	delay-percentile	
			+	ro	rtt-delay? percentile
			+	ro	near-end-delay? percentile
			+	ro	far-end-delay? percentile

```

+--ro jitter-percentile
  +--ro rtt-jitter?      percentile
  +--ro near-end-jitter? percentile
  +--ro far-end-jitter?  percentile
+--ro mid-percentile
  +--ro delay-percentile
    +--ro rtt-delay?      percentile
    +--ro near-end-delay? percentile
    +--ro far-end-delay?  percentile
  +--ro jitter-percentile
    +--ro rtt-jitter?      percentile
    +--ro near-end-jitter? percentile
    +--ro far-end-jitter?  percentile
+--ro high-percentile
  +--ro delay-percentile
    +--ro rtt-delay?      percentile
    +--ro near-end-delay? percentile
    +--ro far-end-delay?  percentile
  +--ro jitter-percentile
    +--ro rtt-jitter?      percentile
    +--ro near-end-jitter? percentile
    +--ro far-end-jitter?  percentile
+--ro two-way-loss
  +--ro loss-count?      int32
  +--ro loss-ratio?      percentage
  +--ro loss-burst-max?  int32
  +--ro loss-burst-min?  int32
  +--ro loss-burst-count? int32
+--ro one-way-loss-far-end
  +--ro loss-count?      int32
  +--ro loss-ratio?      percentage
  +--ro loss-burst-max?  int32
  +--ro loss-burst-min?  int32
  +--ro loss-burst-count? int32
+--ro one-way-loss-near-end
  +--ro loss-count?      int32
  +--ro loss-ratio?      percentage
  +--ro loss-burst-max?  int32
  +--ro loss-burst-min?  int32
  +--ro loss-burst-count? int32
+--ro history-stats* [id]
  +--ro id                uint32
  +--ro end-time          yang:date-and-time
  +--ro number-of-packets? uint32
  +--ro packet-padding-size? uint32
  +--ro interval?         uint32
  +--ro duplicate-packets? uint32
  +--ro reordered-packets? uint32

```

```

+--ro loss-packets?          uint32
+--ro sender-ip              inet:ip-address
+--ro sender-udp-port        inet:port-number
+--ro reflector-ip           inet:ip-address
+--ro reflector-udp-port     inet:port-number
+--ro dscp?                  inet:dscp
+--ro sent-packets?          uint32
+--ro rcv-packets?           uint32
+--ro sent-packets-error?    uint32
+--ro rcv-packets-error?     uint32
+--ro last-sent-seq?         uint32
+--ro last-rcv-seq?         uint32
+--ro two-way-delay
|   +--ro delay
|   |   +--ro min?          yang:gauge32
|   |   +--ro max?          yang:gauge32
|   |   +--ro avg?          yang:gauge32
|   +--ro delay-variation
|   |   +--ro min?          uint32
|   |   +--ro max?          uint32
|   |   +--ro avg?          uint32
+--ro one-way-delay-far-end
|   +--ro delay
|   |   +--ro min?          yang:gauge32
|   |   +--ro max?          yang:gauge32
|   |   +--ro avg?          yang:gauge32
|   +--ro delay-variation
|   |   +--ro min?          uint32
|   |   +--ro max?          uint32
|   |   +--ro avg?          uint32
+--ro one-way-delay-near-end
|   +--ro delay
|   |   +--ro min?          yang:gauge32
|   |   +--ro max?          yang:gauge32
|   |   +--ro avg?          yang:gauge32
|   +--ro delay-variation
|   |   +--ro min?          uint32
|   |   +--ro max?          uint32
|   |   +--ro avg?          uint32
+--ro twamp-light-session-refl-state {session-reflector-light}?
+--ro reflector-light-admin-status  boolean
+--ro test-session-state* [session-id]
+--ro session-id                    uint32
+--ro sent-packets?                  uint32
+--ro rcv-packets?                   uint32
+--ro sent-packets-error?            uint32
+--ro rcv-packets-error?             uint32
+--ro last-sent-seq?                 uint32

```



```

        +---ro last-rcv-seq?          uint32
        +---ro sender-ip              inet:ip-address
        +---ro sender-udp-port         inet:port-number
        +---ro reflector-ip           inet:ip-address
        +---ro reflector-udp-port      inet:port-number

rpcs:
  +---x twamp-sender-start
  |   +---w input
  |   +---w session-id      uint32
  +---x twamp-sender-stop
  |   +---w input
  |   +---w session-id      uint32
```

3.2. YANG Module

<CODE BEGINS> file "ietf-twamp-light@2017-06-13.yang"

```
module ietf-twamp-light {
  namespace "urn:ietf:params:xml:ns:yang:ietf-twamp-light";
  //namespace need to be assigned by IANA
  prefix "ietf-twamp-light";

  import ietf-inet-types {
    prefix inet;
  }
  import ietf-yang-types {
    prefix yang;
  }
  import ietf-key-chain {
    prefix kc;
  }

  organization
    "IETF IPPM (IP Performance Metrics) Working Group";

  contact
    "draft-mirsky-ippm-twamp-light-yang@tools.ietf.org";

  description "TWAMP Light Data Model";

  revision "2017-06-13" {
    description
      "08 version. Appendix I RFC 5357 is covered.";
    reference "RFC 5357";
  }
```

```
feature session-sender-light {
  description
    "This feature relates to the device functions as the
    TWAMP Light Session-Sender";
}

feature session-reflector-light {
  description
    "This feature relates to the device functions as the
    TWAMP Light Session-Reflector";
}

feature twamp-light-authentication {
  description
    "TWAMP Light authentication supported";
}

typedef enable {
  type boolean;
  description "enable";
}

typedef session-reflector-mode {
  type enumeration {
    enum stateful {
      description
        "When the Session-Reflector is stateful,
        i.e. is aware of TWAMP-Test session state.";
    }
    enum stateless {
      description
        "When the Session-Reflector is stateless,
        i.e. is not aware of the state of
        TWAMP-Test session.";
    }
  }
  description "State of the Session-Reflector";
}

typedef session-dscp-mode {
  type enumeration {
    enum copy-received-value {
      description
        "Use DSCP value copied from received
        TWAMP test packet of the test session.";
    }
    enum use-configured-value {
      description
```

```
        "Use DSCP value configured for this
        test session on the Session-Reflector.";
    }
}
description
    "DSCP handling mode by Session-Reflector.";
}

typedef percentage {
    type decimal64 {
        fraction-digits 5;
    }
    description "Percentage";
}

typedef percentile {
    type decimal64 {
        fraction-digits 2;
    }
    description
        "Percentile is a measure used in statistics
        indicating the value below which a given
        percentage of observations in a group of
        observations fall.";
}

grouping maintenance-statistics {
    description "Maintenance statistics grouping";
    leaf sent-packets {
        type uint32;
        description "Packets sent";
    }
    leaf rcv-packets {
        type uint32;
        description "Packets received";
    }
    leaf sent-packets-error {
        type uint32;
        description "Packets sent error";
    }
    leaf rcv-packets-error {
        type uint32;
        description "Packets received error";
    }
    leaf last-sent-seq {
        type uint32;
        description "Last sent sequence number";
    }
}
```

```
    leaf last-rcv-seq {
      type uint32;
      description "Last received sequence number";
    }
  }

  grouping twamp-session-percentile {
    description "Percentile grouping";
    leaf first-percentile {
      type percentile;
      default 95.00;
      description
        "First percentile to report";
    }
    leaf second-percentile {
      type percentile;
      default 99.00;
      description
        "Second percentile to report";
    }
    leaf third-percentile {
      type percentile;
      default 99.90;
      description
        "Third percentile to report";
    }
  }

  grouping delay-statistics {
    description "Delay statistics grouping";
    container delay {
      description "Packets transmitted delay";
      leaf min {
        type yang:gauge32;
        units microseconds;
        description
          "Min of Packets transmitted delay";
      }
      leaf max {
        type yang:gauge32;
        units microseconds;
        description
          "Max of Packets transmitted delay";
      }
      leaf avg {
        type yang:gauge32;
        units microseconds;
        description

```

```
        "Avg of Packets transmitted delay";
    }
}

container delay-variation {
    description
    "Packets transmitted delay variation";
    leaf min {
        type uint32;
        units microseconds;
        description
        "Min of Packets transmitted
        delay variation";
    }
    leaf max {
        type uint32;
        units microseconds;
        description
        "Max of Packets transmitted
        delay variation";
    }
    leaf avg {
        type uint32;
        units microseconds;
        description
        "Avg of Packets transmitted
        delay variation";
    }
}

grouping time-percentile-report {
    description "Delay percentile report grouping";
    container delay-percentile {
        description
        "Report round-trip, near- and far-end delay";
        leaf rtt-delay {
            type percentile;
            description
            "Percentile of round-trip delay";
        }
        leaf near-end-delay {
            type percentile;
            description
            "Percentile of near-end delay";
        }
        leaf far-end-delay {
            type percentile;
            description
            "Percentile of far-end delay";
        }
    }
}
```

```
        "Percentile of far-end delay";
    }
}
container jitter-percentile {
    description
    "Report round-trip, near- and far-end jitter";
    leaf rtt-jitter {
        type percentile;
        description
        "Percentile of round-trip jitter";
    }
    leaf near-end-jitter {
        type percentile;
        description
        "Percentile of near-end jitter";
    }
    leaf far-end-jitter {
        type percentile;
        description
        "Percentile of far-end jitter";
    }
}
}

grouping packet-loss-statistics {
    description
    "Grouping for Packet Loss statistics";
    leaf loss-count {
        type int32;
        description
        "Number of lost packets
        during the test interval.";
    }
    leaf loss-ratio {
        type percentage;
        description
        "Ratio of packets lost to packets
        sent during the test interval.";
    }
    leaf loss-burst-max {
        type int32;
        description
        "Maximum number of consecutively
        lost packets during the test interval.";
    }
    leaf loss-burst-min {
        type int32;
        description

```

```
        "Minimum number of consecutively
        lost packets during the test interval.";
    }

    leaf loss-burst-count {
        type int32;
        description
            "Number of occasions with packet
            loss during the test interval.";
    }
}

grouping session-light-parameters {
    description
        "Parameters common among
        Session-Sender and Session-Reflector";
    leaf sender-ip {
        type inet:ip-address;
        mandatory true;
        description "Sender IP address";
    }
    leaf sender-udp-port {
        type inet:port-number {
            range "49152..65535";
        }
        mandatory true;
        description "Sender UDP port number";
    }
    leaf reflector-ip {
        type inet:ip-address;
        mandatory true;
        description "Reflector IP address";
    }
    leaf reflector-udp-port {
        type inet:port-number {
            range "49152..65535";
        }
        mandatory true;
        description "Reflector UDP port number";
    }
}

grouping session-light-auth-params {
    description
        "Grouping for TWAMP Light authentication parameters";
    container authentication-params {
        if-feature twamp-light-authentication;
        presence "Enables TWAMP Light authentication";
        description

