Internet Engineering Task Force
Internet-Draft
Obsoletes: 5136 (if approved)
Intended status: Informational
Expires: April 19, 2011

X. Cui
Huawei

October 16, 2010

Defining Network Capacity
draft-cui-ippm-rfc5136bis-00

Abstract

This document defines the metric of network capacity, including link capacity aspect, router capacity aspect and path capacity aspect. RFC5136 has defined link capacity and path capacity, where the router impact is implicitly considered in link capacity. However, in this document, router capacity is considerred as a separate factor of path capacity, no longer a factor of link capacity. This document explicitly describes the router capacity and its impact to network capacity, e.g. how to evaluate path capacity.

This document is derived from RFC5136 and obsoletes RFC5136.

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 April 19, 2011.

Copyright Notice

Table of Contents

1. Introduction

The IPPM working group has defined a framework for IP Performance Metrics [RFC2330] and a set of IP Performance Metrics, such as One-way Delay Metric [RFC2679], Packet Delay Variation Metric [RFC3393] and Network Capacity Metric [RFC5136].

Network capacity, which is defined in [RFC5136], is one of the most important IP Performance Metrics in internet. In [RFC5136], network capacity consists of link capacity, path capacity, link usage, link utilization, available link capacity and available path capacity. [RFC5136] also introduces the definitions, measurement and calculation methods and some important formulas.

As stated in [RFC5136], "measuring the capacity of a link or network path is a task that sounds simple, but in reality can be quite complex". There are so many factors and so complicated coupling (between these factors) that the factor of router capacity is not explicitly stated in [RFC5136]. Router is an important element of internet and it is also an essential component of path. In [RFC5136] router impact is implicitly considered in link capacity, but it should be considered in path and path capacity instead, because router is a part of path while not a part of link.

This memo explicitly presents that the router factor should be considered in path, path capacity and related metrics (e.g. available path capacity). For the integrity of network capacity metrics, this memo additionally defines router capacity, router usage, router utilization and available router capacity.

This memo is the latest development based on [RFC5136] and draws heavily from it.

The remainder of this memo is structured as follows.

Section 2.1 contains component definitions and explanations (node, host, router, link, path, etc.)

Section 2.2 contains nominal physical capacity and explanations of link and router.

Section 2.3 give IP-layer capacity definitions and explanstions. It is structured in 11 subsections:

- IP-layer Bits (section 2.3.1)
- IP-type-P Link Capacity (section 2.3.2)
- IP-type-P Link Usage (section 2.3.3)
- IP-type-P Link Utilization (section 2.3.4)

- IP-type-P Available Link Capacity (section 2.3.5)
- IP-type-P Router Capacity (section 2.3.6)
- IP-type-P Router Usage (section 2.3.7)
- IP-type-P Router Utilization (section 2.3.8)
- IP-type-P Available Router Capacity (section 2.3.9)
- IP-type-P Path Capacity (section 2.3.10)
- IP-type-P Available Path Capacity (section 2.3.11)

Section 3 describes changes from [RFC5136]. Section 4 gives some complementary discussion. Section 5 gives discussion conclusion.

1.1. Overview of Capacity

Any physical medium requires that information be encoded and, depending on the medium, there are various schemes to convert information into a sequence of signals that are transmitted physically from one location to another.

While on some media, the maximum frequency of these signals can be thought of as "capacity", on other media, the signal transmission frequency and the information capacity of the medium (channel) may be quite different. For example, a satellite channel may have a carrier frequency of a few gigahertz, but an information-carrying capacity of only a few hundred kilobits per second. Often similar or identical terms are used to refer to these different applications of capacity, adding to the ambiguity and confusion, and the lack of a unified nomenclature makes it difficult to properly build, test, and use various techniques and tools.

We are interested in information-carrying capacity, but even this is not straightforward. Each of the layers, depending on the medium, adds overhead to the task of carrying information. The wired Ethernet uses Manchester coding or 4/5 coding, which cuts down considerably on the "theoretical" capacity. Similarly, RF (radio frequency) communications will often add redundancy to the coding scheme to implement forward error correction because the physical medium (air) is lossy. This can further decrease the information capacity.

In addition to coding schemes, usually the physical layer and the link layer add framing bits for multiplexing and control purposes. For example, on SONET there is physical-layer framing and typically also some layer-2 framing such as High-Level Data Link Control (HDLC), PPP, or ATM.

Aside from questions of coding efficiency, there are issues of how access to the channel is controlled, which also may affect the

capacity. For example, a multiple-access medium with collision detection, avoidance, and recovery mechanisms has a varying capacity from the point of view of the users. This varying capacity depends upon the total number of users contending for the medium, how busy the users are, and bounds resulting from the mechanisms themselves. RF channels may also vary in capacity, depending on range, environmental conditions, mobility, shadowing, etc.

The important points to derive from this discussion are these: First, capacity is only meaningful when defined relative to a given protocol layer in the network. It is meaningless to speak of "link" capacity without qualifying exactly what is meant. Second, capacity is not necessarily fixed, and consequently, a single measure of capacity at any layer may in fact provide a skewed picture (either optimistic or pessimistic) of what is actually available.

1.2. 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. Definitions

In this section, we specify component definitions and capacity definitions.

2.1. Component Definitions

In this section, we specify component definitions for network. We define "node", "Non-IP-Node", "host", "router", "link" and "path" clearly in this section, then we define capacity of network in next section.

2.1.1. Node

IPv6 Specification [RFC2460] defines node is a device that implements IPv6. Framework for IP Performance Metrics [RFC2330] defines host is a computer capable of communicating using the Internet protocols; includes "routers". The notion of host from [RFC2330] is equal to the notion of node from RFC2460. In this document, a node is a computer that implements IP protocol.

Note in this document any node without specicial statement is an IP node.

2.1.2. Non-IP-Node

In this document, a Non-IP-Node is a device that can transmit, receive or forward bit flow, but doesn't implement IP protocol. The examples of Non-IP-Node are ethernet switch and hub.

Note the Non-IP-Node may be part of link and impact the link capacity, for example, consider an ethernet switch that can operate ports at different speeds.

2.1.3. Host

IPv6 Specification [RFC2460] defines a host as any node that is not a router. This document adopts this definition, and the notion of host in this document dosn't includes "routers".

2.1.4. Router

[RFC2460] defines a router is a node that forwards IP packets not explicitly addressed to itself. This document adopts this definition.

2.1.5. Link

[RFC2460] defines link is a communication facility or medium over which nodes can communicate at the link layer, i.e., the layer immediately below IPv6. Examples are Ethernets (simple or bridged); PPP links; X.25, Frame Relay, or ATM networks; and internet (or higher) layer "tunnels", such as tunnels over IPv4 or IPv6 itself. [RFC2330] defines link is a single link-level connection between two (or more) hosts; includes leased lines, ethernets, frame relay clouds, etc. This document adopts the definition from [RFC2460].

Note that link is a bidirectional concept, link terminal and link-layer middle-box are included in link.

2.1.6. Path

As defined in [RFC2330], a path of length n is a sequence of the form (N0, L1, N1, ..., Ln, Nn), where n >= 0, each Ni is a node, each Li is a link between Ni-1 and Ni, each N1...Nn-1 is a router. A pair (Li, Ni) is termed a 'hop'. In an appropriate operational configuration, the links and routers in the path facilitate network-layer communication of packets from N0 to Nn.

Note that path is a unidirectional concept and a path of length one is not equal to the corresponding link. In this case, the Link (i.e., L1) is a part of the path, i.e., the sequence of (N0, L1, N1).

2.2. Definition: Nominal Physical Capacity

Nominal physical link capacity, NomCap(L), is the theoretical maximum amount of data that the link L can support. For example, an OC-3 link would be capable of 155.520 Mbit/s. We stress that this is a measurement at the physical layer and not the network IP layer, which we will define separately. While NomCap(L) is typically constant over time, there are links whose characteristics may allow otherwise, such as the dynamic activation of additional transponders for a satellite link.

Note when we define nominal physical capacity of link, link terminals are considered while the nodes (host or router) which are connected by the link are not gathered. This is because the link terminal (e.g., network interface card) is not integrant of computer, it is only an accessory of the computer. However, there may be some Non-IP-Node in the link, such as an ethereal switch. The physical link capacity is affected by the switch’s ability to process and forward information bits for the given link.

The nominal physical link capacity is provided as a means to help distinguish between the commonly used link-layer capacities and the remaining definitions for IP-layer capacity. The nominal physical capacity provides an upper bound on link capaciy of both IP-layer and link-layer.

However, it is difficult to define the nominal physical capacity of a router. The routers are designed under many limitation, such as physical bound of CPU, memory and system bus. We usually use a pair of common principle to estimate a router: packet per second and bit per second. These two principles are coupled together, and in general, we almost can not correctly estimate a router by either of them. So we don’t define nominal physical router capacity in this document.

2.3. Capacity at the IP Layer

There are many factors that can reduce the IP information carrying capacity of the link. However, the goal of this document is not to become an exhaustive list of such factors. Rather, we outline some of the major examples in the following section, thus providing food for thought to those implementing the algorithms or tools that attempt to measure capacity accurately.

The remaining definitions are all given in terms of "IP-layer bits" in order to distinguish these definitions from the nominal physical capacity of the link.

2.3.1. Definition: IP-layer Bits

IP-layer bits are defined as eight (8) times the number of octets in all IP packets received, from the first octet of the IP header to the last octet of the IP packet payload, inclusive.

IP-layer bits are recorded at the destination D beginning at time T and ending at a time T+I. Since the definitions are based on averages, the two time parameters, T and I, must accompany any report or estimate of the following values in order for them to remain meaningful. It is not required that the interval boundary points fall between packet arrivals at D. However, boundaries that fall within a packet will invalidate the packets on which they fall. Specifically, the data from the partial packet that is contained within the interval will not be counted. This may artificially bias some of the values, depending on the length of the interval and the amount of data received during that interval. We elaborate on what constitutes correctly received data in the next section.

2.3.1.1. Standard or Correctly Formed Packets

The definitions in this document specify that IP packets must be received correctly. The IPPM framework recommends a set of criteria for such standard-formed packets in Section 15 of [RFC2330]. However, it is inadequate for use with this document. Thus, we outline our own criteria below while pointing out any variations or similarities to [RFC2330].

First, data that is in error at layers below IP and cannot be properly passed to the IP layer must not be counted. For example, wireless media often have a considerably larger error rate than wired media, resulting in a reduction in IP link capacity. In accordance with the IPPM framework, packets that fail validation of the IP header must be discarded. Specifically, the requirements in [RFC1812], Section 5.2.2, on IP header validation must be checked, which includes a valid length, checksum, and version field.

The IPPM framework specifies further restrictions, requiring that any transport header be checked for correctness and that any packets with IP options be ignored. However, the definitions in this document are concerned with the traversal of IP-layer bits. As a result, data from the higher layers is not required to be valid or understood as that data is simply regarded as part of the IP packet. The same holds true for IP options. Valid IP fragments must also be counted as they expend the resources of a link even though assembly of the full packet may not be possible. The IPPM framework differs in this area, discarding IP fragments.