```

```
    "Parameters for TWAMP Light authentication";
    leaf key-chain {
        type kc:key-chain-ref;
        description "Name of key-chain";
    }
}

/*Configuration Data*/
container twamp-light {
    description
        "Top level container for TWAMP-Light configuration";

    container twamp-light-session-sender {
        if-feature session-sender-light;
        description "TWAMP-Light Session-Sender container";

        leaf sender-light-enable {
            type enable;
            default "true";
            description
                "Whether this network element is enabled to
                act as TWAMP-Light Sender";
        }

        list test-session {
            key "session-id";
            unique "sender-ip sender-udp-port reflector-ip"
                +" reflector-udp-port dscp-value";
            description
                "This structure is a container of test session
                managed objects";

            leaf session-id {
                type uint32;
                description "Session ID";
            }

            leaf test-session-enable {
                type enable;
                default "true";
                description
                    "Whether this TWAMP Test session is enabled";
            }

            leaf number-of-packets {
                type union {
                    type uint32 {
```



```
        range 1..4294967294 {
            description
                "The overall number of UDP test packet
                to be transmitted by the sender for this
                test session";
        }
    }
    type enumeration {
        enum forever {
            description
                "Indicates that the test session SHALL
                be run *forever*.";
        }
    }
}
default 10;
description
    "This value determines if the TWAMP-Test session is
    bound by number of test packets or not.";
}

leaf packet-padding-size {
    type uint32;
    default 27;
    description
        "Size of the Packet Padding. Suggested to run
        Path MTU Discovery to avoid packet fragmentation in
        IPv4 and packet blackholing in IPv6";
}

leaf interval {
    type uint32;
    units microseconds;
    description
        "Time interval between transmission of two
        consecutive packets in the test session in
        microseconds";
}

    leaf session-timeout {
        when "../number-of-packets != 'forever'" {
            description
                "Test session timeout only valid if the
                test mode is periodic.";
        }
        type uint32;
        units "seconds";
        default 900;
    }
```

```
description
"The timeout value for the Session-Sender to
collect outstanding reflected packets.";
}

leaf measurement-interval {
  when "../number-of-packets = 'forever'" {
    description
    "Valid only when the test to run forever,
    i.e. continuously.";
  }
  type uint32;
  units "seconds";
  default 60;
  description
  "Interval to calculate performance metric when
  the test mode is 'continuous'.";
}

leaf repeat {
  type union {
    type uint32 {
      range 0..4294967294;
    }
    type enumeration {
      enum forever {
        description
        "Indicates that the test session SHALL
        be repeated *forever* using the
        information in repeat-interval
        parameter, and SHALL NOT decrement
        the value.";
      }
    }
  }
  default 0;
  description
  "This value determines if the TWAMP-Test session must
  be repeated. When a test session has completed, the
  repeat parameter is checked. The default value
  of 0 indicates that the session MUST NOT be repeated.
  If the repeat value is 1 through 4,294,967,294
  then the test session SHALL be repeated using the
  information in repeat-interval parameter.
  The implementation MUST decrement the value of repeat
  after determining a repeated session is expected.";
}
```

```
    leaf repeat-interval {
        when "../repeat != '0'";
        type uint32;
        units seconds;
        default 0;
        description
            "This parameter determines the timing of repeated
            TWAMP-Test sessions when repeat is more than 0.";
    }

    leaf dscp-value {
        type inet:dscp;
        default 0;
        description
            "DSCP value to be set in the test packet.";
    }

    leaf test-session-reflector-mode {
        type session-reflector-mode;
        default "stateless";
        description
            "The mode of TWAMP-Reflector for the test session.";
    }

    uses session-light-parameters;
    uses session-light-auth-params;
    uses twamp-session-percentile;
}

container twamp-light-session-reflector {
    if-feature session-reflector-light;
    description
        "TWAMP-Light Session-Reflector container";
    leaf reflector-light-enable {
        type enable;
        default "true";
        description
            "Whether this network element is enabled to
            act as TWAMP-Light Reflector";
    }

    leaf ref-wait {
        type uint32 {
            range 1..604800;
        }
        units seconds;
        default 900;
    }
}
```

```
    description
    "REFWAIT(TWAMP test session timeout in seconds),
    the default value is 900";
}

leaf reflector-light-mode-state {
    type session-reflector-mode;
    default stateless;
    description
    "The state of the mode of the TWAMP-Light
    Session-Reflector";
}

list test-session {
    key "session-id";
    unique "sender-ip sender-udp-port reflector-ip"
    +" reflector-udp-port";
    description
    "This structure is a container of test session
    managed objects";

    leaf session-id {
        type uint32;
        description "Session ID";
    }

    leaf dscp-handling-mode {
        type session-dscp-mode;
        default copy-received-value;
        description
        "Session-Reflector handling of DSCP:
        - use value copied from received TWAMP-Test packet;
        - use value explicitly configured";
    }

    leaf dscp-value {
        when "../dscp-handling-mode = 'use-configured-value'";
        type inet:dscp;
        default 0;
        description
        "DSCP value to be set in the reflected packet
        if dscp-handling-mode is set to use-configured-value.";
    }

    uses session-light-parameters;
    uses session-light-auth-params;
}
}
```

```
    }

/*Operational state data nodes*/
container twamp-light-state{
  config "false";
  description
    "Top level container for TWAMP-Light state data";

  container twamp-light-session-sender-state {
    if-feature session-sender-light;
    description
      "Session-Sender container for state data";
    list test-session-state{
      key "session-id";
      description
        "This structure is a container of test session
        managed objects";

      leaf session-id {
        type uint32;
        description "Session ID";
      }

      leaf sender-session-state {
        type enumeration {
          enum active {
            description "Test session is active";
          }
          enum ready {
            description "Test session is idle";
          }
        }
        description
          "State of the particular TWAMP-Light test
          session at the sender";
      }
    }
  }

  container current-stats {
    description
      "This container contains the results for the current
      Measurement Interval in a Measurement session ";
    leaf start-time {
      type yang:date-and-time;
      mandatory true;
      description
        "The time that the current Measurement Interval started";
    }
  }
}
```

```
leaf packet-padding-size {
    type uint32;
    default 27;
    description
        "Size of the Packet Padding. Suggested to run
        Path MTU Discovery to avoid packet fragmentation
        in IPv4 and packet backholing in IPv6";
}

leaf interval {
    type uint32;
    units microseconds;
    description
        "Time interval between transmission of two
        consecutive packets in the test session";
}

leaf duplicate-packets {
    type uint32;
    description "Duplicate packets";
}

leaf reordered-packets {
    type uint32;
    description "Reordered packets";
}

uses session-light-parameters;
leaf dscp {
    type inet:dscp;
    description
        "The DSCP value that was placed in the header of
        TWAMP UDP test packets by the Session-Sender.";
}

uses maintenance-statistics;

container two-way-delay {
    description
        "two way delay result of the test session";
    uses delay-statistics;
}

container one-way-delay-far-end {
    description
        "one way delay far-end of the test session";
    uses delay-statistics;
}

container one-way-delay-near-end {
```

```
    description
    "one way delay near-end of the test session";
    uses delay-statistics;
}

container low-percentile {
    when "/twamp-light/twamp-light-session-sender/"
    +"test-session[session-id]/"
    +"first-percentile != '0.00'" {
        description
        "Only valid if the
        the first-percentile is not NULL";
    }
    description
    "Low percentile report";
    uses time-percentile-report;
}

container mid-percentile {
    when "/twamp-light/twamp-light-session-sender/"
    +"test-session[session-id]/"
    +"second-percentile != '0.00'" {
        description
        "Only valid if the
        the first-percentile is not NULL";
    }
    description
    "Mid percentile report";
    uses time-percentile-report;
}

container high-percentile {
    when "/twamp-light/twamp-light-session-sender/"
    +"test-session[session-id]/"
    +"third-percentile != '0.00'" {
        description
        "Only valid if the
        the first-percentile is not NULL";
    }
    description
    "High percentile report";
    uses time-percentile-report;
}

container two-way-loss {
    description
    "two way loss count and ratio result of
    the test session";
}
```

```
        uses packet-loss-statistics;
    }
    container one-way-loss-far-end {
        when "/twamp-light/twamp-light-session-sender/"
        +"test-session[session-id]/"
        +"test-session-reflector-mode = 'stateful'" {
            description
                "One-way statistic is only valid if the
                session-reflector is in stateful mode.";
        }
        description
            "one way loss count and ratio far-end of
            the test session";
        uses packet-loss-statistics;
    }
    container one-way-loss-near-end {
        when "/twamp-light/twamp-light-session-sender/"
        +"test-session[session-id]/"
        +"test-session-reflector-mode = 'stateful'" {
            description
                "One-way statistic is only valid if the
                session-reflector is in stateful mode.";
        }
        description
            "one way loss count and ratio near-end of
            the test session";
        uses packet-loss-statistics;
    }
}

list history-stats {
    key id;
    description
        "This container contains the results for the history
        Measurement Interval in a Measurement session ";
    leaf id {
        type uint32;
        description
            "The identifier for the Measurement Interval
            within this session";
    }
    leaf end-time {
        type yang:date-and-time;
        mandatory true;
        description
            "The time that the Measurement Interval ended";
    }
    leaf number-of-packets {
```



```
    type uint32;
    description
        "The overall number of UDP test packets to be
        transmitted by the sender for this test session";
}

leaf packet-padding-size {
    type uint32;
    default 27;
    description
        "Size of the Packet Padding. Suggested to run
        Path MTU Discovery to avoid packet fragmentation
        in IPv4 and packet blackholing in IPv6";
}

leaf interval {
    type uint32;
    units microseconds;
    description
        "Time interval between transmission of two
        consecutive packets in the test session";
}
leaf duplicate-packets {
    type uint32;
    description "Duplicate packets";
}
leaf reordered-packets {
    type uint32;
    description "Reordered packets";
}
leaf loss-packets {
    type uint32;
    description "Loss packets";
}

uses session-light-parameters;
leaf dscp {
    type inet:dscp;
    description
        "The DSCP value that was placed in the header of
        TWAMP UDP test packets by the Session-Sender.";
}
uses maintenance-statistics;

container two-way-delay{
    description
        "two way delay result of the test session";
    uses delay-statistics;
```

```
    }
    container one-way-delay-far-end{
        description
            "one way delay far end of the test session";
        uses delay-statistics;
    }
    container one-way-delay-near-end{
        description
            "one way delay near end of the test session";
        uses delay-statistics;
    }
}
}
}

container twamp-light-session-refl-state {
    if-feature session-reflector-light;
    description
        "TWAMP-Light Session-Reflector container for
        state data";
    leaf reflector-light-admin-status {
        type boolean;
        mandatory "true";
        description
            "Whether this network element is enabled to
            act as TWAMP-Light Reflector";
    }

    list test-session-state {
        key "session-id";
        description
            "This structure is a container of test session
            managed objects";

        leaf session-id {
            type uint32;
            description "Session ID";
        }

        uses maintenance-statistics;
        uses session-light-parameters;
    }
}

rpc twamp-sender-start {
    description
        "start the configured sender session";
```

```
    input {
      leaf session-id {
        type uint32;
        mandatory true;
        description
          "The session to be started";
      }
    }
  }
}

rpc twamp-sender-stop {
  description
    "stop the configured sender session";
  input {
    leaf session-id {
      type uint32;
      mandatory true;
      description
        "The session to be stopped";
    }
  }
}
}
```

<CODE ENDS>

4. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made.

URI: urn:ietf:params:xml:ns:yang:ietf-twamp-light

Registrant Contact: The IPPM WG of the IETF.

XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC6020].

name: ietf-twamp-light

namespace: urn:ietf:params:xml:ns:yang:ietf-twamp-light

prefix: twamp

reference: RFC XXXX

5. Security Considerations

The configuration, state, action data defined in this document may be accessed via the NETCONF protocol [RFC6241]. SSH [RFC6242] is mandatory secure transport that is the lowest NETCONF layer. The NETCONF access control model [RFC6536] provides means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

But, in general, this TWAMP Light YANG module does not change any underlying security issues that already may exist in [I-D.elteto-ippm-twamp-mib].

6. References

6.1. 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, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<http://www.rfc-editor.org/info/rfc3688>>.
- [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, <<http://www.rfc-editor.org/info/rfc5357>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<http://www.rfc-editor.org/info/rfc6020>>.
- [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, <<http://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, <<http://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, <<http://www.rfc-editor.org/info/rfc6536>>.
- [RFC7750] Hedin, J., Mirsky, G., and S. Baillargeon, "Differentiated Service Code Point and Explicit Congestion Notification Monitoring in the Two-Way Active Measurement Protocol (TWAMP)", RFC 7750, DOI 10.17487/RFC7750, February 2016, <<http://www.rfc-editor.org/info/rfc7750>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<http://www.rfc-editor.org/info/rfc8174>>.

6.2. Informative References

- [I-D.elteto-ippm-twamp-mib] Elteto, T. and G. Mirsky, "Two-Way Active Measurement Protocol (TWAMP) Management Information Base (MIB)", draft-elteto-ippm-twamp-mib-01 (work in progress), January 2014.
- [I-D.ietf-ippm-twamp-yang] Civil, R., Morton, A., Rahman, R., Jethanandani, M., and K. Pentikousis, "Two-Way Active Measurement Protocol (TWAMP) Data Model", draft-ietf-ippm-twamp-yang-03 (work in progress), February 2017.

Appendix A. Acknowledgements

TBD

Authors' Addresses

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

Xiao Min
ZTE Corp.

Email: xiao.min2@zte.com.cn

Adrian Pan
Ericsson

Email: adrian.pan@ericsson.com

Wei S Luo
Ericsson

Email: wei.s.luo@ericsson.com

Network Working Group
Internet-Draft
Updates: 5357 (if approved)
Intended status: Standards Track
Expires: December 2, 2017

G. Mirsky
ZTE Corp.
M. Perumal
Ericsson
R. Foote
Nokia
L. M. Contreras
Telefonica
L. Jalil
Verizon
May 31, 2017

UDP Port Allocation for the Receiver Port in Two-Way Active Measurement
Protocol (TWAMP)
draft-mirsky-ippm-twamp-refl-registered-port-03

Abstract

This document arguments and requests re-allocation of an UDP port number from the System Ports range for a Reflector in Two-Way Active Measurement Protocol (TWAMP). This document, if accepted, will be an update to the TWAMP Test protocol specified in RFC 5357.

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 December 2, 2017.

Copyright Notice

Copyright (c) 2017 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. Requirements Language	2
3. Impact to TWAMP-Control Protocol	3
4. Impact to TWAMP-Test Protocol	3
5. IANA Considerations	4
6. Security Considerations	4
7. Acknowledgments	5
8. Normative References	5
Authors' Addresses	5

1. Introduction

One particular compelling vision of the Two-Way Active Measurement Protocol (TWAMP) [RFC5357] is widespread deployment of open servers that would make IP Performance Metrics (IPPM) measurements a commonplace. This is complemented by the proliferation of the Internet of Things (IoT) devices, such as sensors, and the need for obtaining IPPM measurements from those devices by the service provider. IoT devices are often constrained by limited processing power and memory and benefit from TWAMP Light, as defined in Appendix I [RFC5357].

TWAMP Light provides a simple solution for devices to act as test points in the network, by avoiding the need for the TWAMP-Control protocol [RFC5357]. In the absence of TWAMP-Control, a registered (default) UDP port that can be used as the Receiver Port for TWAMP-Test will simplify configuration and management of the TWAMP-Light test sessions.

This document requests re-allocation of the UDP port number from the System Ports range [RFC6335] as Receiver Port for TWAMP-Test.

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 [RFC2119].

3. Impact to TWAMP-Control Protocol

Section 3.5 [RFC5357] describes in details the process of negotiating value of the Receiver Port. The Control-Client, acting on behalf of the Session-Sender, proposes the port number from the Dynamic Port range [RFC6335]:

"The Receiver Port is the desired UDP port to which TWAMP-Test packets will be sent by the Session-Sender (the port where the Session-Reflector is asked to receive test packets). The Receiver Port is also the UDP port from which TWAMP-Test packets will be sent by the Session-Reflector (the Session-Reflector will use the same UDP port to send and receive packets)."

But the proposed Receiver Port may be not available, e.g. being in use by other test session or another application. In this case:

"... the Server at the Session-Reflector MAY suggest an alternate and available port for this session in the Port field. The Session-Sender either accepts the alternate port, or composes a new Session-Request message with suitable parameters. Otherwise, the Server uses the Accept field to convey other forms of session rejection or failure to the Control Client and MUST NOT suggest an alternate port; in this case, the Port field MUST be set to zero."

The allocated TWAMP Receiver Port number Section 5 MAY be advertised by the Server. The Control Client that supports use of the allocated TWAMP Receiver Port MUST accept the port number advertised by the Server. If the Server does not support the allocated TWAMP Receiver Port, then it sends another Session-Request message with new parameters. Thus the deployment of the allocated TWAMP Receiver Port number is backward compatible with existing TWAMP-Control solutions that are based on [RFC5357]. At the same time, use of the UDP port number allocated from the User Port range [RFC6335] will help to avoid the situation when the Server finds the proposed port being already in use.

4. Impact to TWAMP-Test Protocol

TWAMP-Test may be used to measure IP performance metrics in an Equal Cost Multipath (ECMP) environment. Though algorithms to balance IP flows among available paths had not been standardized, the most common is the Five-tuple that uses destination IP address, source IP address, protocol type, destination port number, and source port number. To attempt to monitor different paths in ECMP network is sufficient to variate only one of five parameters, e.g. the source port number. Thus, there will be no negative impact on ability to have concurrent TWAMP test sessions between the same test points to

monitor different paths in the ECMP network when using the allocated UDP port number as the Receiver Port.

The allocation of the TWAMP Receiver Port from the User Port Range [RFC6335] benefits TWAMP Light mode of the TWAMP-Test. The allocated UDP port number Section 5 may be used as default value for the Receiver Port to simplify configuration and management of the TWAMP-Light test sessions.

5. IANA Considerations

The Service Name and Transport Protocol Port Number Registry defined in [RFC6335].

[RFC5357] has been allocated UDP port 862 for TWAMP-Control protocol. IANA is requested to re-assign UDP port 862 as follows:

Service Name	Port Number	Transport Protocol	Description	Semantics Definition	Reference
twamp-test	862	UDP	TWAMP-Test Receiver Port	Section 4	This document

Table 1: TWAMP Receiver Port

6. Security Considerations

The registered UDP port as the Receiver Port for TWAMP-Test may be used as target of denial-of-service (DoS) or used by man-in-the-middle (MitM) attack. To improve protection from the DoS following methods are recommended:

- o filtering access to the TWAMP Receiver Port by access list;
- o non-routable IPs outside of the domain for the TWAMP loopback.

MitM attack may try to modify the content of the TWAMP-Test packet thus altering measurement results. An implementation can use data consistency check to detect modification of data. In addition, it can use encryption of TWAMP-Test packets to prevent eavesdropping.

7. Acknowledgments

TBD

8. 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, <<http://www.rfc-editor.org/info/rfc2119>>.
- [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, <<http://www.rfc-editor.org/info/rfc5357>>.
- [RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, DOI 10.17487/RFC6335, August 2011, <<http://www.rfc-editor.org/info/rfc6335>>.

Authors' Addresses

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

Muthu Arul Mozhi Perumal
Ericsson
Ferns Icon
Doddanekundi, Mahadevapura
Bangalore, Karnataka 560037
India

Email: p.muthu.arul.mozhi@ericsson.com

Richard Foote
Nokia

Email: footer.foote@nokia.com

Luis M. Contreras
Telefonica

Email: luismiguel.contrerasmurillo@telefonica.com

Luay Jalil
Verizon

Email: luay.jalil@verizon.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: March 28, 2018

T. Mizrahi
Marvell
J. Fabini
Vienna University of Technology
A. Morton
AT&T Labs
September 24, 2017

Guidelines for Defining Packet Timestamps
draft-mizrahi-intarea-packet-timestamps-01

Abstract

This document specifies guidelines for defining binary packet timestamp formats in networking protocols at various layers. It also presents three recommended timestamp formats. The target audience of this memo includes network protocol designers. It is expected that a new network protocol that requires a packet timestamp will, in most cases, use one of the recommended timestamp formats. If none of the recommended formats fits the protocol requirements, the new protocol specification should specify the format of the packet timestamp according to the guidelines in this document.

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 March 28, 2018.

Copyright Notice

Copyright (c) 2017 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. Terminology	3
2.1. Requirements Language	3
2.2. Abbreviations	3
3. Packet Timestamp Format Specification	3
4. Recommended Timestamp Formats	4
4.1. Using a Recommended Timestamp Format	5
4.2. NTP Timestamp Formats	5
4.2.1. NTP 64-bit Timestamp Format	5
4.2.2. NTP 32-bit Timestamp Format	6
4.3. The PTP Truncated Timestamp Format	8
5. Timestamp Use Cases	9
5.1. Example 1	9
5.2. Example 2	10
6. Packet Timestamp Control Field	10
7. IANA Considerations	11
8. Security Considerations	11
9. Acknowledgments	12
10. References	12
10.1. Normative References	12
10.2. Informative References	12
Authors' Addresses	14

1. Introduction

Timestamps are widely used in network protocols for various purposes, including delay measurement, clock synchronization, and logging or reporting the time of an event.

Timestamps are represented in the RFC series in one of two forms: text-based timestamps, and packet timestamps. Text-based timestamps [RFC3339] are represented as user-friendly strings, and are widely used in the RFC series, for example in information objects and data models, e.g., [RFC5646], [RFC6991], and [RFC7493]. Packet timestamps, on the other hand, are represented by a compact binary field that has a fixed size, and are not intended to have a human-friendly format. Packet timestamps are also very common in the RFC

series, and are used for example for measuring delay and for synchronizing clocks, e.g., [RFC5905], [RFC4656], and [RFC1323].

This memo presents guidelines for defining a packet timestamp format in network protocols. Three recommended timestamp formats are presented. It is expected that a new network protocol that requires a packet timestamp will, in most cases, use one of the recommended timestamp formats. If none of the recommended formats fits the protocol requirements, the new protocol specification should specify the format of the packet timestamp according to the guidelines in this document.

2. Terminology

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 RFC 2119 [RFC2119].

2.2. Abbreviations

NTP Network Time Protocol [RFC5905]

PTP Precision Time Protocol [IEEE1588]

3. Packet Timestamp Format Specification

This memo recommends to use the timestamp formats defined in Section 4. In cases where these timestamp formats do not satisfy the protocol requirements, the timestamp specification should clearly state the reasons for defining a new format. Moreover, it is recommended to derive the new timestamp format from an existing timestamp format, either a timestamp format from this memo, or any other previously defined timestamp format.

This section defines a template for specifying packet timestamp formats. A timestamp format specification **MUST** include the following aspects:

Timestamp field format:

The format of the timestamp field consists of:

+ Size: The number of bits (or octets) used to represent the packet timestamp field.

+ Units: The units used to represent the timestamp.

If the timestamp is comprised of more than one field, the format of each field is specified.

Epoch:

The origin of the timescale used for the timestamp; the moment in time used as a reference for the timestamp value.

Resolution:

The timestamp resolution; the resolution is equal to the timestamp field unit. If the timestamp consists of two or more fields using different time units, then the resolution is the smallest time unit.

Wraparound:

The wraparound period of the timestamp; any further wraparound-related considerations should be described here.

4. Recommended Timestamp Formats

This memo recommends to use one of the timestamp formats specified below.

Clearly, different network protocols may have different requirements and constraints, and consequently may use different timestamp formats. The choice of the specific timestamp format for a given protocol may depend on a various factors. A few examples of factors that may affect the choice of the timestamp format:

- o Timestamp size: while some network protocols may allow a large timestamp fields, in other cases there may be constraints with respect to the timestamp size, affecting the choice of the timestamp format.
- o Resolution: the time resolution is another factor that may directly affect the selected timestamp format. Similarly, the wraparound periodicity of the timestamp may also affect the selected format.
- o Wraparound period: the length of the time interval in which the timestamp is unique may also be an important factor in choosing the timestamp format. Along with the timestamp resolution, these two factors determine the required number of bits in the timestamp.

- o Common format for multiple protocols: if there are two or more network protocols that use timestamps and are often used together in typical systems, using a common timestamp format should be preferred if possible. Specifically, if the network protocol that is being defined typically runs on a PC, then an NTP-based timestamp format may allow easier integration with an NTP-synchronized timer. In contrast, a protocol that is typically deployed on a hardware-based platform, may make better use of a PTP-based timestamp, allowing more efficient integration with a PTP-synchronized timer.

4.1. Using a Recommended Timestamp Format

A specification that uses one of the recommended timestamp formats should specify explicitly that this is a recommended timestamp format, and point to the relevant section in the current memo.

A specification that uses one of the recommended timestamp formats should also include a section on Synchronization Aspects. Any assumptions or requirements related to synchronization should be specified in this section. For example, the synchronization aspects may specify whether nodes populating the timestamps should be synchronized among themselves, and whether the timestamp is measured with respect to a central reference clock such as an NTP server. If time is assumed to be synchronized to a time standard such as UTC or TAI, it should be specified in this section. Further considerations may be discussed in this section, such as required accuracy, or leap second handling.

4.2. NTP Timestamp Formats

4.2.1. NTP 64-bit Timestamp Format

The Network Time Protocol (NTP) 64-bit timestamp format is defined in [RFC5905]. This timestamp format is used in several network protocols, including [RFC6374], [RFC4656], and [RFC5357]. Since this timestamp format is used in NTP, this timestamp format should be preferred in network protocols that are typically deployed in concert with NTP.

The format is presented in this section according to the template defined in Section 3.

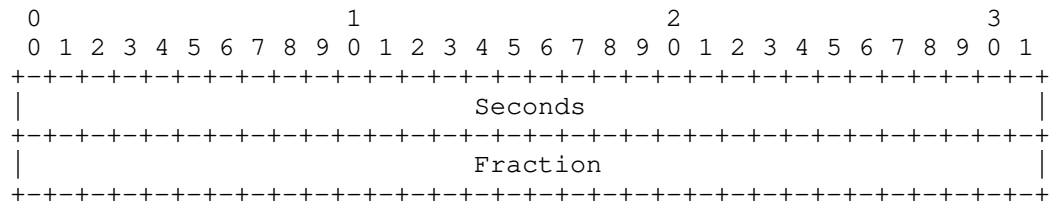


Figure 1: NTP [RFC5905] 64-bit Timestamp Format

Timestamp field format:

Seconds: specifies the integer portion of the number of seconds since the epoch.

+ Size: 32 bits.

+ Units: seconds.

Fraction: specifies the fractional portion of the number of seconds since the epoch.

+ Size: 32 bits.

+ Units: the unit is 2^{-32} seconds, which is roughly equal to 233 picoseconds.

Epoch:

The epoch is 1 January 1900 at 00:00 UTC.

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.

4.2.2. NTP 32-bit Timestamp Format

The Network Time Protocol (NTP) 32-bit timestamp format is defined in [RFC5905]. This timestamp format is used in [I-D.morton-ippm-mbm-registry]. This timestamp format should be preferred in network protocols that are typically deployed in concert with NTP. The 32-bit format can be used either when space

constraints do not allow the use of the 64-bit format, or when the 32-bit format satisfies the resolution and wraparound requirements.

The format is presented in this section according to the template defined in Section 3.

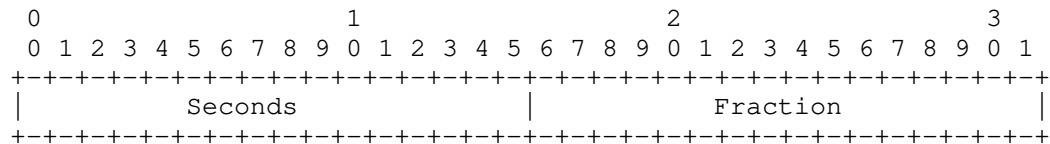


Figure 2: NTP [RFC5905] 32-bit Timestamp Format

Timestamp field format:

Seconds: specifies the integer portion of the number of seconds since the epoch.

+ Size: 16 bits.

+ Units: seconds.

Fraction: specifies the fractional portion of the number of seconds since the epoch.

+ Size: 16 bits.

+ Units: the unit is 2^{-16} seconds, which is roughly equal to 15.3 microseconds.

Epoch:

The epoch is 1 January 1900 at 00:00 UTC.

Resolution:

The resolution is 2^{-16} seconds.

Wraparound:

This time format wraps around every 2^{16} seconds, which is roughly 18 hours.

4.3. The PTP Truncated Timestamp Format

The Precision Time Protocol (PTP) [IEEE1588] 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. Since this timestamp format is similar to the one used in PTP, this timestamp format should be preferred in network protocols that are typically deployed in PTP-capable devices.

The PTP truncated timestamp format is used in several protocols, such as [RFC6374], [RFC7456], [RFC8186] and [ITU-T-Y.1731].

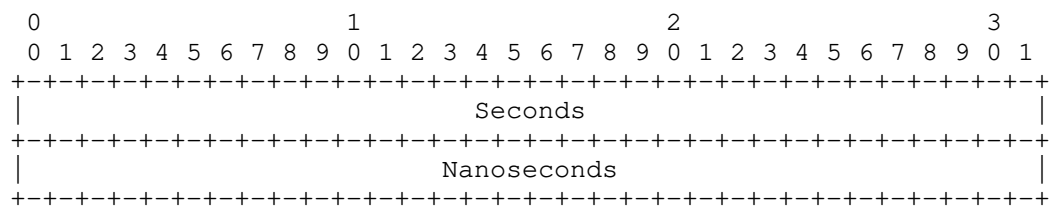


Figure 3: PTP [IEEE1588] Truncated Timestamp Format

Timestamp field format:

Seconds: specifies the integer portion of the number of seconds since the epoch.

+ Size: 32 bits.

+ Units: seconds.

Nanoseconds: specifies the fractional portion of the number of seconds since the epoch.

+ Size: 32 bits.

+ Units: nanoseconds. The value of this field is in the range 0 to $(10^9)-1$.

Epoch:

The PTP [IEEE1588] epoch is 1 January 1970 00:00:00 TAI, which is 31 December 1969 23:59:51.999918 UTC.

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.

5. Timestamp Use Cases

Packet timestamps are used in various network protocols. Typical applications of packet timestamps include delay measurement, clock synchronization, and others. The following table presents a (non-exhaustive) list of protocols that use packet timestamps, and the timestamp formats used in each of these protocols.

Protocol	Recommended formats			Other format
	NTP 64-bit	NTP 32-bit	PTP Trunc.	
NTP [RFC5905]	+			
OWAMP [RFC4656]	+			
TWAMP [RFC5357] TWAMP [RFC8186]	+		+	
TRILL [RFC7456]			+	
MPLS [RFC6374]			+	
TCP [RFC1323]				+
RTP [RFC3550]	+			+

Figure 4: Protocols that use Packet Timestamps

The rest of this section presents two hypothetical examples of network protocol specifications that use one of the recommended timestamp formats. The examples include the text that specifies the information related to the timestamp format.

5.1. Example 1

Timestamp:

The timestamp format used in this specification is the NTP [RFC5905] 64-bit format, as specified in Section 4.2.1 of [I-D.mizrahi-intarea-packet-timestamps].

Synchronization aspects:

It is assumed that nodes that run this protocol are synchronized to UTC using a synchronization mechanism that is outside the scope of this document. In typical deployments this protocol will be run on a machine that uses NTP [RFC5905] for synchronization. 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.

5.2. Example 2

Timestamp:

The timestamp format used in this specification is the PTP [IEEE1588] Truncated format, as specified in Section 4.2.3 of [I-D.mizrahi-intarea-packet-timestamps].

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 [IEEE1588] 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. Packet Timestamp Control Field

In some cases it is desirable to have a control field that includes information about the timestamp format. This section defines a recommended format of a timestamp-related control field that is intended for network protocols that require such timestamp-related control information.

The recommended control field includes the following sub-fields:

- o Timestamp format.
- o Precision - the resolution or granularity of the system clock.
- o Epoch.

- o Era - the number of times the time has wrapped around since the epoch.

7. IANA Considerations

This memo includes no request to IANA.

8. Security Considerations

A network protocol that uses a packet timestamp MUST specify the security considerations that result from using the timestamp. This section provides an overview of some of the common security considerations of using timestamps.

Any metadata that is attached to control or data packets, and specifically packet timestamps, can facilitate network reconnaissance; by passively eavesdropping to timestamped packets an attacker can gather information about the network performance, and about the level of synchronization between nodes.

Timestamps can be spoofed or modified by on-path attackers, thus attacking the application that uses the timestamps. For example, if timestamps are used in a delay measurement protocol, an attacker can modify en route timestamps in a way that manipulates the measurement results. Integrity protection mechanisms, such as Hashed Message Authentication Codes (HMAC), can mitigate such attacks. The specification of an integrity protection mechanism is outside the scope of this document, as typically integrity protection will be defined on a per-network-protocol basis, and not specifically for the timestamp field.

Another potential threat that can have a similar impact is delay attacks. An attacker can maliciously delay some or all of the en route messages, with the same harmful implications as described in the previous paragraph. Mitigating delay attacks is a significant challenge; in contrast to spoofing and modification attacks, the delay attack cannot be prevented by cryptographic integrity protection mechanisms. In some cases delay attacks can be mitigated by sending the timestamped information through multiple paths, allowing to detect and to be resilient to an attacker that has access to one of the paths.

In many cases timestamping relies on an underlying synchronization mechanism. Thus, any attack that compromises the synchronization mechanism can also compromise protocols that use timestamping. Attacks on time protocols are discussed in detail in [RFC7384].

9. Acknowledgments

The authors thank Yaakov Stein and other members of the TICTOC and NTP working groups for many helpful comments.

10. References

10.1. 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>>.

10.2. Informative References

- [I-D.mizrahi-intarea-packet-timestamps]
Mizrahi, T., Fabini, J., and A. Morton, "Guidelines for Defining Packet Timestamps", draft-mizrahi-intarea-packet-timestamps-00 (work in progress), June 2017.
- [I-D.morton-ippm-mbm-registry]
Morton, A. and M. Mathis, "Initial Performance Metric Registry Entries Part 2: MBM", draft-morton-ippm-mbm-registry-01 (work in progress), March 2017.
- [IEEE1588]
IEEE, "IEEE 1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Version 2", 2008.
- [ITU-T-Y.1731]
ITU-T, "OAM functions and mechanisms for Ethernet based Networks", 2013.
- [RFC1323] Jacobson, V., Braden, R., and D. Borman, "TCP Extensions for High Performance", RFC 1323, DOI 10.17487/RFC1323, May 1992, <<https://www.rfc-editor.org/info/rfc1323>>.
- [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>>.
- [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>>.