For a discussion of duplicates, please see Section 4.2.

In summary, any IP packet that can be properly processed must be included in these calculations.

2.3.1.2. Type P Packets

The definitions in this document refer to "Type P" packets to designate a particular type of flow or sets of flows. As defined in [RFC2330], Section 13, "Type P" is a placeholder for what may be an explicit specification of the packet flows referenced by the metric, or it may be a very loose specification encompassing aggregates. We use the "Type P" designation in these definitions in order to emphasize two things: First, that the value of the capacity measurement depends on the types of flows referenced in the definition. This is because networks may treat packets differently (in terms of queuing and scheduling) based on their markings and classification. Networks may also arbitrarily decide to flow-balance based on the packet type or flow type and thereby affect capacity measurements. Second, the measurement of capacity depends not only on the type of the reference packets, but also on the types of the packets in the "population" with which the flows of interest share the links in the path.

All of this indicates two different approaches to measuring: One is to measure capacity using a broad spectrum of packet types, suggesting that "Type P" should be set as generic as possible. The second is to focus narrowly on the types of flows of particular interest, which suggests that "Type P" should be very specific and narrowly defined. The first approach is likely to be of interest to providers, the second to application users.

As a practical matter, it should be noted that some providers may treat packets with certain characteristics differently than other packets. For example, access control lists, routing policies, and other mechanisms may be used to filter ICMP packets or forward packets with certain IP options through different routes. If a capacity-measurement tool uses these special packets and they are included in the "Type P" designation, the tool may not be measuring the path that it was intended to measure. Tool authors, as well as users, may wish to check this point with their service providers.

2.3.2. Definition: IP-type-P Link Capacity

We define the IP-layer link capacity, C(L,T,I), to be the maximum number of IP-layer bits that can be transmitted from the source S and correctly received by the destination D over the link L during the interval [T, T+I], divided by I. The "maximum" means that IP-type-P

link capacity is the capacity representation when the link is fully utilized (i.e., nominal physical link capacity is fully used.)

In theory, IP-layer link capacity may be calculated out from nominal physical link capacity. Usually, for any link whose link protocol is given, we can know well the encapsulation, overhead and overtail of the link layer protocol. In these cases, for Type P Packets, whose length is Lp, we can get IP-layer link capacity as:

C(L,T,I)
= [Lp/(Lh + Lp + Lt)] * [1 - BER(T, T+I)] * [1 - BDR(T, T+I)] * P(L)

In this formula,
- Lp denotes type P packet length (in IP layer),
- Lh denotes link layer protocol overhead length,
- Lt denotes link layer protocol overtail length,
- BER(T, T+I) denotes Block Error or Lost Rate during the interval [T, T+I],
- BDR(T, T+I) denotes Block Duplication Rate during the interval [T, T+I],
- P(L) denotes nominal physical link capacity of the given link.

Like nominal physical link capacity, IP-type-P link capacity is also a theoretical maximum value. But IP-type-P link capacity is not constant over time, becauese there are many types of link layer protocol and BER and BDR (e.g., BER/BDR of radio channel) may vary in different period.

As defined in section 2.1.5, link is the layer 2 connection between nodes, so the nodes which are connected by the link are not part of the given link. However, there may be some Non-IP-Node in the link, such as an ethereal switch. The IP-type-P link capacity is affected by the switch’s ability to process and forward IP packets for the given link.

IP-type-P link capacity is affected by on-way Non-IP-Node but not affected by the nodes which are connected by the IP link. This means, the injecting node may affect how many packets are transpored between the source S and the destination D during the interval [T, T+I], and the incepting node may also affect how many packets are correctly received in the destination D, but these factors do not affect IP-type-P link capacity, because the capacity is the maximum value can be represented by the link.

IP-type-P link capacity is similar to IP-type-P link usage in some percent. The comparison is described in the next section.

2.3.3. Definition: IP-type-P Link Usage

The average usage of a link L, Used(L,T,I), is the actual number of IP-layer bits from any source, correctly received over link L during the interval [T, T+I], divided by I.

An important distinction between usage and capacity is that the capacity is a theoretical value (constant number) while the usage is a factually represented value (variable number). This is to say, Used(L,T,I) is not the maximum number, but rather, the actual average rate that IP bits are correctly received.

The information transmitted across the link can be generated by any source, including those sources that may not be directly attached to either side of the link. In addition, each information flow from these sources may share any number (from one to n) of links in the overall path between S and D.

2.3.4. Definition: IP-type-P Link Utilization

We express usage as a fraction of the overall IP-layer link capacity.

Util(L,T,I) = (Used(L,T,I) / C(L,T,I))

Thus, the utilization now represents the fraction of the capacity that is being used and is a value between zero (meaning nothing is used) and one (meaning the link is fully saturated). Multiplying the utilization by 100 yields the percent utilization of the link. By using the above, we can now define the available capacity over the link.

2.3.5. Definition: IP-type-P Available Link Capacity

We can now determine the amount of available capacity on a congested link by multiplying the IP-layer link capacity with the complement of the IP-layer link utilization. Thus, the IP-layer available link capacity becomes:

AvailCap(L,T,I) = C(L,T,I) * (1 - Util(L,T,I))

2.3.6. Definition: IP-type-P Router Capacity

As mentioned in section 2.2, we don’t define nominal physical router capacity in this document, we only discuss the router IP capaicy for given packet type.

We define the IP-type-P router capacity, C(R,T,I), to be the maximum number of IP-layer bits (in the formation of type P packet) that can

be correctly transfered from the ingress interfaces to the egress interfaces during the interval [T, T+I], divided by I. Like nominal physical link capacity and IP-layer link capacity, IP-type-P router capacity is also a theoretical maximum value and typically constant over time.

Note this is only a nominal value or an approximation, because the accurate IP layer router capacity depends on many factors. Any router faces the common challenge, its capacity representation depends on its atchitect design, memory (e.g. queueing) management, interface deployment and other implementation issues. for example, a router can support 1000 interfaces and the capacity of 1T bps for IP type P at best. When we configure this router with 100 interfaces/ links, we can get this capacity value (i.e., 1T bps). But if the router is configured with only one ingress interface/link and one egress interface/link, maybe the maximum capacity value this router can present is less than 1T bps, because of its internal bus structure factors, even each link has the IP layer capacity of 2T bps.

On the other hand, as link capacity is node-independent, router capacity is not dependent on bits injection. The ingress link (i.e., the link which is attached to the ingress interface) may affect how many packets are injected to the router and the egress link (i.e., the link which is attached to the egress interface) may affect how many packets are forwarded to the next hop, but note the router capacity is the maximum number that we can get in all cases, for the given type P packets.

2.3.7. Definition: IP-type-P Router Usage

The average usage of a Router R, Used(R,T,I), is the actual number of IP-layer bits (in the formation of type P packet) correctly transfered from any ingress interface to the right engree interface during the interval [T, T+I], divided by I.

An important distinction between usage and capacity is that Used(R,T,I) is not the maximum number, but rather, the actual number of IP bits that are correctly transfered.

The information forwarded through the router can be generated by any source, including those sources that are not directly attached to the router. In addition, each information flow from these sources may share the router in their respective path.

2.3.8. Definition: IP-type-P Router Utilization

We express usage as a fraction of the overall IP-layer router capacity.

Util(R,T,I) = (Used(R,T,I) / C(R,T,I))

Thus, the utilization now represents the fraction of the capacity that is being used and is a value between zero (meaning nothing is used) and one (meaning the router is fully saturated). Multiplying the utilization by 100 yields the percent utilization of the router. By using the above, we can now define the capacity available throuth the router.

2.3.9. Definition: IP-type-P Available Router Capacity

We can now determine the amount of available capacity on a congested router by multiplying the IP-layer router capacity with the complement of the IP-layer router utilization. Thus, the IP-layer available router capacity becomes:

AvailCap(R,T,I) = C(R,T,I) * (1 - Util(R,T,I))

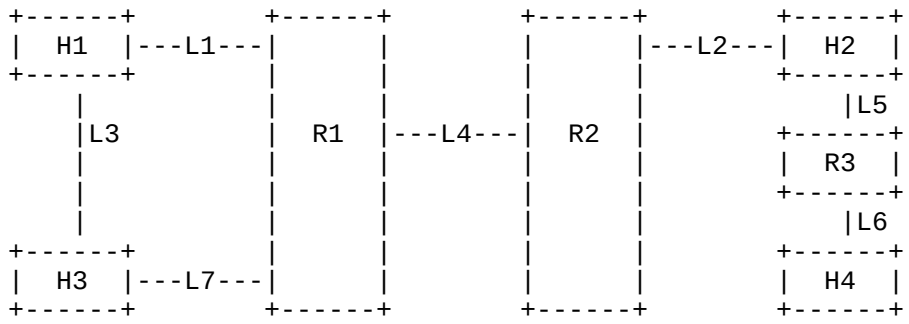
As mentioned in router capacity section, AvailCap(R,T,I) is only an approximation, because the accurate available router capacity depends on many internal factors.

2.3.10. Definition: IP-type-P Path Capacity

Using our definition for IP-layer link capacity and IP-layer router capacity, we can then extend these notions to an entire path.

We define the IP-type-P IP-layer path capacity, C(P,T,I), to be the maximum number of IP-layer bits (in the formation of type P packet) that can be correctly transfered from the source to the destination during the interval [T, T+I], divided by I. Like link capacity and router capacity, path capacity is also a theoretical number.

As mentioned earlier, the path of length n is a sequence of the form (N0, L1, N1, ..., Ln, Nn) and N1, N2, ..., Nn-1 are all routers and part of the path. But these links and routers may be part of one or multiple paths, for example, in the following scenario:



Host, Router, Link and Path

Figure 1

There are multiple paths in this network, such as:

Path P1 (from H1 to H2) -- (H1, L1, R1, L4, R2, L2, H2);
Path P2 (from H1 to H3) -- (H1, L3, H3);
Path P3 (from H1 to H3) -- (H1, L1, R1, L7, H3);
Path P4 (from H2 to H1) -- (H2, L2, R2, L4, R1, L1, H1);
Path P5 (from H2 to H3) -- (H2, L2, R2, L4, R1, L7, H3); and,
Path P6 (from H2 to H4) -- (H2, L5, R3, L6, H4).

Note this is not an exhaustive list. There are many other paths in this network, e.g., (R1, L4, R2).

In this scenario, the path (H1, L3, H4) and the path (H2, L5, R3, L6, H4) are exclusive path. The IP-layer capacity of an exclusive path may be calculated by:

C(P,T,I) = min {1..n} {C(Ln,T,I), C(Rn,T,I)}

we can also find that the link of L1, L2, L4 are all shared by multiple paths and the router of R1 and R2 are the same. Because of the capacity sharing, path capacity rather depends on the capacity contribution from the links and the routers than the IP-layer capacity of themselves. So for any given path whose link or router overlaps with other path, the IP-layer path capacity becomes more complex, it depends on not only the IP-layer capacity of the links and the routers but also the "competitive" traffic (also in formation of type P packet) of other paths, which have overlap segment with the given path. This means the capacity of non-exclusive path is a variable, is external situation dependent.