- [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>>.
- [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>>.
- [RFC5646] Phillips, A., Ed. and M. Davis, Ed., "Tags for Identifying Languages", BCP 47, RFC 5646, DOI 10.17487/RFC5646, September 2009, <<https://www.rfc-editor.org/info/rfc5646>>.
- [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>>.
- [RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, DOI 10.17487/RFC6374, September 2011, <<https://www.rfc-editor.org/info/rfc6374>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [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>>.
- [RFC7456] Mizrahi, T., Senevirathne, T., Salam, S., Kumar, D., and D. Eastlake 3rd, "Loss and Delay Measurement in Transparent Interconnection of Lots of Links (TRILL)", RFC 7456, DOI 10.17487/RFC7456, March 2015, <<https://www.rfc-editor.org/info/rfc7456>>.
- [RFC7493] Bray, T., Ed., "The I-JSON Message Format", RFC 7493, DOI 10.17487/RFC7493, March 2015, <<https://www.rfc-editor.org/info/rfc7493>>.
- [RFC8186] Mirsky, G. and I. Meilik, "Support of the IEEE 1588 Timestamp Format in a Two-Way Active Measurement Protocol (TWAMP)", RFC 8186, DOI 10.17487/RFC8186, June 2017, <<https://www.rfc-editor.org/info/rfc8186>>.

Authors' Addresses

Tal Mizrahi
Marvell
6 Hamada st.
Yokneam
Israel

Email: talmi@marvell.com

Joachim Fabini
Vienna University of Technology
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/>

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
URI: <http://home.comcast.net/~acmacm/>

Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 31, 2017

T. Mizrahi
C. Arad
Marvell
G. Fioccola
M. Cociglio
Telecom Italia
M. Chen
L. Zheng
Huawei Technologies
G. Mirsky
ZTE Corp.
June 29, 2017

Compact Alternate Marking Methods for Passive Performance Monitoring
draft-mizrahi-ippm-multiplexed-alternate-marking-02

Abstract

This memo introduces new alternate marking methods that require a compact overhead of either a single bit per packet, or zero bits per packet. This memo also presents a summary of alternate marking methods, and discusses the tradeoffs among them. The target audience of this document is network protocol designers; this document is intended to help protocol designers choose the best alternate marking method(s) based on the protocol's constraints and requirements.

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 December 31, 2017.

Copyright Notice

Copyright (c) 2017 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
1.1. Background	3
1.2. The Scope of This Document	4
2. Terminology	5
2.1. Requirements Language	5
2.2. Abbreviations	5
3. Marking Abstractions	5
4. Double Marking	7
5. Single-bit Marking	8
5.1. Single Marking Using the First Packet	8
5.2. Single Marking using the Mean Delay	8
5.3. Alternate Marking using a Multiplexed Marking Bit	8
5.3.1. Overview	8
5.3.2. Timing and Synchronization Aspects	9
5.4. Pulse Marking	11
6. Zero-bit Marking	12
6.1. Hash-based Sampling	12
6.2. Hashed Pulse Marking	13
6.3. Hashed Double Marking	13
6.4. Mixed Hashed Marking	14
7. Summary of Marking Methods	14
8. Alternate Marking using Reserved Values	17
9. IANA Considerations	18
10. Security Considerations	18
11. References	18
11.1. Normative References	18
11.2. Informative References	19
Authors' Addresses	20

1. Introduction

1.1. Background

Alternate marking, defined in [I-D.ietf-ippm-alt-mark], is a method for measuring packet loss, packet delay, and packet delay variation. Typical delay measurement protocols require the two measurement points (MPs) to exchange timestamped test packets. In contrast, the alternate marking method does not require control packets to be exchanged. Instead, every data packet carries a color indicator, which divides the traffic into consecutive blocks of packets.

The color value is toggled periodically, as illustrated in Figure 1.

A: packet with color 0
B: packet with color 1

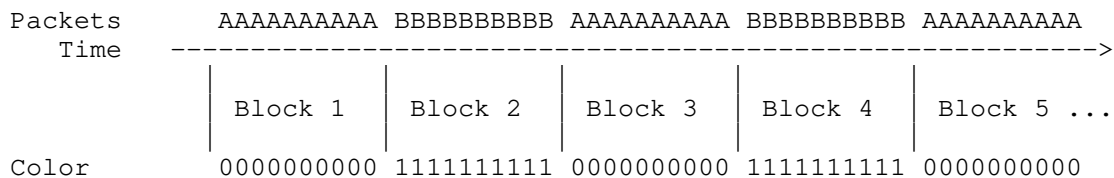


Figure 1: Alternate marking: packets are monitored on a per-color basis.

Alternate marking is used between two MPs, the initiating MP, and the monitoring MP. The initiating MP incorporates the marking field into en-route packets, allowing the monitoring MP to use the marking field in order to bind each packet to the corresponding block.

Each of the MPs maintains two counters, one per color. At the end of each block the counter values can be collected by a central management system, and analyzed; the packet loss can be computed by comparing the counter values of the two MPs.

When using alternate marking delay measurement can be performed in one of three ways (as per [I-D.ietf-ippm-alt-mark]):

- o Single marking using the first packet: in this method each packet uses a single marking bit, used as a color indicator. The first packet of each block is used by both MPs as a reference for delay measurement. The timestamp of this packet is measured by the two measurement points, and can be collected by the management system from each of the measurement points, which can compute the path delay by comparing the two timestamps. The drawback of this

approach is that it is not accurate when packets arrive out-of-order, as the two MPs may have a different view of which packet was the first in the block.

- o Single marking using the mean delay: as in the previous method, each packet uses a single marking method, indicating the color. Each of the MPs computes the average packet timestamp of each block. The management system can then compute the delay by comparing the average times of the two MPs. The drawback of this approach is that it may be computationally heavy, or difficult to implement at the data plane.
- o Double marking: each packet uses two marking bits. One bit is used as a color indicator, and one is used as a timestamping indicator. This method resolves the drawbacks raised for the two previous methods, at the expense of an extra bit in the packet header.

The double marking method is the most straightforward approach. It allows for accurate measurement without incurring expensive computational load. However, in some cases allocating two bits for passive measurement is not possible. For example, if alternate marking is implemented over IPv4, allocating 2 marking bits in the IPv4 header is challenging, as every bit in the 20-octet header is costly; one of the possible approaches discussed in [I-D.ietf-ippm-alt-mark] is to reserve one or two bits from the DSCP field for remarking. In this case every marking bit comes at the expense of reducing the DSCP range by a factor of two.

1.2. The Scope of This Document

This memo extends the marking methods of [I-D.ietf-ippm-alt-mark], and introduces methods that require a single marking bit, or zero marking bits.

Two single-bit marking methods are proposed, multiplexed marking and pulse marking. In multiplexed marking the color indicator and the timestamp indicator are multiplexed into a single bit, providing the advantages of the double marking method while using a single bit in the packet header. In pulse marking both delay and loss measurement are triggered by a 'pulse' value in a single marking field.

This document also discusses zero-bit marking methods that leverage well-known hash-based selection [RFC5475] approaches.

Alternate marking is discussed in this memo as a single-bit or a two-bit marking method. However, these methods can similarly be applied to larger fields, such as an IPv6 Flow Label or an MPLS Label;

single-bit marking can be applied using two reserved values, and two-bit marking can be applied using four reserved values. Marking based on reserved values is further discussed in this document, including its application to MPLS and IPv6.

Finally, this memo summarizes the alternate marking methods, and discusses the tradeoffs among them. It is expected that different network protocols will have different constraints, and therefore may choose to use different alternate marking methods. In some cases it may be preferable to support more than one marking method; in this case the particular marking method may be signaled through the control plane.

2. Terminology

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 RFC 2119 [RFC2119].

2.2. Abbreviations

The following abbreviations are used in this document:

DSCP	Differentiated Services Code Point
DM	Delay Measurement
LM	Loss Measurement
LSP	Label Switched Path
MP	Measurement Point
MPLS	Multiprotocol Label Switching
SFL	Synonymous Flow Label [I-D.bryant-mpls-sfl-framework]

3. Marking Abstractions

The marking methods that were discussed in Section 1, as well as the methods introduced in this document, use two basic abstractions, pulse detection, and step detection.

The common thread along the various marking methods is that one or two marking bits are used by the MPs to signal a measurement event.

The value of the marking bit indicates when the event takes place, in one of two ways:

Pulse	An event is detected when the value of the marking bit is toggled in a single packet.
-------	---

Step	An event is detected when the value of the marking bit is toggled, and remains at the new value.
------	--

The double marking method (Section 1) uses pulse-based detection for DM, and step-based detection for LM.

Pulse-based detection affects the processing of a single packet; the packet that indicates the pulse is processed differently than the packets around it. For example, in the double marking method, the marked packet is timestamped for DM, without affecting the packets before or after it. Note that if the marked packet is lost, no pulse is detected, yielding a missing measurement (see Figure 2).

P: indicates a packet

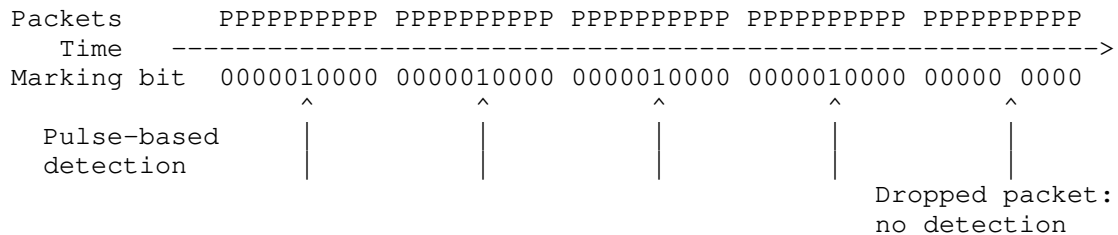


Figure 2: Pulse-based Detection.

In step-based detection the event is detected by observing a value change in stream of packets. Specifically, when the step approach is used for LM (as in the double marking method), two counters are used per flow; each MP decides which counter to use based on the value of the marking bit. Thus, the step-based approach allows accurate counting even when packets arrive out-of-order (see Figure 3). When the step approach is used for DM (e.g., single marking using the first packet), out-of-order causes the delay measurement to be false, without any indication to the management system.

P: indicates a packet

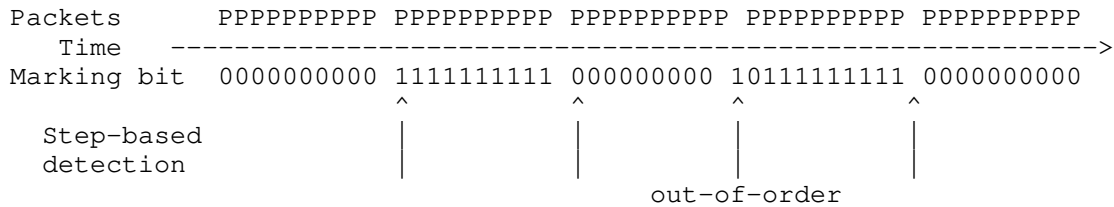


Figure 3: Step-based Detection.

4. Double Marking

The two-bit marking method of [I-D.ietf-ippm-alt-mark] uses two marking bits: a color indicator, and a delay measurement indicator. The color bit is used for step-based LM, while the delay bit is used as a pulse-based DM trigger. This double marking approach is the most straightforward of the approaches discussed in this memo, as it allows accurate measurement, it is resilient to out-of-order delivery, and is relatively simple to implement. The main drawback is that it requires two bits, which are not always available.

Figure 4 illustrates the double marking method: each block of packets includes a packet that is marked for timestamping, and therefore has its delay bit set.

```
A: packet with color 0
B: packet with color 1
```

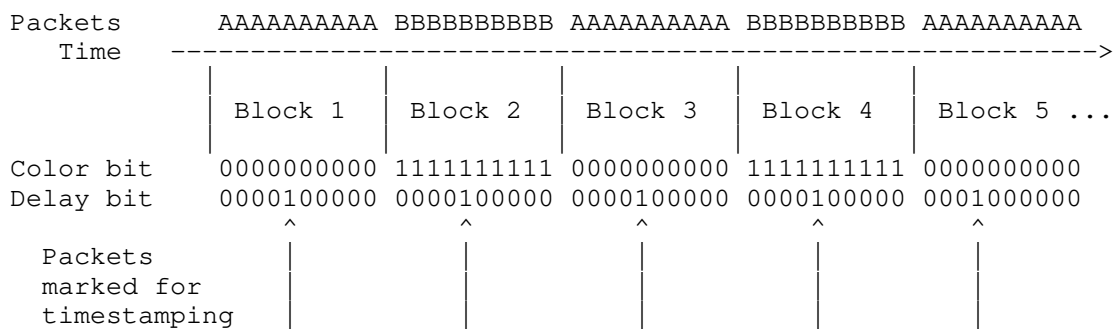


Figure 4: The double marking method.

5. Single-bit Marking

5.1. Single Marking Using the First Packet

This method uses a single marking bit that indicates the color, as described in [I-D.ietf-ippm-alt-mark]. Both LM and DM are implemented using a step-based approach; LM is implemented using two color-based counters per flow. The first packet of every period is used by the two MPs as the reference for measuring the delay. As denoted above, the delay computed in this method may be erroneous when packets are delivered out-of-order.

A: packet with color 0
B: packet with color 1

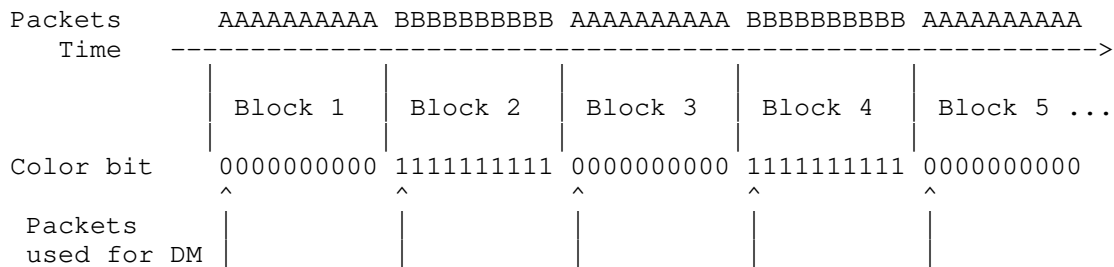


Figure 5: Single marking using the first packet of the block.

5.2. Single Marking using the Mean Delay

As in the first-packet approach, in the mean delay approach ([I-D.ietf-ippm-alt-mark]) a single marking bit is used to indicate the color, enabling step-based loss measurement. Delay is measured in each period by averaging the measured delay over all the packets in the period. As discussed above, this approach is not sensitive to out-of-order delivery, but may be heavy from a computational perspective.

5.3. Alternate Marking using a Multiplexed Marking Bit

5.3.1. Overview

This section introduces a method that uses a single marking bit that serves two purposes: a color indicator, and a timestamp indicator. The double marking method that was discussed in the previous section uses two 1-bit values: a color indicator C, and a timestamp indicator

T. The multiplexed marking bit, denoted by M, is an exclusive or between these two values: $M = C \text{ XOR } T$.

An example of the use of the multiplexed marking bit is depicted in Figure 6. The example considers two routers, R1 and R2, that use the multiplexed bit method to measure traffic from R1 to R2. In each block R1 designates one of the packets for delay measurement. In each of these designated packets the value of the multiplexed bit is reversed compared to the other packets in the same block, allowing R2 to distinguish the designated packets from the other packets.

A: packet with color 0

B: packet with color 1

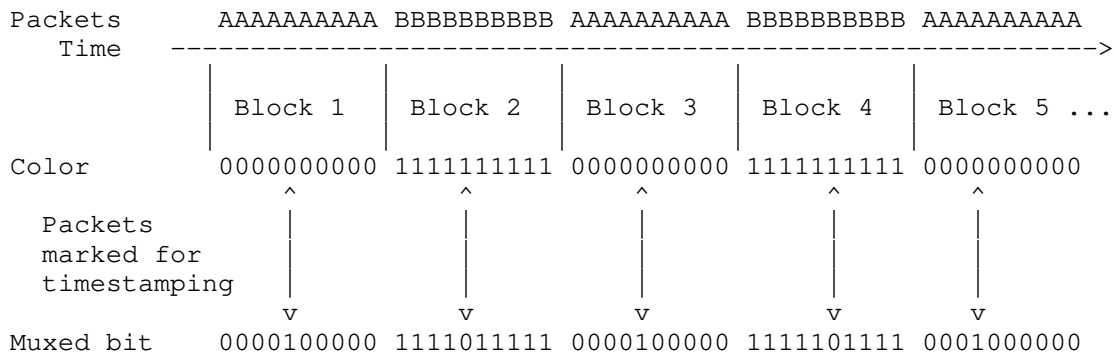


Figure 6: Alternate marking with multiplexed bit.

5.3.2. Timing and Synchronization Aspects

It is assumed that all MPs are synchronized to a common reference time with an accuracy of $\pm A/2$. Thus, the difference between the clock values of any two MPs is bounded by A. Clocks can be synchronized for example using NTP [RFC5905], PTP [IEEE1588], or by other means. The common reference time is used for dividing the time domain into equal-sized measurement periods, such that all packets forwarded during a measurement period have the same color, and consecutive periods have alternating colors.

The single marking bit incorporates two multiplexed values. From the monitoring MP's perspective, the two values are Time-Division Multiplexed (TDM), as depicted in Figure 7. It is assumed that the start time of every measurement period is known to both the initiating MP and the monitoring MP. If the measurement period is L, then during the first and the last L/4 time units of each block the

marking bit is interpreted by the monitoring MP as a color indicator. During the middle part of the block, the marking bit is interpreted as a timestamp indicator; if the value of this bit is different than the color value, the corresponding packet is used as a reference for delay measurement.

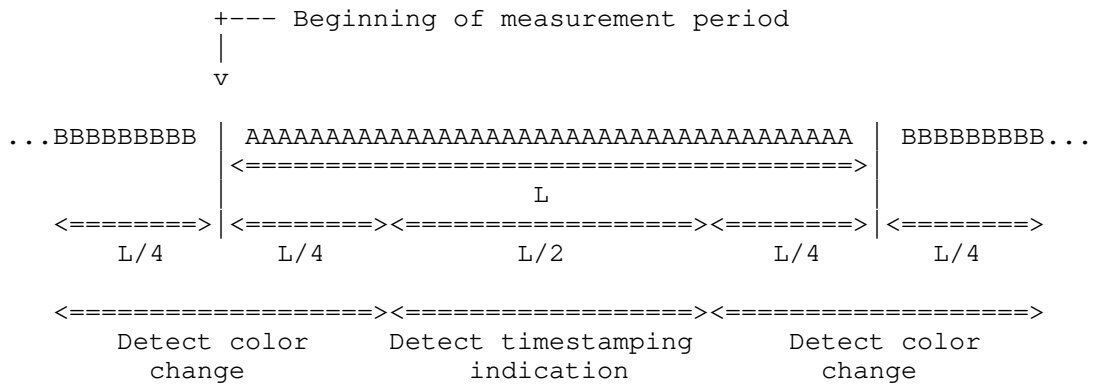


Figure 7: Multiplexed marking field interpretation at the receiving measurement point.

In order to prevent ambiguity in the receiver's interpretation of the marking field, the initiating MP is permitted to set the timestamp indication only during a specific interval, as depicted in Figure 8. Since the receiver is willing to receive the timestamp indication during the middle L/2 time units of the block, the sender refrains from sending the timestamp indication during a guardband interval of d time units at the beginning and end of the L/2-period.

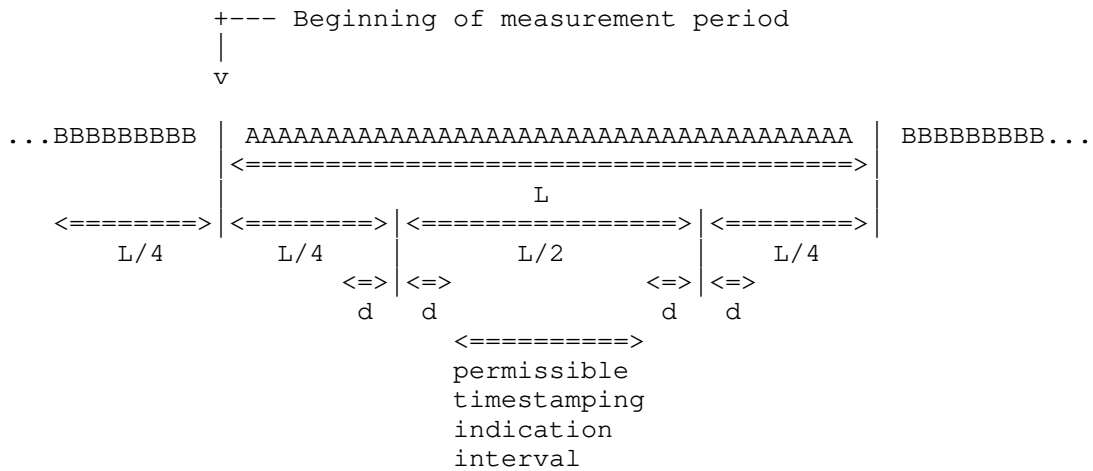


Figure 8: A time domain view.

The guardband d is given by $d = A + D_{\max} - D_{\min}$, where A is the clock accuracy, D_{\max} is an upper bound on the network delay between the MPs, and D_{\min} is a lower bound on the delay. It is straightforward from Figure 8 that $d < L/4$ must be satisfied. The latter implies a minimal requirement on the synchronization accuracy.

All MPs must be synchronized to the same reference time with an accuracy of $\pm L/8$. Depending on the system topology, in some systems the accuracy requirement will be even more stringent, subject to $d < L/4$. Note that the accuracy requirement of the conventional alternate marking method [I-D.ietf-ippm-alt-mark] is $\pm L/2$, while the multiplexed marking method requires an accuracy of $\pm L/8$.

Note that we assume that the middle $L/2$ -period is designated as the timestamp indication period, allowing a sufficiently long guardband between the transitions. However, a system may be configured to use a longer timestamp indication period or a shorter one, if it is guaranteed that the synchronization accuracy meets the guardband requirements (i.e., the constraints on d).

5.4. Pulse Marking

Pulse marking uses a single marking bit that is used as a trigger for both LM and DM. In this method the two MPs maintain a single per-flow counter for LM, in contrast to the color-based methods which require two counters per flow. In each block one of the packets is marked. The marked packet triggers two actions in each of MPs:

- o The timestamp is captured for DM.

- o The value of the counter is captured for LM.

In each period, each of the MPs exports the timestamp and counter-stamp to the management system, which can then compute the loss and delay in that period. It should be noted that as in [I-D.ietf-ippm-alt-mark], if the length of the measurement period is L time units, then all network devices must be synchronized to the same clock reference with an accuracy of $\pm L/2$ time units.

The pulse marking approach is illustrated in Figure 9. Since both LM and DM use a pulse-based trigger, if the marked packet is lost then no measurement is available in this period. Moreover, the LM accuracy may be affected by out-of-order delivery.

P: packet - all packets have the same color

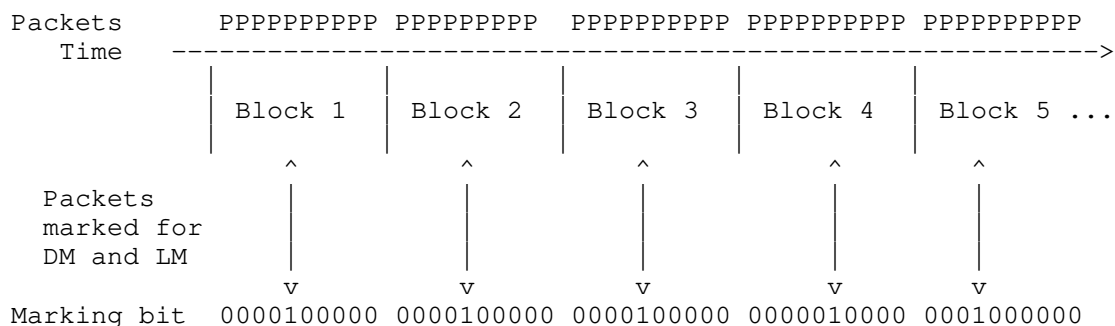


Figure 9: Pulse marking method.

6. Zero-bit Marking

6.1. Hash-based Sampling

Hash based selection [RFC5475] is a well-known method for sampling a subset of packets. As defined in [RFC5475]:

A Hash Function h maps the Packet Content c , or some portion of it, onto a Hash Range R . The packet is selected if $h(c)$ is an element of S , which is a subset of R called the Hash Selection Range.

Hash-based selection can be leveraged as a marking method, allowing a zero-bit marking approach. Specifically, the pulse and step abstractions can be implemented using hashed selection:

- o Hashed pulse-based trigger: in this approach, a packet is selected if $h(c)$ is an element of S , which is a strict subset of the hash range R . When $|S| \ll |R|$, the average sampling period is long, reducing the probability of ambiguity between consecutive packets. $|S|$ and $|R|$ denote the number of elements in S and R , respectively.
- o Hashed step-based trigger: the hash values of a given traffic flow are said to be monotonically increasing if for two packets p_1 and p_2 , if p_1 is sent before p_2 then $h(p_1) \leq h(p_2)$. If it is guaranteed that the hash values of a flow are monotonically increasing, then a step-based approach can be used on the range R . For example, in an IPv4 flow the Identification field can be used as the hash value of each packet. Since the Identification field is monotonically increasing, the step-based trigger can be implemented using consecutive ranges of the Identification value. For example, the fourth bit of the Identification field is toggled every 8 packets. Thus, a possible hash function simply takes the fourth bit of the Identification field as the hash value. This hash value is toggled every 8 packets, simulating the alternate marking behavior of Section 4.

Note that as opposed to the double marking and single marking methods, hashed sampling is not based on fixed time intervals, as the duration between sampled packets depends only on the hash value.

It is also important to note that all methods that use hash-based marking require the hash function and the set S to be configured consistently across the MPs.

6.2. Hashed Pulse Marking

In this approach a hash is computed over the packet content, and both LM and DM are triggered based on the pulse-based trigger (Section 6.1). A pulse is detected when the hash value $h(c)$ is equal to one of the values in S . The hash function h and the set S determine the probability (or frequency) of the pulse event.

6.3. Hashed Double Marking

As in the previous approach, hashed double marking also uses a hash that is computed over the packet content. In this approach DM is performed using a pulse-based trigger, whereas the LM trigger is step-based (Section 6.1). The main drawback of this method is that the step-based trigger is possible only under the assumption that the hash function is monotonically increasing, which is not necessarily possible in all cases. Specifically, a measured flow is not necessarily an IPv4 5-tuple. For example, a measured flow may

include multiple IPv4 5-tuple flows, and in this case the Identification field is not monotonically increasing.

6.4. Mixed Hashed Marking

Mixed hashed marking combines the single marking approach with hash-based sampling. A single marking bit is used in the packet header as a color indicator, while a hash-based pulse is used to trigger DM. Although this method requires a single bit, it is described in this section as it is closely related to the other hash-based methods that require zero marking bits.

7. Summary of Marking Methods

This section summarizes the marking methods described in this memo. Each row in the table of Figure 10 represents a marking method. For each method the table specifies the number of bits required in the header, the number of counters per flow for LM, the methods used for LM and DM (pulse or step), and also the resilience to disturbances.

Method	# of bits	# of counters	LM Method	DM Method	Resilience to Reordering		Resilience to packet drops	
					LM	DM	LM	DM
Double marking	2	2	Step	Pulse	+	+	+	-
Single marking - 1st packet	1	2	Step	Step	+	--	+	--
Single marking - mean delay	1	2	Step	Mean	+	+	+	-
Multiplexed marking	1	2	Step	Pulse	+	+	+	-
Pulse marking	1	1	Pulse	Pulse	--	+	-	-
Hashed pulse marking	0	1	Hashed pulse	Hashed pulse	--	+	-	-
Hashed double marking	0	2	Hashed step*	Hashed pulse	+	+	+	-
Mixed hashed marking	1	2	Step	Hashed pulse	+	+	+	-

+ Accurate measurement.

- No measurement in case of disturbance (detectable).

-- False measurement in case of disturbance (not detectable).

* Hashed step works only when the hash is monotonically increasing.

Figure 10: Summary of Marking Methods

In the context of this comparison two possible disturbances are considered: out-of-order delivery, and packet drops. Generally speaking, pulse based methods are sensitive to packet drops, since if the marked packet is dropped no measurement is recorded in the current period. Notably, a missing measurement is detectable by the management system, and is not as severe as a false measurement. Step-based triggers are generally resilient to out-of-order delivery for LM, but are not resilient to out-of-order delivery for DM. Notably, a step-based trigger may yield a false delay measurement when packets are delivered out-of-order, and this inaccuracy is not detectable.

As mentioned above, the double marking method is the most straightforward approach, and is resilient to most of the disturbances that were analyzed. Its obvious drawback is that it requires two marking bits.

Several single marking methods are discussed in this memo. In this case there is no clear verdict which method is the optimal one. The first packet method may be simple to implement, but may present erroneous delay measurements in case of dropped or reordered packets. Arguably, the mean delay approach and the multiplexed approach may be more difficult to implement (depending on the underlying platform), but are more resilient to the disturbances that were considered here. Note that the computational complexity of the mean delay approach can be reduced by combining it with a hashed approach, i.e., by computing the mean delay over a hash-based subset of the packets. The pulse marking method requires only a single counter per flow, while the other methods require two counters per flow.

The hash-based sampling approaches reduce the overhead to zero bits, which is a significant advantage. However, the sampling period in these approaches is not associated with a fixed time interval. Therefore, in some cases adjacent packets may be selected for the sampling, potentially causing measurement errors. Furthermore, when the traffic rate is low, measurements may become significantly infrequent.

It should be noted that most of the marking methods that were presented in this memo are intended for point-to-point measurements, e.g., from MP1 to MP2 in Figure 11. In point-to-multipoint measurements, the mean delay method can be used to measure the loss and delay of the entire point-to-multipoint flow (which includes all the traffic from MP3 to either MP4 or MP5), while other methods such as double marking can be used to measure the point-to-point performance, for example from MP3 to MP5. Alternate marking in multipoint scenarios is discussed in detail in [I-D.fioccola-ippm-multipoint-alt-mark].

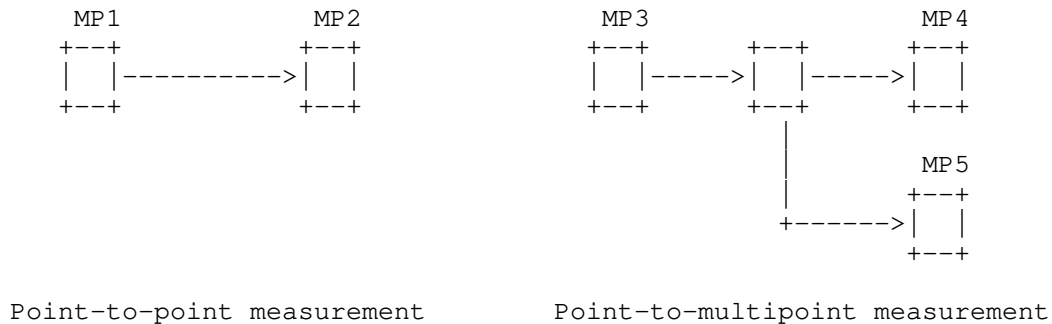


Figure 11: Point-to-point and point-to-multipoint measurements.