It is very difficult to calculate IP-type-P path capacity of non-exclusive path in general but we can get out the maximum number of path capacity from links and routers, to indicate the upper bound on path capacity.

The maximum number of IP-layer capacity of non-exclusive path may be calculated by:

Cmax(P,T,I) = min {1..n} {C(Ln,T,I), C(Rn,T,I)}

2.3.11. Definition: IP-type-P Available Path Capacity

Using our definition for IP-layer available link capacity and IP-layer available router capacity, we can then extend these notions to an entire path, such that the IP-layer available path capacity simply becomes that of the link and router with the smallest available capacity along that path.

AvailCap(P,T,I) = min {1..n} {AvailCap(Ln,T,I), AvailCap(Rn,T,I)}

Since measurements of available capacity are more volatile than that of link capacity, we stress the importance that both the time and interval be specified as their values have a great deal of influence on the results. In addition, a sequence of measurements may be beneficial in offsetting the volatility when attempting to characterize available capacity.

3. Changes from RFC5136

In general, this document clarifies some definitions (e.g., path) and expounds that the capacity metrics (e.g., IP-type-P link capacity) are theoretical number. In addition, usage metrics (e.g., IP-type-P link usage) are very different from capacity metrics because they are actual number represented in measurement cases.

3.1. Node Definition

Section 2.1 from [RFC5136] has been changed from:

We define nodes as hosts, routers, Ethernet switches, or any other device where the input and output links can have different characteristics.

to Section 2.1.1 of this memo:

Node is a computer that implements IP protocol.

Reason/summarization:

The reason for this modification is to follow the most idiomatic definition. Non-IP device is excluded in the notion of node.

3.2. Link Definition

Section 2.1 from [RFC5136] has been changed from:

A link is a connection between two of these network devices or nodes.

to Section 2.1.5 of this memo:

Link is a communication facility or medium over which nodes can communicate at the link layer, i.e., the layer immediately below IPv6. Examples are Ethernets (simple or bridged); PPP links; X.25, Frame Relay, or ATM networks; and internet (or higher) layer "tunnels", such as tunnels over IPv4 or IPv6 itself.

Reason/summarization:

The reason for this modification is to clarify the notion. The connection between layer1 or layer2 devices is not an absolute link, but only a segment of link.

3.3. Path Definition

Section 2.1 from [RFC5136] has been changed from:

We then define a path P of length n as a series of links (L1, L2, ..., Ln) connecting a sequence of nodes (N1, N2, ..., Nn+1). A source S and destination D reside at N1 and Nn+1, respectively.

to Section 2.1.6 of this memo:

A path of length n is a sequence of the form (N0, L1, N1, ..., Ln, Nn), where n >= 0, each Ni is a node, each Li is a link between Ni-1 and Ni, each N1...Nn-1 is a router. A pair (Li, Ni) is termed a 'hop'. In an appropriate operational configuration, the links and routers in the path facilitate network-layer communication of packets from N0 to Nn.

Reason/summarization:

The reason for this modification is to emphasize that routers in the path are essential component of the path.

3.4. Definition: Nominal Physical Capacity

Section 2.2 from [RFC5136] has been changed from "Definition: Nominal Physical Link Capacity" to "Definition: Nominal Physical Capacity". And some statement are added, including:

Note when we define nominal physical capacity of link, link terminals are considered while the nodes (host or router) which are connected by the link are not gathered. This is because the link terminal (e.g., network interface card) is not integrant of computer, it is only an accessory of the computer. However, there may be some non-IP-node in the link, such as the ethereal switch. The physical link capacity is affected by the switch's ability to process and forward information bits for the given link.

and,

However, it is difficult to define the nominal physical capacity of a router. The routers are designed under many limitation, such as physical bound of CPU, memory and system bus. We usually use a pair of common principle to estimate a router: packet per second and bit per second. These two principles are coupled together, and in general, we almost can not correctly estimate a router by either of them. So we don't define nominal physical router capacity in this document.

3.5. IP-type-P Link Capacity

Section 2.3.2 from [RFC5136] has been changed from:

We define the IP-layer link capacity, C(L,T,I), to be the maximum number of IP-layer bits that can be transmitted from the source S and correctly received by the destination D over the link L during the interval [T, T+I], divided by I.

As mentioned earlier, this definition is affected by many factors that may change over time. For example, a device's ability to process and forward IP packets for a particular link may have varying effect on capacity, depending on the amount or type of traffic being processed.

to Section 2.3.2 of this memo:

We define the IP-layer link capacity, C(L,T,I), to be the maximum number of IP-layer bits that can be transmitted from the source S and correctly received by the destination D over the link L during the interval [T, T+I], divided by I. The "maximum" means that IP-type-P link capacity is the capacity representation when the link is fully

utilized (i.e., nominal physical link capacity is fully used.)

In theory, IP-layer link capacity may be calculated out from nominal physical link capacity. Usually, for any link whose link protocol is given, we can know well the encapsulation, overhead and overtail of the link layer protocol. In these cases, for Type P Packets, whose length is Lp, we can get IP-layer link capacity as:

C(L,T,I)
= [Lp/(Lh + Lp + Lt)] * [1 - BER(T, T+I)] * [1 - BDR(T, T+I)] * P(L)

In this formula,
- Lp denotes type P packet length (in IP layer),
- Lh denotes link layer protocol overhead length,
- Lt denotes link layer protocol overtail length,
- BER(T, T+I) denotes Block Error or Lost Rate during the interval [T, T+I],
- BDR(T, T+I) denotes Block Duplication Rate during the interval [T, T+I],
- P(L) denotes nominal physical link capacity of the given link.

Like nominal physical link capacity, IP-type-P link capacity is also a theoretical maximum value. But IP-type-P link capacity is not constant over time, becauese there are many types of link layer protocol and BER and BDR (e.g., BER/BDR of radio channel) may vary in different period.

As defined in section 2.1.5, link is the layer 2 connection between nodes, so the nodes which are connected by the link are not part of the given link. However, there may be some Non-IP-Node in the link, such as the ethereal switch. The IP-type-P link capacity is affected by the switch's ability to process and forward IP packets for the given link.

IP-type-P link capacity is affected by on-way Non-IP-Node but not affected by the nodes which are connected by the IP link. This means, the injecting node may affect how many packets are transpored between the source S and the destination D during the interval [T, T+I], and the incepting node may also affect how many packets are correctly received in the destination D, but these factors do not affect IP-type-P link capacity, because the capacity is the maximum value can be represented by the link.

IP-type-P link capacity is similar to IP-type-P link usage in some percent. The comparison is described in the next section.

Reason/summarization:

This modification clarifies the definition and calculation of link capacity and explicitly indicates that node doesn't affect link capacity but the Non-IP-Node which is part of link does.

3.6. IP-type-P Router Capacity

Section 2.3.6 is newly added in this memo, as:

As mentioned in section 2.2, we don't define nominal physical router capacity in this document, we only discuss the router IP capaicy for given packet type.

We define the IP-type-P router capacity, C(R,T,I), to be the maximum number of IP-layer bits (in the formation of type P packet) that can be correctly transfered from the ingress interfaces to the egress interfaces during the interval [T, T+I], divided by I. Like nominal physical link capacity and IP-layer link capacity, IP-type-P router capacity is also a theoretical maximum value and typically constant over time

Note this is only a nominal value or an approximation, because the accurate IP layer router capacity depends on many factors. Any router faces the common challenge, its capacity representation depends on its atchitect design, memory (e.g. queueing) management, interface deployment and other implementation issues. for example, a router can support 1000 interfaces and the capacity of 1T bps for IP type P at best. When we configure this router with 100 interfaces/ links, we can get this capacity value (i.e., 1T bps). But if the router is configured with only one ingress interface/link and one egress interface/link, maybe the maximum capacity value this router can present is less than 1T bps, because of its internal bus structure factors, even each link has the IP layer capacity of 2T bps.

On the other hand, as link capacity is node-independent, router capacity is not dependent on bits injection. The ingress link (i.e., the link which is attached to the ingress interface) may affect how many packets are injected to the router and the egress link (i.e., the link which is attached to the egress interface) may affect how many packets are forwarded to the next hop, but note the router capacity is the maximum number that we can get in all cases, for the given type P packets.

Reason/summarization:

This modification defines IP-layer router capacity aspect from network capacity.

3.7. IP-type-P Router Usage

Section 2.3.7 is newly added in this memo, as:

The average usage of a Router R, Used(R,T,I), is the actual number of IP-layer bits (in the formation of type P packet) correctly transfered from any ingress interface to the right engree interface during the interval [T, T+I], divided by I.

An important distinction between usage and capacity is that Used(R,T,I) is not the maximum number, but rather, the actual number of IP bits that are correctly transfered.

The information forwarded through the router can be generated by any source, including those sources that are not directly attached to the router. In addition, each information flow from these sources may share the router in their respective path.

Reason/summarization:

This modification defines IP-layer router usage aspect from network capacity.

3.8. IP-type-P Router Utilization

Section 2.3.8 is newly added in this memo, as:

We express usage as a fraction of the overall IP-layer router capacity.

Util(R,T,I) = (Used(R,T,I) / C(R,T,I))

Thus, the utilization now represents the fraction of the capacity that is being used and is a value between zero (meaning nothing is used) and one (meaning the router is fully saturated). Multiplying the utilization by 100 yields the percent utilization of the router. By using the above, we can now define the capacity available throuth the router.

Reason/summarization:

This modification defines IP-layer router utilization aspect from network capacity.

3.9. IP-type-P Available Router Capacity

Section 2.3.9 is newly added in this memo, as:

We can now determine the amount of available capacity on a congested router by multiplying the IP-layer router capacity with the complement of the IP-layer router utilization. Thus, the IP-layer available router capacity becomes:

AvailCap(R,T,I) = C(R,T,I) * (1 - Util(R,T,I))

As mentioned in router capacity section, AvailCap(R,T,I) is only an approximation, because the accurate available router capacity depends on many internal factors.

Reason/summarization:

This modification defines IP-layer available router capacity aspect from network capacity.

3.10. IP-type-P Path Capacity

Section 2.3.3 from [RFC5136] has been changed from:

Using our definition for IP-layer link capacity, we can then extend this notion to an entire path, such that the IP-layer path capacity simply becomes that of the link with the smallest capacity along that path.

C(P,T,I) = min {1..n} {C(Ln,T,I)}

The previous definitions specify the number of IP-layer bits that can be transmitted across a link or path should the resource be free of any congestion. It represents the full capacity available for traffic between the source and destination. Determining how much capacity is available for use on a congested link is potentially much more useful. However, in order to define the available capacity, we must first specify how much is being used.

to Section 2.3.10 of this memo:

We define the IP-type-P IP-layer path capacity, C(P,T,I), to be the maximum number of IP-layer bits (in the formation of type P packet) that can be correctly transfered from the source to the destination during the interval [T, T+I], divided by I. Like link capacity and router capacity, path capacity is also a theoretical number.

The IP-layer capacity of an exclusive path may be calculated by:

C(P,T,I) = min {1..n} {C(Ln,T,I), C(Rn,T,I)}

It is very difficult to calculate IP-type-P path capacity of non-

exclusive path in general but we can get out the maximum number of path capacity from links and routers, to indicate the upper bound on path capacity.

The maximum number of IP-layer capacity of non-exclusive path may be calculated by:

Cmax(P,T,I) = min {1..n} {C(Ln,T,I), C(Rn,T,I)}

Reason/summarization:

This modification clarifies how to correctly evaluate path capacity. Router capacity is considered for path capacity.

3.11. IP-type-P Available Path Capacity

Section 2.3.7 from [RFC5136] has been changed from:

Using our definition for IP-layer available link capacity, we can then extend this notion to an entire path, such that the IP-layer available path capacity simply becomes that of the link with the smallest available capacity along that path.

AvailCap(P,T,I) = min {1..n} {AvailCap(Ln,T,I)}

to Section 2.3.11 of this memo:

Using our definition for IP-layer available link capacity and IP-layer available router capacity, we can then extend these notions to an entire path, such that the IP-layer available path capacity simply becomes that of the link and router with the smallest available capacity along that path.

AvailCap(P,T,I) = min {1..n} {AvailCap(Ln,T,I), AvailCap(Rn,T,I)}

Reason/summarization:

This modification clarifies how to correctly evaluate available path capacity. Available router capacity is considered for available path capacity.

4. Discussion

4.1. Time and Sampling

We must emphasize the importance of time in the basic definitions of these quantities. We know that traffic on the Internet is highly

variable across all time scales. This argues that the time and length of measurements are critical variables in reporting available capacity measurements and must be reported when using these definitions.

The closer to "instantaneous" a metric is, the more important it is to have a plan for sampling the metric over a time period that is sufficiently large. By doing so, we allow valid statistical inferences to be made from the measurements. An obvious pitfall here is sampling in a way that causes bias. For example, a situation where the sampling frequency is a multiple of the frequency of an underlying condition.

4.2. Hardware Duplicates

We briefly consider the effects of paths where hardware duplication of packets may occur. In such an environment, a node in the network path may duplicate packets, and the destination may receive multiple, identical copies of these packets. Both the original packet and the duplicates can be properly received and appear to be originating from the sender. Thus, in the most generic form, duplicate IP packets are counted in these definitions. However, hardware duplication can affect these definitions depending on the use of "Type P" to add additional restrictions on packet reception. For instance, a restriction only to count uniquely-sent packets may be more useful to users concerned with capacity for meaningful data. In contrast, the more general, unrestricted metric may be suitable for a user who is concerned with raw capacity. Thus, it is up to the user to properly scope and interpret results in situations where hardware duplicates may be prevalent.

4.3. Other Potential Factors

IP encapsulation does not affect the definitions as all IP header and payload bits must be counted regardless of content. However, IP packets of different sizes can lead to a variation in the amount of overhead needed at the lower layers to transmit the data, thus altering the overall IP link-layer capacity.

Should the link happen to employ a compression scheme such as RObust Header Compression (ROHC) [RFC3095] or V.44 [V44], some of the original bits are not transmitted across the link. However, the inflated (not compressed) number of IP-layer bits should be counted.

4.4. Common Terminology in Literature

Certain terms are often used to characterize specific aspects of the presented definitions. The link with the smallest capacity is

commonly referred to as the "narrow link" of a path. Also, the link with the smallest available capacity is often referred to as the "tight link" within a path. So, while a given link may have a very large capacity, the overall congestion level on the link makes it the likely bottleneck of a connection. Conversely, a link that has the smallest capacity may not be the bottleneck should it be lightly loaded in relation to the rest of the path.

Also, literature often overloads the term "bandwidth" to refer to what we have described as capacity in this document. For example, when inquiring about the bandwidth of a 802.11b link, a network engineer will likely answer with 11 Mbit/s. However, an electrical engineer may answer with 25 MHz, and an end user may tell you that his observed bandwidth is 8 Mbit/s. In contrast, the term "capacity" is not quite as overloaded and is an appropriate term that better reflects what is actually being measured.

4.5. Comparison to Bulk Transfer Capacity (BTC)

Bulk Transfer Capacity (BTC) [RFC3148] provides a distinct perspective on path capacity that differs from the definitions in this document in several fundamental ways. First, BTC operates at the transport layer, gauging the amount of capacity available to an application that wishes to send data. Only unique data is measured, meaning header and retransmitted data are not included in the calculation. In contrast, IP-layer link capacity includes the IP header and is indifferent to the uniqueness of the data contained within the packet payload. (Hardware duplication of packets is an anomaly addressed in a previous section.) Second, BTC utilizes a single congestion-aware transport connection, such as TCP, to obtain measurements. As a result, BTC implementations react strongly to different path characteristics, topologies, and distances. Since these differences can affect the control loop (propagation delays, segment reordering, etc.), the reaction is further dependent on the algorithms being employed for the measurements. For example, consider a single event where a link suffers a large duration of bit errors. The event could cause IP-layer packets to be discarded, and the lost packets would reduce the IP-layer link capacity. However, the same event and subsequent losses would trigger loss recovery for a BTC measurement resulting in the retransmission of data and a potentially reduced sending rate. Thus, a measurement of BTC does not correspond to any of the definitions in this document. Both techniques are useful in exploring the characteristics of a network path, but from different perspectives.

5. Conclusion

In this document, we have defined a set of quantities related to the capacity of links, routers and paths in an IP network. In these definitions, we have tried to be as clear as possible and take into account various characteristics that links, routers and paths can have. The goal of these definitions is to enable researchers who propose capacity metrics to relate those metrics to these definitions and to evaluate those metrics with respect to how well they approximate these quantities.

In addition, we have pointed out some key auxiliary parameters and opened a discussion of issues related to valid inferences from available capacity metrics.

6. Security Considerations

This document specifies definitions regarding IP traffic traveling between a source and destination in an IP network. These definitions do not raise any security issues and do not have a direct impact on the networking protocol suite.

Tools that attempt to implement these definitions may introduce security issues specific to each implementation. Both active and passive measurement techniques can be abused, impacting the security, privacy, and performance of the network. Any measurement techniques based upon these definitions must include a discussion of the techniques needed to protect the network on which the measurements are being performed.

7. IANA Considerations

This document has no actions for IANA.

8. Acknowledgments

The author would especially like to acknowledge Phil Chimento and Joseph Ishac for their great contribution on the item of network capacity. The author would like to acknowledge Mark Allman, Patrik Arlos, Matt Mathis, Al Morton, Stanislav Shalunov, and Matt Zekauskas for their contribution on [RFC5136], which is the basis of this document.

The author would also like to acknowledge Brian E Carpenter, Adrian Farrel, Spencer Dawkins, David Harrington and Barry Leiba for their

review and discussion in the early stage of this document.

9. References

9.1. Normative References

[RFC1812] Baker, F., "Requirements for IP Version 4 Routers", RFC 1812, June 1995.

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

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

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

9.2. Informative References

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

[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, July 2001.

[RFC3148] Mathis, M. and M. Allman, "A Framework for Defining Empirical Bulk Transfer Capacity Metrics", RFC 3148, July 2001.

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

[RFC5136] Chimento, P. and J. Ishac, "Defining Network Capacity", RFC 5136, February 2008.

[V44] ITU Telecommunication Standardization Sector (ITU-T) Recommendation V.44, "Data Compression Procedures", November 2000.

Author's Address

Xiangsong Cui (editor)
Huawei
KuiKe Bld., No.9 Xinxi Rd., Shang-Di Information Industry Base
Beijing, 100085
P.R. China

Phone:
Email: Xiangsong.Cui@huawei.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: September 15, 2011

R. Geib, Ed.
Deutsche Telekom
A. Morton
AT&T Labs
R. Fardid
Cariden Technologies
A. Steinmitz
HS Fulda

March 14, 2011

IPPM standard advancement testing
draft-ietf-ippm-metrictest-02

Abstract

This document specifies tests to determine if multiple independent instantiations of a performance metric RFC have implemented the specifications in the same way. This is the performance metric equivalent of interoperability, required to advance RFCs along the standards track. Results from different implementations of metric RFCs will be collected under the same underlying network conditions and compared using state of the art statistical methods. The goal is an evaluation of the metric RFC itself, whether its definitions are clear and unambiguous to implementors and therefore a candidate for advancement on the IETF standards track.

Status of this Memo

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

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

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

This Internet-Draft will expire on September 15, 2011.

Copyright Notice

Table of Contents

1. Introduction

The Internet Standards Process RFC2026 [RFC2026] requires that for a IETF specification to advance beyond the Proposed Standard level, at least two genetically unrelated implementations must be shown to interoperate correctly with all features and options. This requirement can be met by supplying:

- o evidence that (at least a sub-set of) the specification has been implemented by multiple parties, thus indicating adoption by the IETF community and the extent of feature coverage.

- o evidence that each feature of the specification is sufficiently well-described to support interoperability, as demonstrated through testing and/or user experience with deployment.

In the case of a protocol specification, the notion of "interoperability" is reasonably intuitive - the implementations must successfully "talk to each other", while exercising all features and options. To achieve interoperability, two implementors need to interpret the protocol specifications in equivalent ways. In the case of IP Performance Metrics (IPPM), this definition of interoperability is only useful for test and control protocols like the One-Way Active Measurement Protocol, OWAMP [RFC4656], and the Two-Way Active Measurement Protocol, TWAMP [RFC5357].

A metric specification RFC describes one or more metric definitions, methods of measurement and a way to report the results of measurement. One example would be a way to test and report the One-way Delay that data packets incur while being sent from one network location to another, One-way Delay Metric.

In the case of metric specifications, the conditions that satisfy the "interoperability" requirement are less obvious, and there was a need for IETF agreement on practices to judge metric specification "interoperability" in the context of the IETF Standards Process. This memo provides methods which should be suitable to evaluate metric specifications for standards track advancement. The methods proposed here MAY be generally applicable to metric specification RFCs beyond those developed under the IPPM Framework [RFC2330].

Since many implementations of IP metrics are embedded in measurement systems that do not interact with one another (they were built before OWAMP and TWAMP), the interoperability evaluation called for in the IETF standards process cannot be determined by observing that independent implementations interact properly for various protocol exchanges. Instead, verifying that different implementations give statistically equivalent results under controlled measurement

conditions takes the place of interoperability observations. Even when evaluating OWAMP and TWAMP RFCs for standards track advancement, the methods described here are useful to evaluate the measurement results because their validity would not be ascertained in typical interoperability testing.

The standards advancement process aims at producing confidence that the metric definitions and supporting material are clearly worded and unambiguous, or reveals ways in which the metric definitions can be revised to achieve clarity. The process also permits identification of options that were not implemented, so that they can be removed from the advancing specification. Thus, the product of this process is information about the metric specification RFC itself: determination of the specifications or definitions that are clear and unambiguous and those that are not (as opposed to an evaluation of the implementations which assist in the process).