8. Alternate Marking using Reserved Values

As mentioned in Section 1, a marking bit is not necessarily a single bit, but may be implemented by using two well-known values in one of the header fields. Similarly, two-bit marking can be implemented using four reserved values.

A notable example is MPLS Synonymous Flow Labels (SFL), as defined in [I-D.bryant-mpls-rfc6374-sfl]. Two MPLS Label values can be used to indicate the two colors of a given LSP: the original Label value, and an SFL value. A similar approach can be applied to IPv6 using the Flow Label field.

The following example illustrates how alternate marking can be implemented using reserved values. The bit multiplexing approach of Section 5.3 is applicable not only to single-bit color indicators, but also to two-value indicators; instead of using a single bit that is toggled between '0' and '1', two values of the indicator field, U and W, can be used in the same manner, allowing both loss and delay measurement to be performed using only two reserved values. Thus, the multiplexing approach of Figure 6 can be illustrated more generally with two values, U and W, as depicted in Figure 12.

A: packet with color 0
 B: packet with color 1

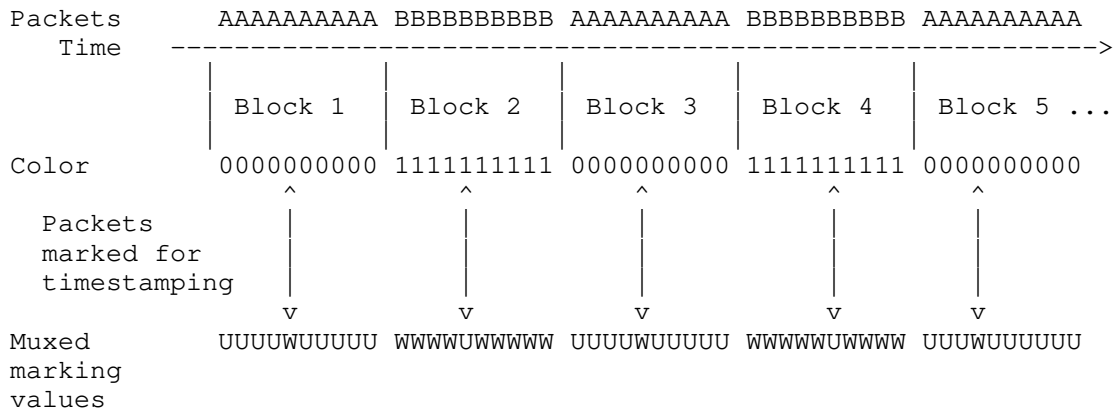


Figure 12: Alternate marking with two multiplexed marking values, U and W.

9. IANA Considerations

This memo includes no requests from IANA.

10. Security Considerations

The security considerations of the alternate marking method are discussed in [I-D.ietf-ippm-alt-mark]. The analysis of Section 7 emphasizes the sensitivity of some of the alternate marking methods to packet drops and to packet reordering. Thus, a malicious attacker may attempt to tamper with the measurements by either selectively dropping packets, or by selectively reordering specific packets. The multiplexed marking method Section 5.3 that is defined in this document requires slightly more stringent synchronization than the conventional marking method, potentially making the method more vulnerable to attacks on the time synchronization protocol. A detailed discussion about the threats against time protocols and how to mitigate them is presented in [RFC7384].

11. References

11.1. Normative References

- [I-D.ietf-ippm-alt-mark]
Fioccola, G., Capello, A., Cociglio, M., Castaldelli, L.,
Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi,
"Alternate Marking method for passive and hybrid
performance monitoring", draft-ietf-ippm-alt-mark-05 (work
in progress), June 2017.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,
<<http://www.rfc-editor.org/info/rfc2119>>.

11.2. Informative References

- [I-D.bryant-mpls-rfc6374-sfl]
Bryant, S., Chen, M., Li, Z., Swallow, G., Sivabalan, S.,
Mirsky, G., and G. Fioccola, "RFC6374 Synonymous Flow
Labels", draft-bryant-mpls-rfc6374-sfl-04 (work in
progress), April 2017.
- [I-D.bryant-mpls-sfl-framework]
Bryant, S., Chen, M., Li, Z., Swallow, G., Sivabalan, S.,
and G. Mirsky, "Synonymous Flow Label Framework", draft-
bryant-mpls-sfl-framework-05 (work in progress), June
2017.
- [I-D.fioccola-ippm-multipoint-alt-mark]
Fioccola, G., Cociglio, M., Sapio, A., and R. Sisto,
"Multipoint Alternate Marking method for passive and
hybrid performance monitoring", draft-fioccola-ippm-
multipoint-alt-mark-00 (work in progress), June 2017.
- [IEEE1588]
IEEE, "IEEE 1588 Standard for a Precision Clock
Synchronization Protocol for Networked Measurement and
Control Systems Version 2", 2008.
- [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,
<<http://www.rfc-editor.org/info/rfc5475>>.
- [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,
<<http://www.rfc-editor.org/info/rfc5905>>.

[RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", RFC 7384, DOI 10.17487/RFC7384, October 2014, <<http://www.rfc-editor.org/info/rfc7384>>.

Authors' Addresses

Tal Mizrahi
Marvell
6 Hamada st.
Yokneam
Israel

Email: talmi@marvell.com

Carmi Arad
Marvell
6 Hamada st.
Yokneam
Israel

Email: carmi@marvell.com

Giuseppe Fioccola
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

Mach(Guoyi) Chen
Huawei Technologies

Email: mach.chen@huawei.com

Lianshu Zheng
Huawei Technologies

Email: vero.zheng@huawei.com

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

Network Working Group
Internet-Draft
Updates: 4656 and 5357 (if approved)
Intended status: Standards Track
Expires: May 17, 2018

A. Morton, Ed.
AT&T Labs
G. Mirsky, Ed.
ZTE Corp.
November 13, 2017

OWAMP and TWAMP Well-Known Port Assignments
draft-morton-ippm-port-twamp-test-02

Abstract

This memo explains the motivation and describes the re-assignment of well-known ports for the OWAMP and TWAMP protocols for control and measurement, and clarifies the meaning and composition of these standards track protocol names for the industry.

The memo updates RFC 4656 and RFC 5357, in terms of the UDP well-known port assignments.

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 17, 2018.

Copyright Notice

Copyright (c) 2017 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. Requirements Language	2
3. Scope	3
4. Definitions	3
5. New Well-Known Ports	4
5.1. Impact on TWAMP-Control Protocol	5
5.2. Impact on OWAMP-Control Protocol	5
5.3. Impact on OWAMP/TWAMP-Test Protocols	6
6. Security Considerations	6
7. IANA Considerations	7
8. Contributors	7
9. Acknowledgements	7
10. References	7
10.1. Normative References	7
10.2. Informative References	8
Authors' Addresses	8

1. Introduction

The IETF IP Performance Metrics (IPPM) working group first developed the One-Way Active Measurement Protocol, OWAMP, specified in [RFC4656]. Further protocol development to support testing resulted in the Two-Way Active Measurement Protocol, TWAMP, specified in [RFC5357].

Both OWAMP and TWAMP require the implementation of a control and mode negotiation protocol (OWAMP-Control and TWAMP-Control) which employs the reliable transport services of TCP (including security configuration and key derivation). The control protocols arrange for the configuration and management of test sessions using the associated test protocol (OWAMP-Test or TWAMP-Test) on UDP transport.

This memo recognizes the value of assigning a well-known UDP port to the *-Test protocols, and that this goal can easily be arranged through port re-assignments.

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

[RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Scope

The scope of this memo is to re-allocate well-known ports for the UDP Test protocols that compose necessary parts of their respective standards track protocols, OWAMP and TWAMP, along with clarifications of the complete protocol composition for the industry.

The memo updates [RFC4656] and [RFC5357], in terms of the UDP well-known port assignments.

4. Definitions

This section defines key terms and clarifies the required composition of the OWAMP and TWAMP standards-track protocols.

OWAMP-Control is the protocol defined in Section 3 of [RFC4656].

OWAMP-Test is the protocol defined in Section 4 of [RFC4656].

OWAMP is described in a direct quote from Section 1.1 of [RFC4656]: "OWAMP actually consists of two inter-related protocols: OWAMP-Control and OWAMP-Test." A similar sentence appears in Section 2 of [RFC4656]. Since the consensus of many dictionary definitions of "consist" is "composed or made up of", implementation of both OWAMP-Control and OWAMP-Test are REQUIRED for standards-track OWAMP specified in [RFC4656].

TWAMP-Control is the protocol defined in Section 3 of [RFC5357].

TWAMP-Test is the protocol defined in Section 4 of [RFC5357].

TWAMP is described in a direct quote from Section 1.1 of [RFC5357]: "Similar to OWAMP [RFC4656], TWAMP consists of two inter-related protocols: TWAMP-Control and TWAMP-Test." Since the consensus of many dictionary definitions of "consist" is "composed or made up of", implementation of both TWAMP-Control and TWAMP-Test are REQUIRED for standards-track TWAMP specified in [RFC5357].

TWAMP Light is an idea described in Informative Appendix I of [RFC5357], and includes an un-specified control protocol (possibly communicating through non-standard means) combined with the TWAMP-Test protocol. The TWAMP Light idea was relegated to the Appendix because it failed to meet the requirements for IETF protocols (there are no specifications for negotiating this form of

operation, and no specifications for mandatory-to-implement security features), as described in the references below:

- o Lars Eggert's Area Director review [LarsAD], where he pointed out that having two variants of TWAMP, Light and Complete (called standards track TWAMP here), required a protocol mechanism to negotiate which variant will be used. See Lars' comment on Sec 5.2. The working group consensus was to place the TWAMP Light description in Appendix I, and to refer to the Appendix only as an "incremental path to adopting TWAMP, by implementing the TWAMP-Test protocol first".
- o Tim Polk's DISCUSS Ballot, which points out that TWAMP Light was an incomplete specification because the key required for authenticated and encrypted modes depended on the TWAMP-Control Session key. See Tim's DISCUSS on 2008-07-16 [TimDISCUSS]. Additional requirement statements were added in the Appendix to address Tim's DISCUSS Ballot (see the last three paragraphs of Appendix I in [RFC5357]).

Since the idea of TWAMP Light clearly includes the TWAMP-Test component of TWAMP, it is considered reasonable for future systems to use the TWAMP-Test well-known UDP port (whose re-allocated assignment is requested here). Clearly, the TWAMP Light idea envisions many components and communication capabilities beyond TWAMP-Test (implementing the security requirements, for example), otherwise the Appendix would be one sentence long (equivocating TWAMP Light with TWAMP-Test only).

5. New Well-Known Ports

Originally, both TCP and UDP well-known ports were assigned to the control protocols that are essential components of standards track OWAMP and TWAMP.

Since OWAMP-Control and TWAMP-Control require TCP transport, they cannot make use of the UDP ports which were originally assigned. However, test sessions using OWAMP-Test or TWAMP-Test operate on UDP transport.

This memo requests re-assignment of the UDP well-known port from the Control protocol to the Test protocol (see the IANA Considerations Section 7). Use of this UDP port is OPTIONAL in standards-track OWAMP and TWAMP. It may simplify some operations to have a well-known port available for the Test protocols, or for future specifications involving TWAMP-Test to use this port as a default port.

5.1. Impact on TWAMP-Control Protocol

Section 3.5 [RFC5357] describes the detailed process of negotiating the Receiver Port number, on which the TWAMP Session-Reflector will send and receive TWAMP-Test packets. The Control-Client, acting on behalf of the Session-Sender, proposes the Receiver port number from the Dynamic Port range [RFC6335]:

"The Receiver Port is the desired UDP port to which TWAMP-Test packets will be sent by the Session-Sender (the port where the Session-Reflector is asked to receive test packets). The Receiver Port is also the UDP port from which TWAMP-Test packets will be sent by the Session-Reflector (the Session-Reflector will use the same UDP port to send and receive packets)."

It is possible that the proposed Receiver Port may be not available, e.g., the port is in use by another test session or another application. In this case:

"... the Server at the Session-Reflector MAY suggest an alternate and available port for this session in the Port field. The Control-Client either accepts the alternate port, or composes a new Session-Request message with suitable parameters. Otherwise, the Server uses the Accept field to convey other forms of session rejection or failure to the Control Client and MUST NOT suggest an alternate port; in this case, the Port field MUST be set to zero."

A Control Client that supports use of the allocated TWAMP-Test Receiver Port Section 7 MAY request to use that port number in the Request-TW-Session Command. If the Server does not support the allocated TWAMP-Test Receiver Port, then it sends an alternate port number in the Accept-Session message with Accept field = 0. Thus the deployment of the allocated TWAMP Receiver Port number is backward compatible with existing TWAMP-Control solutions that are based on [RFC5357]. Of course, use of a UDP port number chosen from the Dynamic Port range [RFC6335] will help to avoid the situation when the Control-Client or Server finds the proposed port being already in use.

5.2. Impact on OWAMP-Control Protocol

As described above, an OWAMP Control Client that supports use of the allocated OWAMP-Test Receiver Port Section 7 MAY request to use that port number in the Request-Session Command. If the Server does not support the allocated OWAMP-Test Receiver Port (or does not have the port available), then it sends an alternate port number in the Accept-Session message with Accept field = 0. Further exchanges proceed as already specified.

5.3. Impact on OWAMP/TWAMP-Test Protocols

OWAMP/TWAMP-Test may be used to measure IP performance metrics in an Equal Cost Multipath (ECMP) environment. Though algorithms to balance IP flows among available paths have not been standardized, the most common is the five-tuple that uses destination IP address, source IP address, protocol type, destination port number, and source port number. When attempting to monitor different paths in ECMP network, it is sufficient to vary only one of five parameters, e.g. the source port number. Thus, there will be no negative impact on ability to arrange concurrent OWAMP/TWAMP test sessions between the same test points to monitor different paths in the ECMP network when using the re-allocated UDP port number as the Receiver Port, as use of the port is optional.

6. Security Considerations

The security considerations that apply to any active measurement of live paths are relevant here as well (see [RFC4656] and [RFC5357]).

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 security and privacy considerations described in the Large Scale Measurement of Broadband Performance (LMAP) Framework [RFC7594], which covers both active and passive techniques.

The registered UDP port as the Receiver Port for OWAMP/TWAMP-Test could become a target of denial-of-service (DoS) or used to aid man-in-the-middle (MITM) attacks. To improve protection from the DoS following methods are recommended:

- o filtering access to the OWAMP/TWAMP Receiver Port by access list;
- o using a non-globally routable IP address for the OWAMP/TWAMP Session-Reflector address.

A MITM attack may try to modify the content of the OWAMP/TWAMP-Test packets in order to alter the measurement results. However, an implementation can use authenticated mode to detect modification of data. In addition, use encrypted mode to prevent eavesdropping and un-detected modification of the OWAMP/TWAMP-Test packets.

7. IANA Considerations

This memo requests re-allocation of two UDP port numbers from the System Ports range [RFC6335]. Specifically, this memo requests that IANA re-allocate UDP ports 861 and 862 as shown below, leaving the TCP port assignments as-is:

Service Name	Port Number	Transport Protocol	Description	Reference
owamp-control	861	tcp	OWAMP-Control	[RFC4656]
owamp-test	861	udp	OWAMP-Test	[RFCXXXX]
twamp-control	861	tcp	TWAMP-Control	[RFC5357]
twamp-test	862	udp	TWAMP-Test Receiver Port	[RFCXXXX]

Table 1 Re-allocated OWAMP and TWAMP Ports

where RFCXXXX is this memo when published.