This document defines a process to verify that implementations (or practically, measurement systems) have interpreted the metric specifications in equivalent ways, and produce equivalent results.

Testing for statistical equivalence requires ensuring identical test setups (or awareness of differences) to the best possible extent. Thus, producing identical test conditions is a core goal of the memo. Another important aspect of this process is to test individual implementations against specific requirements in the metric specifications using customized tests for each requirement. These tests can distinguish equivalent interpretations of each specific requirement.

Conclusions on equivalence are reached by two measures.

First, implementations are compared against individual metric specifications to make sure that differences in implementation are minimised or at least known.

Second, a test setup is proposed ensuring identical networking conditions so that unknowns are minimized and comparisons are simplified. The resulting separate data sets may be seen as samples taken from the same underlying distribution. Using state of the art statistical methods, the equivalence of the results is verified. To illustrate application of the process and methods defined here, evaluation of the One-way Delay Metric [RFC2679] is provided in an Appendix. While test setups will vary with the metrics to be validated, the general methodology of determining equivalent results will not. Documents defining test setups to evaluate other metrics should be developed once the process proposed here has been agreed and approved.

The metric RFC advancement process begins with a request for protocol action accompanied by a memo that documents the supporting tests and results. The procedures of [RFC2026] are expanded in[RFC5657], including sample implementation and interoperability reports. Section 3 of [morton-advance-metrics-01] can serve as a template for a metric RFC report which accompanies the protocol action request to the Area Director, including description of the test set-up, procedures, results for each implementation and conclusions.

Changes from WG-01 to WG-02:

- Clarification of the number of test streams recommended in section 3.2.
- Clarifications on testing details in sections 3.3 and 3.4.
- Spelling corrections throughout.

Changes from WG -00 to WG -01 draft

- Discussion on merits and requirements of a distributed lab test using only local load generators.
- Proposal of metrics suitable for tests using the proposed measurement configuration.
- Hint on delay caused by software based L2TPv3 implementation.
- Added an appendix with a test configuration allowing remote tests comparing different implementations across the network.
- Proposal for maximum error of "equivalence", based on performance comparison of identical implementations. This may be useful for both ADK and non-ADK comparisons.

Changes from prior ID -02 to WG -00 draft

- Incorporation of aspects of reporting to support the protocol action request in the Introduction and section 3.5
- Overhaul of section 3.2 regarding tunneling: Added generic tunneling requirements and L2TPv3 as an example tunneling mechanism fulfilling the tunneling requirements. Removed and adapted some of the prior references to other tunneling protocols
- Softened a requirement within section 3.4 (MUST to SHOULD on precision) and removed some comments of the authors.

- Updated contact information of one author and added a new author.
- Added example C++ code of an Anderson-Darling two sample test implementation.

Changes from ID -01 to ID -02 version

- Major editorial review, rewording and clarifications on all contents.
- Additional text on parallel testing using VLANs and GRE or Pseudowire tunnels.
- Additional examples and a glossary.

Changes from ID -00 to ID -01 version

- Addition of a comparison of individual metric implementations against the metric specification (trying to pick up problems and solutions for metric advancement [morton-advance-metrics]).
- More emphasis on the requirement to carefully design and document the measurement setup of the metric comparison.
- Proposal of testing conditions under identical WAN network conditions using IP in IP tunneling or Pseudo Wires and parallel measurement streams.
- Proposing the requirement to document the smallest resolution at which an ADK test was passed by 95%. As no minimum resolution is specified, IPPM metric compliance is not linked to a particular performance of an implementation.
- Reference to RFC 2330 and RFC 2679 for the 95% confidence interval as preferred criterion to decide on statistical equivalence
- Reducing the proposed statistical test to ADK with 95% confidence.

1.1. 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. Basic idea

The implementation of a standard compliant metric is expected to meet

the requirements of the related metric specification. So before comparing two metric implementations, each metric implementation is individually compared against the metric specification.

Most metric specifications leave freedom to implementors on non-fundamental aspects of an individual metric (or options). Comparing different measurement results using a statistical test with the assumption of identical test path and testing conditions requires knowledge of all differences in the overall test setup. Metric specification options chosen by implementors have to be documented. It is REQUIRED to use identical implementation options wherever possible for any test proposed here. Calibrations proposed by metric standards should be performed to further identify (and possibly reduce) potential sources of errors in the test setup.

The Framework for IP Performance Metrics [RFC2330] expects that a "methodology for a metric should have the property that it is repeatable: if the methodology is used multiple times under identical conditions, it should result in consistent measurements." This means an implementation is expected to repeatedly measure a metric with consistent results (repeatability with the same result). Small deviations in the test setup are expected to lead to small deviations in results only. To characterise statistical equivalence in the case of small deviations, RFC 2330 and [RFC2679] suggest to apply a 95% confidence interval. Quoting RFC 2679, "95 percent was chosen because ... a particular confidence level should be specified so that the results of independent implementations can be compared."

Two different implementations are expected to produce statistically equivalent results if they both measure a metric under the same networking conditions. Formulating in statistical terms: separate metric implementations collect separate samples from the same underlying statistical process (the same network conditions). The statistical hypothesis to be tested is the expectation that both samples do not expose statistically different properties. This requires careful test design:

- The measurement test setup must be self-consistent to the largest possible extent. To minimize the influence of the test and measurement setup on the result, network conditions and paths MUST be identical for the compared implementations to the largest possible degree. This includes both the stability and non-ambiguity of routes taken by the measurement packets. See RFC 2330 for a discussion on self-consistency.

- The error induced by the sample size must be small enough to minimize its influence on the test result. This may have to be respected, especially if two implementations measure with

different average probing rates.

- Every comparison must be repeated several times based on different measurement data to avoid random indications of compatibility (or the lack of it).

- To minimize the influence of implementation options on the result, metric implementations SHOULD use identical options and parameters for the metric under evaluation.

- The implementation with the lowest probing frequency determines the smallest temporal interval for which samples can be compared.

The metric specifications themselves are the primary focus of evaluation, rather than the implementations of metrics. The documentation produced by the advancement process should identify which metric definitions and supporting material were found to be clearly worded and unambiguous, OR, it should identify ways in which the metric specification text should be revised to achieve clarity and unified interpretation.

The process should also permit identification of options that were not implemented, so that they can be removed from the advancing specification (this is an aspect more typical of protocol advancement along the standards track).

Note that this document does not propose to base interoperability indications of performance metric implementations on comparisons of individual singletons. Individual singletons may be impacted by many statistical effects while they are measured. Comparing two singletons of different implementations may result in failures with higher probability than comparing samples.

3. Verification of conformance to a metric specification

This section specifies how to verify compliance of two or more IPPM implementations against a metric specification. This document only proposes a general methodology. Compliance criteria to a specific metric implementation need to be defined for each individual metric specification. The only exception is the statistical test comparing two metric implementations which are simultaneously tested. This test is applicable without metric specific decision criteria.

Several testing options exist to compare two or more implementations:

- o Use a single test lab to compare the implementations and emulate the Internet with an impairment generator.

- o Use a single test lab to compare the implementations and measure across the Internet.

- o Use remotely separated test labs to compare the implementations and emulate the Internet with two "identically" configured impairment generators.

- o Use remotely separated test labs to compare the implementations and measure across the Internet.

- o Use remotely separated test labs to compare the implementations and measure across the Internet and include a single impairment generator to impact all measurement flows in non discriminatory way.

The first two approaches work, but cause higher expenses than the other ones (due to travel and/or shipping+installation). For the third option, ensuring two identically configured impairment generators requires well defined test cases and possibly identical hard- and software. >>>Comment: for some specific tests, impairment generator accuracy requirements are less-demanding than others, and in such cases there is more flexibility in impairment generator configuration. <<<

It is a fair question, whether the last two options can result in any applicable test set up at all. While an experimental approach is given in Appendix C, the trade off that measurement packets of different sites pass the path segments but always in a different order of segments probably can’t be avoided.

The question of which option above results in identical networking conditions and is broadly accepted can’t be answered without more practical experience in comparing implementations. The last proposal has the advantage that, while the measurement equipment is remotely distributed, a single network impairment generator and the Internet can be used in combination to impact all measurement flows.

3.1. Tests of an individual implementation against a metric specification

A metric implementation MUST support the requirements classified as "MUST" and "REQUIRED" of the related metric specification to be compliant to the latter.

Further, supported options of a metric implementation SHOULD be

documented in sufficient detail. The documentation of chosen options is RECOMMENDED to minimise (and recognise) differences in the test setup if two metric implementations are compared. Further, this documentation is used to validate and improve the underlying metric specification option, to remove options which saw no implementation or which are badly specified from the metric specification to be promoted to a standard. This documentation SHOULD be made for all implementation-relevant specifications of a metric picked for a comparison that are not explicitly marked as "MUST" or "REQUIRED" in the RFC text. This applies for the following sections of all metric specifications:

- Singleton Definition of the Metric.
- Sample Definition of the Metric.
- Statistics Definition of the Metric. As statistics are compared by the test specified here, this documentation is required even in the case, that the metric specification does not contain a Statistics Definition.
- Timing and Synchronisation related specification (if relevant for the Metric).
- Any other technical part present or missing in the metric specification, which is relevant for the implementation of the Metric.

RFC2330 and RFC2679 emphasise precision as an aim of IPPM metric implementations. A single IPPM conformant implementation MUST under otherwise identical network conditions produce precise results for repeated measurements of the same metric.

RFC 2330 prefers the "empirical distribution function" EDF to describe collections of measurements. RFC 2330 determines, that "unless otherwise stated, IPPM goodness-of-fit tests are done using 5% significance." The goodness of fit test determines by which precision two or more samples of a metric implementation belong to the same underlying distribution (of measured network performance events). The goodness of fit test to be applied is the Anderson-Darling K sample test (ADK sample test, K stands for the number of samples to be compared) [ADK]. Please note that RFC 2330 and RFC 2679 apply an Anderson Darling goodness of fit test too.

The results of a repeated test with a single implementation MUST pass an ADK sample test with confidence level of 95%. The resolution for which the ADK test has been passed with the specified confidence level MUST be documented. To formulate this differently: The

requirement is to document the smallest resolution, at which the results of the tested metric implementation pass an ADK test with a confidence level of 95%. The minimum resolution available in the reported results from each implementation MUST be taken into account in the ADK test.

3.2. Test setup resulting in identical live network testing conditions

Two major issues complicate tests for metric compliance across live networks under identical testing conditions. One is the general point that metric definition implementations cannot be conveniently examined in field measurement scenarios. The other one is more broadly described as "parallelism in devices and networks", including mechanisms like those that achieve load balancing (see [RFC4928]).

This section proposes two measures to deal with both issues. Tunneling mechanisms can be used to avoid parallel processing of different flows in the network. Measuring by separate parallel probe flows results in repeated collection of data. If both measures are combined, WAN network conditions are identical for a number of independent measurement flows, no matter what the network conditions are in detail.

Any measurement setup MUST be made to avoid the probing traffic itself to impede the metric measurement. The created measurement load MUST NOT result in congestion at the access link connecting the measurement implementation to the WAN. The created measurement load MUST NOT overload the measurement implementation itself, e.g., by causing a high CPU load or by creating imprecisions due to internal transmit (receive respectively) probe packet collisions.

Tunneling multiple flows reaching a network element on a single physical port may allow to transmit all packets of the tunnel via the same path. Applying tunnels to avoid undesired influence of standard routing for measurement purposes is a concept known from literature, see e.g. GRE encapsulated multicast probing [GU+Duffield]. An existing IP in IP tunnel protocol can be applied to avoid Equal-Cost Multi-Path (ECMP) routing of different measurement streams if it meets the following criteria:

- Inner IP packets from different measurement implementations are mapped into a single tunnel with single outer IP origin and destination address as well as origin and destination port numbers which are identical for all packets.

- An easily accessible commodity tunneling protocol allows to carry out a metric test from more test sites.

- o A low operational overhead may enable a broader audience to set up a metric test with the desired properties.

- o The tunneling protocol should be reliable and stable in set up and operation to avoid disturbances or influence on the test results.

- o The tunneling protocol should not incur any extra cost for those interested in setting up a metric test.

An illustration of a test setup with two tunnels and two flows between two linecards of one implementation is given in Figure 1.

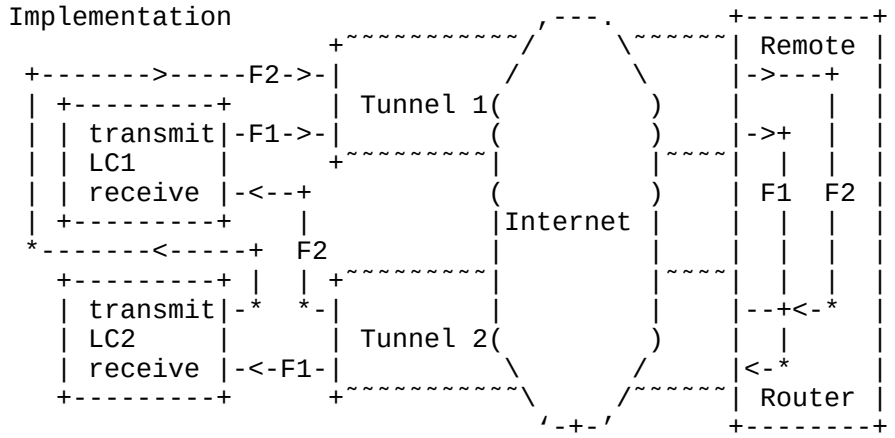


Illustration of a test setup with two tunnels. For simplicity, only two linecards of one implementation and two flows F between them are shown.

Figure 1

Figure 2 shows the network elements required to set up GRE tunnels or as shown by figure 1.

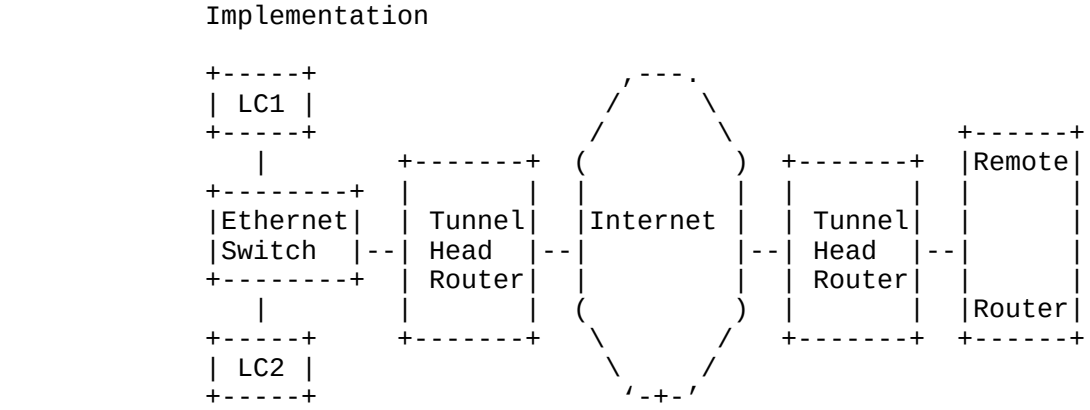


Illustration of a hardware setup to realise the test setup illustrated by figure 1 with GRE tunnels or Pseudowires.

Figure 2

If tunneling is applied, two tunnels MUST carry all test traffic in between the test site and the remote site. For example, if 802.1Q Ethernet Virtual LANs (VLAN) are applied and the measurement streams are carried in different VLANs, the IP tunnel or Pseudo Wires respectively MUST be set up in physical port mode to avoid set up of Pseudo Wires per VLAN (which may see different paths due to ECMP routing), see RFC 4448. The remote router and the Ethernet switch shown in figure 2 must support 802.1Q in this set up.

The IP packet size of the metric implementation SHOULD be chosen small enough to avoid fragmentation due to the added Ethernet and tunnel headers. Otherwise, the impact of tunnel overhead on fragmentation and interface MTU size MUST be understood and taken into account (see [RFC4459]).

An Ethernet port mode IP tunnel carrying several 802.1Q VLANs each containing measurement traffic of a single measurement system was set up as a proof of concept using RFC4719 [RFC4719], Transport of Ethernet Frames over L2TPv3. Ethernet over L2TPv3 seems to fulfill most of the desired tunneling protocol criteria mentioned above.

The following headers may have to be accounted for when calculating total packet length, if VLANs and Ethernet over L2TPv3 tunnels are applied:

- Ethernet 802.1Q: 22 Byte.
- L2TPv3 Header: 4-16 Byte for L2TPv3 data messages over IP; 16-28 Byte for L2TPv3 data messages over UDP.

- IPv4 Header (outer IP header): 20 Byte.

- MPLS Labels may be added by a carrier. Each MPLS Label has a length of 4 Bytes. By the time of writing, between 1 and 4 Labels seems to be a fair guess of what's expectable.

The applicability of one or more of the following tunneling protocols may be investigated by interested parties if Ethernet over L2TPv3 is felt to be not suitable: IP in IP [RFC2003] or Generic Routing Encapsulation (GRE) [RFC2784]. RFC 4928 [RFC4928] proposes measures how to avoid ECMP treatment in MPLS networks.

L2TP is a commodity tunneling protocol [RFC2661]. By the time of writing, L2TPv3 [RFC3931]is the latest version of L2TP. If L2TPv3 is applied, software based implementations of this protocol are not suitable for the test set up, as such implementations may cause incalculable delay shifts.

Ethernet Pseudo Wires may also be set up on MPLS networks [RFC4448]. While there's no technical issue with this solution, MPLS interfaces are mostly found in the network provider domain. Hence not all of the above tunneling criteria are met.

Appendix C provides an experimental tunneling set up for metric implementation testing between two (or more) remote sites.

Each test SHOULD be conducted multiple times. Sequential testing is possible, but may not be a useful metric test option because WAN conditions are likely to change over time. It is RECOMMENDED that tests be carried out by establishing at least 2 different parallel measurement flows. Two linecards per implementation that send and receive measurement flows should be sufficient to create 4 parallel measurement flows (when each card sends and receives 2 flows). Other options are to separate flows by DiffServ marks (without deploying any QoS in the inner or outer tunnel) or using a single CBR flow and evaluating every n-th singleton to belong to a specific measurement flow.

Some additional rules to calculate and compare samples have to be respected to perform a metric test:

- To compare different probes of a common underlying distribution in terms of metrics characterising a communication network requires to respect the temporal nature for which the assumption of common underlying distribution may hold. Any singletons or samples to be compared MUST be captured within the same time interval.

- Whenever statistical events like singletons or rates are used to characterise measured metrics of a time-interval, at least 5 singletons of a relevant metric SHOULD be present to ensure a minimum confidence into the reported value (see Wikipedia on confidence [Rule of thumb]). Note that this criterion also is to be respected e.g. when comparing packet loss metrics. Any packet loss measurement interval to be compared with the results of another implementation SHOULD contain at least five lost packets to have a minimum confidence that the observed loss rate wasn't caused by a small number of random packet drops.

- The minimum number of singletons or samples to be compared by an Anderson-Darling test SHOULD be 100 per tested metric implementation. Note that the Anderson-Darling test detects small differences in distributions fairly well and will fail for high number of compared results (RFC2330 mentions an example with 8192 measurements where an Anderson-Darling test always failed).

- Generally, the Anderson-Darling test is sensitive to differences in the accuracy or bias associated with varying implementations or test conditions. These dissimilarities may result in differing averages of samples to be compared. An example may be different packet sizes, resulting in a constant delay difference between compared samples. Therefore samples to be compared by an Anderson-Darling test MAY be calibrated by the difference of the average values of the samples. Any calibration of this kind MUST be documented in the test result.

3.3. Tests of two or more different implementations against a metric specification

RFC2330 expects "a methodology for a given metric [to] exhibit continuity if, for small variations in conditions, it results in small variations in the resulting measurements. Slightly more precisely, for every positive epsilon, there exists a positive delta, such that if two sets of conditions are within delta of each other, then the resulting measurements will be within epsilon of each other." A small variation in conditions in the context of the metric test proposed here can be seen as different implementations measuring the same metric along the same path.

IPPM metric specifications however allow for implementor options to the largest possible degree. It can not be expected that two implementors pick identical value ranges in options for the implementations. Implementors SHOULD to the highest degree possible pick the same configurations for their systems when comparing their implementations by a metric test.

In some cases, a goodness of fit test may not be possible or show disappointing results. To clarify the difficulties arising from different implementation options, the individual options picked for every compared implementation SHOULD be documented in sufficient detail. Based on this documentation, the underlying metric specification should be improved before it is promoted to a standard.

The same statistical test as applicable to quantify precision of a single metric implementation MUST be used to compare metric result equivalence for different implementations. To document compatibility, the smallest measurement resolution at which the compared implementations passed the ADK sample test MUST be documented.

For different implementations of the same metric, "variations in conditions" are reasonably expected. The ADK test comparing samples of the different implementations MAY result in a lower precision than the test for precision in the same-implementation comparison.

3.4. Clock synchronisation

Clock synchronization effects require special attention. Accuracy of one-way active delay measurements for any metrics implementation depends on clock synchronization between the source and destination of tests. Ideally, one-way active delay measurement (RFC 2679, [RFC2679]) test endpoints either have direct access to independent GPS or CDMA-based time sources or indirect access to nearby NTP primary (stratum 1) time sources, equipped with GPS receivers. Access to these time sources may not be available at all test locations associated with different Internet paths, for a variety of reasons out of scope of this document.

When secondary (stratum 2 and above) time sources are used with NTP running across the same network, whose metrics are subject to comparative implementation tests, network impairments can affect clock synchronization, distort sample one-way values and their interval statistics. It is RECOMMENDED to discard sample one-way delay values for any implementation, when one of the following reliability conditions is met:

- Delay is measured and is finite in one direction, but not the other.
- Absolute value of the difference between the sum of one-way measurements in both directions and round-trip measurement is greater than X% of the latter value.