8. Contributors

Richard Foote and Luis M. Contreras made notable contributions on this topic.

9. Acknowledgements

The authors thank the IPPM working group for their rapid review; also Muthu Arul Mozhi Perumal and Luay Jalil for their participation and suggestions.

10. References

10.1. 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>>.

- [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>>.
- [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>>.
- [RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, DOI 10.17487/RFC6335, August 2011, <<https://www.rfc-editor.org/info/rfc6335>>.
- [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>>.
- [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>>.

10.2. Informative References

- [LarsAD] "<https://mailarchive.ietf.org/arch/msg/ippm/LzcTPYhPhWhbb5-ncR046XKpnzo>", April 2008.
- [TimDISCUSS] "<https://datatracker.ietf.org/doc/rfc5357/history/>", July 2008.

Authors' Addresses

Al Morton (editor)
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

Greg Mirsky (editor)
ZTE Corp.

Email: gregimirsky@gmail.com

ippm
Internet-Draft
Intended status: Experimental
Expires: December 29, 2017

H. Song, Ed.
T. Zhou
Huawei
June 27, 2017

On Scalability of In-situ OAM
draft-song-ippm-ioam-scalability-01

Abstract

This document describes several potential scalability issues when implementing in-situ OAM based on the current in-situ OAM documents and proposes the corresponding solutions and modifications to the current in-situ OAM specification. Specifically, we extend in-situ OAM to support more standard tracing data than is currently defined and add new features to avoid limitations on MTU, bandwidth, forwarding path length, and node processing capability. We provide use cases to motivate our proposal and base the changes on the current in-situ OAM header format specification.

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 December 29, 2017.

Copyright Notice

Copyright (c) 2017 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. Motivation for Better iOAM Scalability	2
2.1. Support Data Type Extensions	3
2.1.1. Motivating Use Cases	3
2.2. Cope with Packet Size Limitation	4
2.2.1. Motivating Use Cases	4
2.3. Adapt to Node Processing Capability	4
2.3.1. Motivating Use Cases	5
3. Scalable Data Type Extension	5
3.1. Data Type Bitmap	5
3.2. Scalable Data Type Extension Use Cases	6
3.3. Consideration for Data Packing	7
3.4. Other Data Extension Possibilities	7
4. Segment In-situ OAM	7
4.1. Segment and Hops	7
4.2. Considerations for Data Handling	8
4.3. Segment iOAM Use Cases	8
5. In-situ OAM Sampling and Data Validation	9
5.1. Valid Node Bitmap and Valid Data Bitmap	9
5.2. iOAM Sampling and Data Validation Use Cases	10
6. Security Considerations	11
7. IANA Considerations	11
8. Acknowledgments	11
9. Contributors	11
10. Informative References	11
Authors' Addresses	11

1. Introduction

In-situ OAM (iOAM) [I-D.brockners-inband-oam-requirements] records OAM information within user packets while the packets traverse a network. The data types and data formats for in-situ OAM data records have been defined in [I-D.brockners-inband-oam-data]. We identify several scalability issues for implementing the current iOAM specification and propose solutions in this draft.

2. Motivation for Better iOAM Scalability

2.1. Support Data Type Extensions

Currently 11 data types and associated formats (including wide format and short format of the same data) are defined in [I-D.brockners-inband-oam-data] . The presence of data is indicated by a 16-bit bitmap in the "OAM-Trace-Type" field.

In the current specification only five bits are left to identify new data types. Moreover, some data is forced to be bundled together as a single unit to save bitmap space and pack data to the ideal size (e.g., the hop limit and the node id are bundled, and the ingress interface id and the egress interface id are bundled), regardless of the fact that an application may only ask for a part of the data. Last but not the least, each data is forced to be 4-byte aligned for easier access, resulting in waste of header space in many cases.

Since the data plane bandwidth, the data plane packet processing, and the management plane data handling are all precious yet scarce resource, the scheme should strive to be simple and precise. The application should be able to control the exact type and format of data it needs to collect and analyze. It is conceivable that more types of data may be introduced in the future. However, the current scheme cannot support it after all the bits in the bitmap are used up.

Currently, bit 7 is used to indicate the presence of variable length opaque state snapshot data. While this data field can be used to store arbitrary data, the data is difficult to be standardized and another schema is needed to decode the data, which may lead to low data plane performance.

2.1.1. Motivating Use Cases

When a flow traverses a series of middleboxes (e.g., Firewall, NAT, and load balancer), its identity (e.g., the 5-tuple) is often altered, which makes the OAM system lose track of the flow trace. In this case, we may want to copy some of the original packet header fields into the iOAM header so the original flow can be identified at any point of the network.

In wireless, mobile, and optical network environments, some physical data associated with a flow (e.g., power, temperature, signal strength, GPS location) need to be collected to monitor the service performance.

Both cases require new iOAM data types. More examples are listed in Section 3.2.

2.2. Cope with Packet Size Limitation

The total size of data is limited by the MTU. When the number of required data types is large and the forwarding path length is long, it is possible that there is not enough space in the iOAM header to save the data. The current proposal is to label the overflow status and stop adding new node data to the packet, leading to loss of information.

Even if the header has enough space to hold the iOAM data, the overhead may be too large and consume too much bandwidth. For example, if we assume moderate 20 bytes of data per node, a path with length of 10 will need 200 bytes to hold the data. This will inflate small 64-byte packets by more than four times. Even for the largest packet size (e.g., 1500 bytes), the overhead (>10%) is not negligible. Therefore, we need to limit the iOAM data overhead without sacrificing the data collection capability.

Here we have another interesting related issue. Packets can be dropped anywhere in a network for various reasons. If we can only collect iOAM data at the path end, we lose all data from the dropped packets and have no idea where the packets are dropped. This defies the purpose of iOAM and makes those iOAM-enabled nodes work in vain.

2.2.1. Motivating Use Cases

Some use cases are described in Section 4.3.

2.3. Adapt to Node Processing Capability

iOAM can designate the flow to add the iOAM header and collect data on the flow forwarding path. The flow can have arbitrary granularity. However, processing the data can be a heavy burden for the network nodes, especially when some data needs to be calculated by the node (e.g., the transit delay). If the flow traffic is heavy, the node may not be able to handle the iOAM processing so many performance issues may occur, such as long latency and packet drop.

Although it is good for the OAM applications to gain the detailed information on every packet at every node, in many cases, such information is often repetitive and redundant. The large quantity of data would also burden the management plane which needs to collect and stream the data for analytics. It is also possible that some nodes cannot provide the requested data at all or are unwilling to provide some data for security or privacy concerns. So a trade-off is needed to balance the performance impact and the data availability and completeness.

2.3.1. Motivating Use Cases

To minimize the network impact, a network operator decides to collect the iOAM data only for initial and last flow packets (e.g., TCP packets with SYN, FIN, and RST flags).

A head node alternates two iOAM headers with each requesting a subset of iOAM data. Hence, each node on the flow path only needs to handle partial data. The requests can be balanced without exhausting the network nodes.

A node is temporarily under heavy traffic load. It is in danger of dropping packets if it tries to satisfy all the iOAM data requests. In this case, it would rather deny some requests than drop user traffic.

More examples are listed in Section 5.2.

3. Scalable Data Type Extension

Based on the observation in Section 2.1, we propose a method for data type encoding which can solve the current limitation and address future data requirements.

3.1. Data Type Bitmap

Bitmap is simple and efficient data structure for high performance data plane implementation. The base bitmap size is kept to be 16 bits. We use one bit to indicate a single type of data in a single format. The last bit in the bitmap (i.e., bit 15), if set, is used to indicate the presence of the next data type bitmap, which is 32 bits long. In the second bitmap, bit 31 is again reserved to indicate a third bitmap, and so on. With each extra bitmap, 31 more data types can be defined.

Figure 1 shows an example of the in-situ OAM header format with two extended OAM trace type fields. Except the OAM Trace Type fields, all other fields remain the same as defined in [I-D.brockners-inband-oam-data].

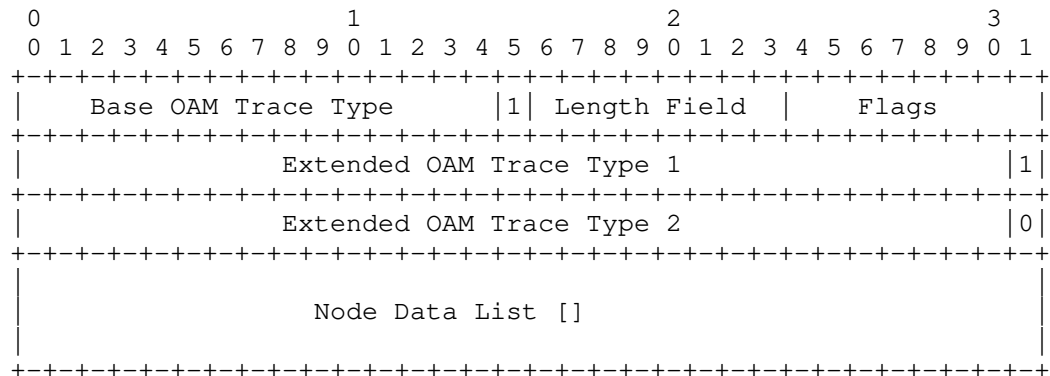


Figure 1: Extended OAM Trace Type Header Format

The specification of the Base OAM Trace Type is the same as the OAM Trace Type in [I-D.brockners-inband-oam-data] except the last bit, which is defined as follows:

- o Bit 15: When set indicates presence of next bit map.

The OAM trace type fields are labeled as Base OAM Trace Type, Extended OAM Trace Type 1, Extended OAM Trace Type 2, and so on. The Base OAM Trace Type is always present. If no data type is asked by the application in Extended OAM Trace Type n and beyond, then the last bit in the previous bitmap is set to 1 and these extended fields are not included in the header. On the other hand, to eliminate ambiguity, if any data is asked for by the application in Extended OAM Trace Type n, then Extended OAM Trace Type 1 to (n-1) must be included in the header, even though no data type in these bitmaps are needed (i.e., all zero bitmap except the last bit).

The actual data in a node is packed together in the same order as listed in the OAM Trace Type bitmap. Each node is padded to be the multiple of 4 bytes.

3.2. Scalable Data Type Extension Use Cases

New types of data can be potentially added and standardized, which demand new bits allocated in the OAM Trace Type bitmaps. Some examples are listed here.

- o Metered flow bandwidth.
- o Time gap between two consecutive flow packets.

- o Remaining time budget to the packet delivery deadline.
- o Buffer occupancy on the Node.
- o Queue depth on each level of hierarchical QoS queues.
- o Packet jitter at the Node.
- o Current packet IP addresses.
- o Current packet port numbers.
- o Other node statistics.

3.3. Consideration for Data Packing

The length of each data must be the multiple of 2 bytes. However, allowing different data type to have different length, while efficient in storage, makes data alignment and packing difficult.

If we can define the maximum number of data types that can be carried per packet, the offset of each data in the node can be pre-calculated and carried in the iOAM header. The overhead can be justified by the overall space saving of the node data list. Otherwise, each data's offset in the node must be calculated in each device, with the help of a table which stores the size of each data type. We can also arrange the bitmap to reflect the data availability order in the system (e.g., the bit for egress_if_id must be after the bit for ingress_if_id), so in a pipeline-based system, the required data can be packed one after one.

3.4. Other Data Extension Possibilities

Bitmap is simple and support parallel processing in hardware, however, it is not the only option to support data type extension. For example, cascaded TLV can be used to support arbitrary number of new data types.

4. Segment In-situ OAM

Based on the observation in Section 2.2, we propose a method to limit the size of the node data list.

4.1. Segment and Hops

A hop is a node on a flow's forwarding path which is capable of processing iOAM data. A segment is a fixed number hops on a flow's forwarding path. While working in the "per hop" mode, the segment

size (SSize) and the remaining hops (RHop), is added to the iOAM header at the edge. Initially, RHop is equal to SSize. At each hop, if RH is not zero, the node data is added to the node data list at the corresponding location and then RH is decremented by 1. If RH is equal to 0 when receiving the packet, the node needs to remove (in incremental trace option) or clear (in pre-allocated trace option) the iOAM node data list and reset RHop to SSize. Then the node will add its data to the node data list as if it is the edge node.

Figure 2 shows the proposed in-situ OAM header format. The last bit (bit 31) in the Flags field is used to indicate the current header is a segment iOAM header. In this context, the third byte of the first word is partitioned into two 4-bit piece. The first piece is used to save the segment size and the second piece is used to save the remaining hops. This limits the maximum segment size to 15.

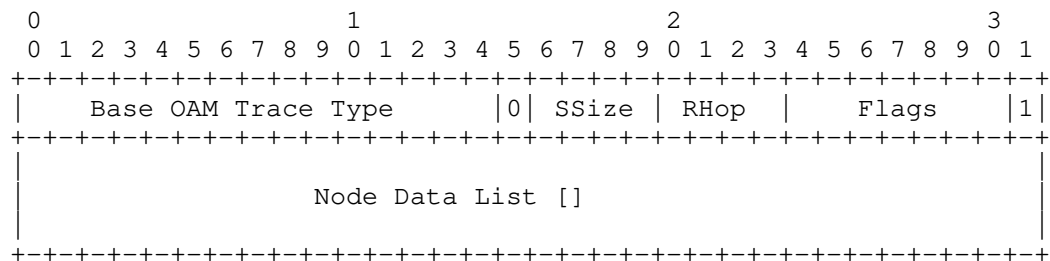


Figure 2: Segment iOAM Header Format

4.2. Considerations for Data Handling

At any hop when RHop is equal to 0, the node data list is copied from the iOAM header. The data can be encapsulated and reported to the controller or the edge node as configured. The encapsulation and report method is beyond the scope of this draft but should be comply with the method used by the iOAM edge node.

The actual size of the last segment may not be equal to SSize but this is not a problem.

4.3. Segment iOAM Use Cases

Segment iOAM is necessary in the following example scenarios:

- o Segment iOAM can be used to detect at which segment the flow packet is dropped. If the SSize is set to 1, then the exact drop

node can be identified. The iOAM data before the dropping point is also retained.

- o The path MTU allows to add at most k node data in the list to avoid fragmentation. Therefore SSize is set to k and at each hop where RHop is 0, the node data list is retrieved and sent in a standalone packet.
- o A flow contains mainly short packets and travels a long path. It would be inefficient to keep a large node data list in the packet so the network bandwidth utilization rate is low. In this case, segment iOAM can be used to limit the ratio of the iOAM data to the flow packet payload.
- o The network allows at most n bytes budget for the iOAM data. There is a tradeoff between the number of data types that can be collected and the number of hops for data collecting. The segment size is therefore necessary to meet the application's data requirement (i.e., $\text{SSize} * \text{Node Data Size} < n$).

5. In-situ OAM Sampling and Data Validation

Based on the observation in Section 1.3, the source edge node should be able to define either the period or the probability to add the iOAM header to the selected flow packet. In this way, only a subset of the flow/sec packets would carry the OAM data, which not only reduces the overall iOAM data quantity but also reduces the processing work load of the network nodes.

5.1. Valid Node Bitmap and Valid Data Bitmap

It is possible that even an iOAM capable node will not add data to the node data list as requested. In some cases, a node can be too busy to handle the data request or some types of the requested data is not available. Therefore, we propose to add two bitmaps, a valid node bitmap and a valid data bit, to the iOAM specification.

The Node Valid Bitmap is inserted before the Node Data List as shown in Figure 3. Each bit in the bitmap corresponds to a hop on the packet's forwarding path. The bits are listed in the same order as the hop on the packet's forwarding path. The bitmap is cleared to all zero at first. If a hop can add data to the Node Data List, the corresponding bit in Node Valid Bitmap is set to 1. The bit location for a hop can be calculated from the length field (e.g, the bit index is equal to $\text{SSize} - \text{RHop}$). The valid node data items in the node data list is equal to the number of 1's in the Node Valid Bitmap.

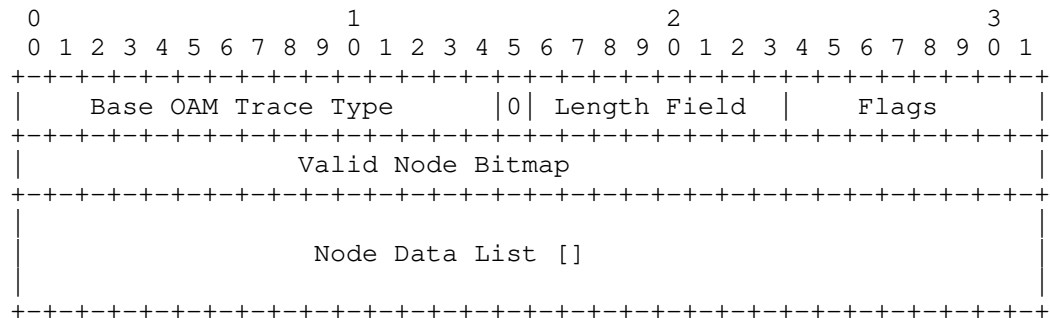


Figure 3: Segment iOAM Header Format

For each node data in the node data list, a Valid Data Bitmap is added before the node data. The number of bits in the Valid Data Bitmap is equal to the number of 1's in the OAM Trace Type bitmaps (excluding the next trace type bitmap indicator bits). When the bit is set, the corresponding data is valid in the node; otherwise, the corresponding data is invalid so the management plane should ignore it after the data is collected.

The size of the bitmap can be padded to two or four bytes, which allow up to 16 or 32 types of data to be included in a node.

5.2. iOAM Sampling and Data Validation Use Cases

We give some examples to show the usefulness of in-situ OAM sampling and data validation features.

- o An application needs to track a flow's forwarding path and knows the path will not change frequently, so it sets a low sampling rate to periodically insert the iOAM header to request the node ID.
- o In a heterogeneous data plane, some nodes support to provide data x but the other nodes do not support it. However, an application is still interested in collecting data x if available. In this case, iOAM header can still be configured to ask for data x but the nodes that cannot provide the data simply invalidates it by resetting the corresponding bit in the valid data bitmap.
- o Multiple sampling rate and multiple data request schema can be defined for a flow based on applications requirements and the data property, so for a flow packet, there can be no iOAM header or different iOAM headers. The node does not need to process all data all the time.

- o For security reason, a node decides to not participate in the iOAM data collection. While it processes the other iOAM header fields as usual, it does not set the node valid bit in the Node Valid Bitmap and add node data to the Node Data List.

6. Security Considerations

There is no extra security considerations beyond those have been identified by in-situ OAM protocol.

7. IANA Considerations

This memo includes no request to IANA.

8. Acknowledgments

We would like to thank Frank Brockners and Carlos Pignataro for helpful comments and suggestions.

9. Contributors

The document is inspired by numerous discussions with James N. Guichard. He also provided significant comments and suggestions to help improve this document.

10. Informative References

[I-D.brockners-inband-oam-data]

Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., and R. <>, "Data Formats for In-situ OAM", draft-brockners-inband-oam-data-02 (work in progress), October 2016.

[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-02 (work in progress), October 2016.

Authors' Addresses

Haoyu Song (editor)
Huawei
2330 Central Expressway
Santa Clara, 95050
USA

Email: haoyu.song@huawei.com

Tianran Zhou
Huawei
156 Beiqing Road
Beijing, 100095
P.R. China

Email: zhoutianran@huawei.com

Internet Engineering Task Force (IETF)
Internet-Draft
Intended status: Informational
Expires: January 4, 2018

H. Zheng
R. Even
Huawei
July 03, 2017

A Proposed Extended Media Delivery Index (eMDI)
draft-zheng-emdi-udp-00

Abstract

A Media Delivery Index (MDI) measurement that can be used as a diagnostic tool or a quality indicator for monitoring flows that are sensitive to arrival time and packet loss is defined in [RFC4445]. This document extends the Media Delivery Index with a new component: Effective Loss Factor (ELF), which takes loss distribution into account when measuring packet loss. ELF is also applicable when certain Forward Error Correction (FEC) schemes are used.

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 January 4, 2018.

Copyright Notice

Copyright (c) 2017 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. Terminology	3
3. Effective Loss Factor (ELF)	4
3.1. Algorithm	4
3.2. Delimitation of Windows	6
4. ELF applicability	8
4.1. ELF Used with FEC	8
5. Security Considerations	9
6. Acknowledgements	9
7. References	9
7.1. Normative References	9
7.2. Informative References	9
Authors' Addresses	9

1. Introduction

[RFC4445] introduces a Media Delivery Index (MDI) to detect network-induced impairments for applications such as streaming video or voice-over-IP. The MDI consists of two components: Delay Factor (DF) and Media Loss Rate (MLR). DF is an indicator of traffic jitter and a measure of deviation from nominal flow rates; MLR counts the number of lost or out-of-order media packets.

This document extends the MDI in [RFC4445], introducing a new component: Effective Loss Factor (ELF). ELF takes loss distribution into account when measuring packet loss. Depending on the type of service, sometimes a seemingly low loss rate flow can have bad Quality of Experience (QoE), since the lost packets aggregate together, causing more severe impairment to the applications. The following is an example:

sequence A:

```
1 2 3 4 5 6 7 8 9
  x   x       x   (packet 2, 5, 9 are lost)
```

sequence B:

```
1 2 3 4 5 6 7 8 9
x x x           (packet 1, 2, 3 are lost)
```

In the above example, sequence A and sequence B have the same loss rate, however sequence B has a higher loss density in the front area, and may result in worse QoE.

The objective of ELF is to give better measurement for such aggregated loss, which MLR and DF in [RFC4445] does not address effectively. MLR provided information about packet loss but does not take into account the distribution of the loss. A large DF may hint a big packet loss but does not address the distribution of the loss and therefore may not provide enough information on the QoE at the endpoint.

Section 4.1 describes how to apply ELF when Forward Error Correction (FEC) is used.

2. Terminology

This document uses the following terms:

sequence: a sequence is all the packets observed over a selected time interval. Packets included in a sequence are in continuous order as sent out from the source.

window: a subsequence of packets in which the calculation of packet loss is confined.

delimitation: a successive mapping of a number of equally sized windows onto a sequence. There are several possible delimitations of a sequence, depending on where it starts. However, a valid delimitation should be within the range of the sequence, and cover the sequence with maximum number of windows.

head extras: for a delimitation that does not start from the first packet of the sequence, head extras consist of any packets in the sequence that are prior to the delimitation.

rear extras: for a delimitation that does not end at the last packet of the sequence, rear extras include any packets in the sequence that are behind the delimitation.

The following figure illustrates the usage of above terms, assuming there are ten packets received in a measurement period of one second, and the windows size is three:

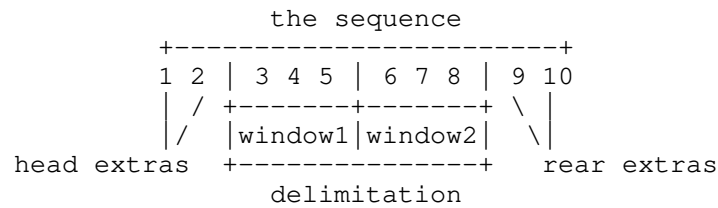


Figure 1: Delimitation of a Sequence

3. Effective Loss Factor (ELF)

3.1. Algorithm

ELF measures loss density in terms of windows. When the number of lost packets in a window exceeds a pre-defined threshold, it indicates the presence of an aggregated loss. The size of windows and threshold are in packets. For example, given a window size of three packets and a threshold of one packet, an aggregated loss is present if the number of lost packets exceeds one, illustrated as below:

```

| 1 2 3 |   (there are three packets in the window)
  x x      (there are two lost packets, denoted by 'x')
  
```

ELF indicates the likelihood of aggregated loss. It is a decimal value between 0 and 1. In the above case, aggregated loss happens in the window, so the value of ELF is 1.

For a sequence of packets, there can be a multiple of windows. The result of ELF should be normalized over all the windows.

The algorithm to calculate ELF over multiple windows is as below:


```

Let:
  k as the ordinal (starting from 1) of the current window
  K as the max ordinal (equal to the number of windows)
  num(k) as the number of lost packets in window k
  R as the loss density threshold
  elf(k) as the ELF value for window k
  J as a variable to accumulate values of elf(k)

J = 0
for k = 1 to K
  if num(k) > R then
    elf(k) = 1
  else
    elf(k) = 0
  endif
  J = J + elf(k)
endfor

ELF over Multiple Windows = J / K

```

Figure 2: Algorithm of Calculating ELF over Multiple Windows

The following is a concrete example:

```

Context:
  Number of Packets = 9
  Window Size = 3
  Loss Density Threshold = 1
  Lost Packets = 2,3,6

Windows:
  | 1 2 3 | 4 5 6 | 7 8 9 |
    x x      x

Window      k      num(k)  elf(k)
| 1 2 3 |    1        2      1
| 4 5 6 |    2        1      0
| 7 8 9 |    3        0      0

```

$J = \text{elf}(1) + \text{elf}(2) + \text{elf}(3) = 1$

$\text{ELF over Multiple Windows} = J / 3 = 1 / 3 = 0.333\dots$

Figure 3: Example of Calculating ELF over Multiple Windows

3.2. Delimitation of Windows

For a sequence of packets, there can be several ways of delimitating it into windows. Each way results in different set of windows, thus the algorithm in Figure 2 may yield different values of ELF. This section proposes a way to generate delimitations of windows, and an algorithm to normalize ELF over different delimitations.

The generation process starts from the default delimitation, which starts from the first packet of a sequence and covers the sequence with maximum number of windows. Taking the example from Figure 1, the default delimitation for the sequence is:

```
| 1 2 3 | 4 5 6 | 7 8 9 | 10
```

Other possible delimitations are found by the following steps:

1. Start from the default delimitation for the sequence.
2. In case there is no packet in rear extras (the last window ends at the last packet of the sequence), break the last window and include all the packets from the last window into rear extras.
3. Slide the boundary of every window one packet forward to make a new delimitation of windows. The movement of windows leaves one packet into the head extras and includes one packet from the rear extras.
4. Repeat step 2 and 3, until the number of packets in head extras equals to (the window size - 1).

The number of delimitations yielded by the above method equals to the window size. For the sequence in Figure 1, all the yielded delimitations are:

```
| 1 2 3 | 4 5 6 | 7 8 9 | 10 (default delimitation)
1 | 2 3 4 | 5 6 7 | 7 8 10 |
1 2 | 3 4 5 | 6 7 8 | 9 10 (note that the last window broke)
```

Figure 4: Example of Delimitations

The algorithm to normalize ELF over different delimitations is as below:

```

Let
  d as the ordinal (starting from 1) of the delimitation
  D as the max ordinal (equal to the window size)
  ELF'(d) as the ELF value for delimitation d
  J as a variable to accumulate values of ELF'(d)

J = 0
for d = 1 to D
  calculate ELF'(d)
  J = J + ELF'(d)
endfor

ELF over Multiple Delimitations = J / D

```

Figure 5: Algorithm of Calculating ELF over Multiple Delimitations

The following is a concrete example:

Context:

```

Number of Packets = 10
Window Size = 3
Loss Density Threshold = 1
Lost Packets = 2,3,6

```

Delimitations:

```

| 1 2 3 | 4 5 6 | 7 8 9 | 10  d = 1
   x x       x               ELF'(1) = 1/3

1 | 2 3 4 | 5 6 7 | 7 8 10 |  d = 2
   x x       x               ELF'(2) = 1/3

1 2 | 3 4 5 | 6 7 8 | 9 10  d = 3
   x  x       x               ELF'(3) = 0

```

$J = \text{ELF}'(1) + \text{ELF}'(2) + \text{ELF}'(3) = 2/3$

$\text{ELF over Multiple Delimitations} = J / 3 = (2/3) / 3 = 0.222\dots$

Figure 6: Example of Calculating ELF over Multiple Delimitations

Note that if not particularly specified, the follow sections refer to ELF as the ELF calculated by the algorithm described in Figure 5.

4. ELF applicability

ELF is reported on a flow. However, how to identify a flow is out of the scope of this document and should be determined by the application, as well as the format and transport that are used to report ELF.

ELF relies on the sequence number of the transport protocol to detect packet loss. For example, when RTP [RFC3550] is used as the transport layer, then packet loss is identified if the sequence numbers of two consecutively received RTP packets are not continuous. The gap between the two sequence numbers can be counted as lost packets. To align with MLR (Section 3.2 of [RFC4445]), out-of-order packets are treated as lost packets.

The two parameters 'Window Size' and 'Loss Density Threshold' together determines the sense of aggregated loss. A configuration of 100:5 ('Window Size': 'Loss Density Threshold') says that an aggregated loss is present if over 5 packets are lost in 100 packets. A configuration of 100:3 suggests a lower sensitivity of aggregated loss than the previous example of 100:5.

As with DF and MLR, ELF is updated and displayed at a selected time interval. The selected time interval should be chosen long enough so that the sequence of packets can be delimited into a number of windows.

Although ELF can be used separately, it is designed to extend MDI as in Section 3.3 of [RFC4445]. Applications can combine ELF with other two components to form an extended MDI:

DF:MLR:ELF

Where:

DF is the Delay Factor
MLR is the Media Loss Rate
ELF is the Effective Loss Factor

In this case, application examples described in Section 3.4 of [RFC4445] also applies to the extended MDI.

4.1. ELF Used with FEC

Applications often use Forward Error Correction (FEC) as a recovery mechanism to compensate for packet loss. ELF can take into account the effectiveness of FEC. Assuming an FEC scheme that encodes a source block of 'L' packets and generates some repair packets. These repair packets can protect the source block from 'M' packet loss. To

reflect the effectiveness of the FEC scheme, set ELF's 'Window Size' to 'L' and 'Loss Density Threshold' to 'M'. This way, small ELF values indicates the loss can be recovered by the FEC scheme effectively. If ELF values grow large, it suggests the FEC scheme is not adequate to recover the loss.

Note that the extra packets generated by FEC scheme is considered to be transmitted out-of-band, apart from the monitored flows.

5. Security Considerations

The new measurement ELF introduced in this document does not introduce any new security issues other than those specified in [RFC4445].

6. Acknowledgements

This document has benefited greatly from the comments of various people. The following individuals have contributed to this document: Rachel Huang, Colin Perkins, Lingyan Wu, Yanfang Zhang.

7. References

7.1. Normative References

[RFC4445] Welch, J. and J. Clark, "A Proposed Media Delivery Index (MDI)", RFC 4445, DOI 10.17487/RFC4445, April 2006, <<http://www.rfc-editor.org/info/rfc4445>>.

7.2. Informative References

[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, <<http://www.rfc-editor.org/info/rfc3550>>.

Authors' Addresses

Hui Zheng (Marvin)
Huawei

Email: marvin.zhenghui@huawei.com

Roni Even
Huawei

Email: roni.even@huawei.com