Examination of the second condition requires RTT measurement for

reference, e.g., based on TWAMP (RFC5357, RFC 5357 [RFC5357]), in conjunction with one-way delay measurement.

Specification of X% to strike a balance between identification of unreliable one-way delay samples and misidentification of reliable samples under a wide range of Internet path RTTs probably requires further study.

An implementation of an RFC that requires synchronized clocks is expected to provide precise measurement results in order to claim that the metric measured is compliant.

IF an implementation publishes a specification of its precision, such as "a precision of 1 ms (+/- 500 us) with a confidence of 95%", then the specification SHOULD be met over a useful measurement duration. For example, if the metric is measured along an Internet path which is stable and not congested, then the precision specification SHOULD be met over durations of an hour or more.

3.5. Recommended Metric Verification Measurement Process

In order to meet their obligations under the IETF Standards Process the IESG must be convinced that each metric specification advanced to Draft Standard or Internet Standard status is clearly written, that there are the a sufficient number of verified equivalent implementations, and that all options have been implemented.

In the context of this document, metrics are designed to measure some characteristic of a data network. An aim of any metric definition should be that it should be specified in a way that can reliably measure the specific characteristic in a repeatable way across multiple independent implementations.

Each metric, statistic or option of those to be validated MUST be compared against a reference measurement or another implementation by at least 5 different basic data sets, each one with sufficient size to reach the specified level of confidence, as specified by this document.

Finally, the metric definitions, embodied in the text of the RFCs, are the objects that require evaluation and possible revision in order to advance to the next step on the standards track.

IF two (or more) implementations do not measure an equivalent metric as specified by this document,

AND sources of measurement error do not adequately explain the lack of agreement,

THEN the details of each implementation should be audited along with the exact definition text, to determine if there is a lack of clarity that has caused the implementations to vary in a way that affects the correspondence of the results.

IF there was a lack of clarity or multiple legitimate interpretations of the definition text,

THEN the text should be modified and the resulting memo proposed for consensus and (possible) advancement along the standards track.

Finally, all the findings MUST be documented in a report that can support advancement on the standards track, similar to those described in [RFC5657]. The list of measurement devices used in testing satisfies the implementation requirement, while the test results provide information on the quality of each specification in the metric RFC (the surrogate for feature interoperability).

The complete process of advancing a metric specification to a standard as defined by this document is illustrated in Figure 3.

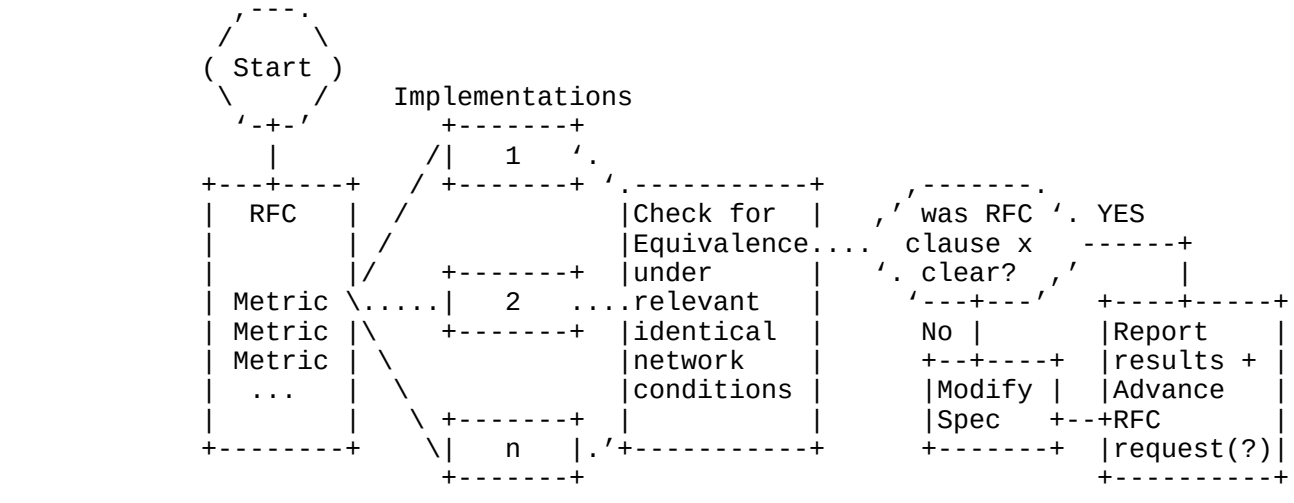


Illustration of the metric standardisation process

Figure 3

Any recommendation for the advancement of a metric specification MUST

be accompanied by an implementation report, as is the case with all requests for the advancement of IETF specifications. The implementation report needs to include the tests performed, the applied test setup, the specific metrics in the RFC and reports of the tests performed with two or more implementations. The test plan needs to specify the precision reached for each measured metric and thus define the meaning of "statistically equivalent" for the specific metrics being tested.

Ideally, the test plan would co-evolve with the development of the metric, since that’s when people have the most context in their thinking regarding the different subtleties that can arise.

In particular, the implementation report MUST as a minimum document:

- The metric compared and the RFC specifying it. This includes statements as required by the section "Tests of an individual implementation against a metric specification" of this document.

- The measurement configuration and setup.

- A complete specification of the measurement stream (mean rate, statistical distribution of packets, packet size or mean packet size and their distribution), DSCP and any other measurement stream properties which could result in deviating results. Deviations in results can be caused also if chosen IP addresses and ports of different implementations can result in different layer 2 or layer 3 paths due to operation of Equal Cost Multi-Path routing in an operational network.

- The duration of each measurement to be used for a metric validation, the number of measurement points collected for each metric during each measurement interval (i.e. the probe size) and the level of confidence derived from this probe size for each measurement interval.

- The result of the statistical tests performed for each metric validation as required by the section "Tests of two or more different implementations against a metric specification" of this document.

- A parameterization of laboratory conditions and applied traffic and network conditions allowing reproduction of these laboratory conditions for readers of the implementation report.

- The documentation helping to improve metric specifications defined by this section.

All of the tests for each set SHOULD be run in a test setup as specified in the section "Test setup resulting in identical live network testing conditions."

If a different test set up is chosen, it is RECOMMENDED to avoid effects falsifying results of validation measurements caused by real data networks (like parallelism in devices and networks). Data networks may forward packets differently in the case of:

- Different packet sizes chosen for different metric implementations. A proposed countermeasure is selecting the same packet size when validating results of two samples or a sample against an original distribution.

- Selection of differing IP addresses and ports used by different metric implementations during metric validation tests. If ECMP is applied on IP or MPLS level, different paths can result (note that it may be impossible to detect an MPLS ECMP path from an IP endpoint). A proposed counter measure is to connect the measurement equipment to be compared by a NAT device, or establishing a single tunnel to transport all measurement traffic The aim is to have the same IP addresses and port for all measurement packets or to avoid ECMP based local routing diversion by using a layer 2 tunnel.

- Different IP options.

- Different DSCP.

- If the N measurements are captured using sequential measurements instead of simultaneous ones, then the following factors come into play: Time varying paths and load conditions.

3.6. Miscellaneous

A minimum amount of singletons per metric is required if results are to be compared. To avoid accidental singletons from impacting a metric comparison, a minimum number of 5 singletons per compared interval was proposed above. Commercial Internet service is not operated to reliably create enough rare events of singletons to characterize bad measurement engineering or bad implementations. In the case that a metric validation requires capturing rare events, an impairment generator may have to be added to the test set up. Inclusion of an impairment generator and the parameterisation of the impairments generated MUST be documented.

A metric characterising a common impairment condition would be one, which by expectation creates a singleton result for each measured

packet. Delay or Delay Variation are examples of this type, and in such cases, the Internet may be used to compare metric implementations.

Rare events are those, where by expectation no or a rather low number of "event is present" singletons are captured during a measurement interval. Packet duplications, packet loss rates above one digit percentages, loss patterns and packet reordering are examples. Note especially that a packet reordering or loss pattern metric implementation comparison may require a more sophisticated test set up than described here. Spatial and temporal effects combine in the case of packet re-ordering and measurements with different packet rates may always lead to different results.

As specified above, 5 singletons are the recommended basis to minimise interference of random events with the statistical test proposed by this document. In the case of ratio measurements (like packet loss), the underlying sum of basic events, against the which the metric's monitored singletons are "rated", determines the resolution of the test. A packet loss statistic with a resolution of 1% requires one packet loss statistic-data point to consist of 500 delay singletons (of which at least 5 were lost). To compare EDFs on packet loss requires one hundred such statistics per flow. That means, all in all at least 50 000 delay singletons are required per single measurement flow. Live network packet loss is assumed to be present during main traffic hours only. Let this interval be 5 hours. The required minimum rate of a single measurement flow in that case is 2.8 packets/sec (assuming a loss of 1% during 5 hours). If this measurement is too demanding under live network conditions, an impairment generator should be used.

3.7. Proposal to determine an "equivalence" threshold for each metric evaluated

This section describes a proposal for maximum error of "equivalence", based on performance comparison of identical implementations. This comparison may be useful for both ADK and non-ADK comparisons.

Each metric tested by two or more implementations (cross-implementation testing).

Each metric is also tested twice simultaneously by the *same* implementation, using different Src/Dst Address pairs and other differences such that the connectivity differences of the cross-implementation tests are also experienced and measured by the same implementation.

Comparative results for the same implementation represent a bound on

cross-implementation equivalence. This should be particularly useful when the metric does *not* produces a continuous distribution of singleton values, such as with a loss metric, or a duplication metric. Appendix A indicates how the ADK will work for 0ne-way delay, and should be likewise applicable to distributions of delay variation.

Proposal: the implementation with the largest difference in homogeneous comparison results is the lower bound on the equivalence threshold, noting that there may be other systematic errors to account for when comparing between implementations.

Thus, when evaluating equivalence in cross-implementation results:

Maximum_Error = Same_Implementation_Error + Systematic_Error

and only the systematic error need be decided beforehand.

In the case of ADK comparison, the largest same-implementation resolution of distribution equivalence can be used as a limit on cross-implementation resolutions (at the same confidence level).

4. Acknowledgements

Gerhard Hasslinger commented a first version of this document, suggested statistical tests and the evaluation of time series information. Henk Uijterwaal and Lars Eggert have encouraged and helped to orgainize this work. Mike Hamilton, Scott Bradner, David Mcdysan and Emile Stephan commented on this draft. Carol Davids reviewed the 01 version of the ID before it was promoted to WG draft.

5. Contributors

Scott Bradner, Vern Paxson and Allison Mankin drafted bradner-metrictest [bradner-metrictest], and major parts of it are included in this document.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

This draft does not raise any specific security issues.

8. References

8.1. Normative References

[RFC2003] Perkins, C., "IP Encapsulation within IP", RFC 2003, October 1996.

[RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

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

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

[RFC2661] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G., and B. Palter, "Layer Two Tunneling Protocol "L2TP"", RFC 2661, August 1999.

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

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

[RFC2681] Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, September 1999.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.

[RFC3931] Lau, J., Townsley, M., and I. Goyret, "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, March 2005.

[RFC4448] Martini, L., Rosen, E., El-Aawar, N., and G. Heron, "Encapsulation Methods for Transport of Ethernet over MPLS Networks", RFC 4448, April 2006.

[RFC4459] Savola, P., "MTU and Fragmentation Issues with In-the-Network Tunneling", RFC 4459, April 2006.

[RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.

[RFC4719] Aggarwal, R., Townsley, M., and M. Dos Santos, "Transport of Ethernet Frames over Layer 2 Tunneling Protocol Version 3 (L2TPv3)", RFC 4719, November 2006.

[RFC4928] Swallow, G., Bryant, S., and L. Andersson, "Avoiding Equal Cost Multipath Treatment in MPLS Networks", BCP 128, RFC 4928, June 2007.

[RFC5657] Dusseault, L. and R. Sparks, "Guidance on Interoperation and Implementation Reports for Advancement to Draft Standard", BCP 9, RFC 5657, September 2009.

8.2. Informative References

[ADK] Scholz, F. and M. Stephens, "K-sample Anderson-Darling Tests of fit, for continuous and discrete cases", University of Washington, Technical Report No. 81, May 1986.

[GU+Duffield] Gu, Y., Duffield, N., Breslau, L., and S. Sen, "GRE Encapsulated Multicast Probing: A Scalable Technique for Measuring One-Way Loss", SIGMETRICS'07 San Diego, California, USA, June 2007.

[RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.

[Rule of thumb] Hardy, M., "Confidence interval", March 2010.

[bradner-metrictest] Bradner, S., Mankin, A., and V. Paxson, "Advancement of metrics specifications on the IETF Standards Track", draft -bradner-metricstest-03, (work in progress), July 2007.

[morton-advance-metrics] Morton, A., "Problems and Possible Solutions for Advancing Metrics on the Standards Track", draft -morton-ippm-advance-metrics-00, (work in progress), July 2009.

[morton-advance-metrics-01] Morton, A., "Lab Test Results for Advancing Metrics on the Standards Track", draft -morton-ippm-advance-metrics-01, (work in progress), June 2010.

Appendix A. An example on a One-way Delay metric validation

The text of this appendix is not binding. It is an example how parts of a One-way Delay metric test could look like. http://xml.resource.org/public/rfc/bibxml/

A.1. Compliance to Metric specification requirements

One-way Delay, Loss threshold, RFC 2679

This test determines if implementations use the same configured maximum waiting time delay from one measurement to another under different delay conditions, and correctly declare packets arriving in excess of the waiting time threshold as lost. See Section 3.5 of RFC2679, 3rd bullet point and also Section 3.8.2 of RFC2679.

(1) Configure a path with 1 sec one-way constant delay.

(2) Measure one-way delay with 2 or more implementations, using identical waiting time thresholds for loss set at 2 seconds.

(3) Configure the path with 3 sec one-way delay.

(4) Repeat measurements.

(5) Observe that the increase measured in step 4 caused all packets to be declared lost, and that all packets that arrive successfully in step 2 are assigned a valid one-way delay.

One-way Delay, First-bit to Last bit, RFC 2679

This test determines if implementations register the same relative increase in delay from one measurement to another under different delay conditions. This test tends to cancel the sources of error which may be present in an implementation. See Section 3.7.2 of RFC2679, and Section 10.2 of RFC2330.

(1) Configure a path with X ms one-way constant delay, and ideally including a low-speed link.

(2) Measure one-way delay with 2 or more implementations, using identical options and equal size small packets (e.g., 100 octet IP payload).

(3) Maintain the same path with X ms one-way delay.

(4) Measure one-way delay with 2 or more implementations, using identical options and equal size large packets (e.g., 1500 octet IP payload).

(5) Observe that the increase measured in steps 2 and 4 is equivalent to the increase in ms expected due to the larger serialization time for each implementation. Most of the measurement errors in each system should cancel, if they are stationary.

One-way Delay, RFC 2679

This test determines if implementations register the same relative increase in delay from one measurement to another under different delay conditions. This test tends to cancel the sources of error which may be present in an implementation. This test is intended to evaluate measurments in sections 3 and 4 of RFC2679.

(1) Configure a path with X ms one-way constant delay.

(2) Measure one-way delay with 2 or more implementations, using identical options.

(3) Configure the path with X+Y ms one-way delay.

(4) Repeat measurements.

(5) Observe that the increase measured in steps 2 and 4 is ~Y ms for each implementation. Most of the measurement errors in each system should cancel, if they are stationary.

Error Calibration, RFC 2679

This is a simple check to determine if an implementation reports the error calibration as required in Section 4.8 of RFC2679. Note that the context (Type-P) must also be reported.

A.2. Examples related to statistical tests for One-way Delay

A one way delay measurement may pass an ADK test with a timestamp resultion of 1 ms. The same test may fail, if timestamps with a resolution of 100 microseconds are eavluated. The implementation then is then conforming to the metric specification up to a timestamp resolution of 1 ms.

Let’s assume another one way delay measurement comparison between implementation 1, probing with a frequency of 2 probes per second and implementation 2 probing at a rate of 2 probes every 3 minutes. To

ensure reasonable confidence in results, sample metrics are calculated from at least 5 singletons per compared time interval. This means, sample delay values are calculated for each system for identical 6 minute intervals for the whole test duration. Per 6 minute interval, the sample metric is calculated from 720 singletons for implementation 1 and from 6 singletons for implementation 2. Note, that if outliers are not filtered, moving averages are an option for an evaluation too. The minimum move of an averaging interval is three minutes in this example.

The data in table 1 may result from measuring One-Way Delay with implementation 1 (see column Implemnt_1) and implementation 2 (see column implemnt_2). Each data point in the table represents a (rounded) average of the sampled delay values per interval. The resolution of the clock is one micro-second. The difference in the delay values may result eg. from different probe packet sizes.

Implemnt_1	Implemnt_2	Implemnt_2 - Delta_Averages
5000	6549	4997
5008	6555	5003
5012	6564	5012
5015	6565	5013
5019	6568	5016
5022	6570	5018
5024	6573	5021
5026	6575	5023
5027	6577	5025
5029	6580	5028
5030	6585	5033
5032	6586	5034
5034	6587	5035
5036	6588	5036
5038	6589	5037
5039	6591	5039
5041	6592	5040
5043	6599	5047
5046	6606	5054
5054	6612	5060

Table 1

Average values of sample metrics captured during identical time intervals are compared. This excludes random differences caused by differing probing intervals or differing temporal distance of singletons resulting from their Poisson distributed sending times.

In the example, 20 values have been picked (note that at least 100 values are recommended for a single run of a real test). Data must be ordered by ascending rank. The data of Implemnt_1 and Implemnt_2 as shown in the first two columns of table 1 clearly fails an ADK test with 95% confidence.

The results of Implemnt_2 are now reduced by difference of the averages of column 2 (rounded to 6581 us) and column 1 (rounded to 5029 us), which is 1552 us. The result may be found in column 3 of table 1. Comparing column 1 and column 3 of the table by an ADK test shows, that the data contained in these columns passes an ADK tests with 95% confidence.

>>> Comment: Extensive averaging was used in this example, because of the vastly different sampling frequencies. As a result, the distributions compared do not exactly align with a metric in [RFC2679], but illustrate the ADK process adequately.

Appendix B. Anderson-Darling 2 sample C++ code

```
/* Routines for computing the Anderson-Darling 2 sample
 * test statistic.
 *
 * Implemented based on the description in
 * "Anderson-Darling K Sample Test" Heckert, Alan and
 * Filliben, James, editors, Dataplot Reference Manual,
 * Chapter 15 Auxiliary, NIST, 2004.
 * Official Reference by 2010
 * Heckert, N. A. (2001). Dataplot website at the
 * National Institute of Standards and Technology:
 * http://www.itl.nist.gov/div898/software/dataplot.html/
 * June 2001.
 */

#include <iostream>
#include <fstream>
#include <vector>
#include <sstream>

using namespace std;

vector<double> vec1, vec2;
double adk_result;
double adk_criterium = 1.993;

/* vec1 and vec2 to be initialised with sample 1 and
```

```
             * sample 2 values in ascending order.
             */

            /* example for iterating the vectors
             * for(vector<double>::iterator it = vec1->begin();
             * it != vec1->end(); it++
             * {
             * cout << *it << endl;
             * }
             */

            static int k, val_st_z_samp1, val_st_z_samp2,
                       val_eq_z_samp1, val_eq_z_samp2,
                           j, n_total, n_sample1, n_sample2, L,
                       max_number_samples, line, maxnumber_z;
            static int column_1, column_2;
            static double adk, n_value, z, sum_adk_samp1,
                   sum_adk_samp2, z_aux;
            static double H_j, F1j, hj, F2j, denom_1_aux, denom_2_aux;
            static bool next_z_sample2, equal_z_both_samples;
            static int stop_loop1, stop_loop2, stop_loop3,old_eq_line2,
                   old_eq_line1;

            static double adk_criterium = 1.993;

            k = 2;
            n_sample1 = vec1->size() - 1;
            n_sample2 = vec2->size() - 1;

            // -1 because vec[0] is a dummy value

            n_total = n_sample1 + n_sample2;

            /* value equal to the line with a value = zj in sample 1.
             * Here j=1, so the line is 1.
             */

            val_eq_z_samp1 = 1;

            /* value equal to the line with a value = zj in sample 2.
             * Here j=1, so the line is 1.
             */

            val_eq_z_samp2 = 1;

            /* value equal to the last line with a value < zj
             * in sample 1. Here j=1, so the line is 0.
             */
```

```
            val_st_z_samp1 = 0;

            /* value equal to the last line with a value < zj
             * in sample 1. Here j=1, so the line is 0.
             */

            val_st_z_samp2 = 0;

            sum_adk_samp1 = 0;
            sum_adk_samp2 = 0;
            j = 1;

            // as mentioned above, j=1

            equal_z_both_samples = false;
            next_z_sample2 = false;

            //assuming the next z to be of sample 1

            stop_loop1 = n_sample1 + 1;

            // + 1 because vec[0] is a dummy, see n_sample1 declaration

            stop_loop2 = n_sample2 + 1;
            stop_loop3 = n_total + 1;

            /* The required z values are calculated until all values
             * of both samples have been taken into account. See the
             * lines above for the stoploop values. Construct required
             * to avoid a mathematical operation in the While condition
             */

             while (((stop_loop1 > val_eq_z_samp1)
                   || (stop_loop2 > val_eq_z_samp2)) && stop_loop3 > j)
               {
               if(val_eq_z_samp1 < n_sample1+1)
                   {

            /* here, a preliminary zj value is set.
             * See below how to calculate the actual zj.
             */

                    z = (*vec1)[val_eq_z_samp1];

            /* this while sequence calculates the number of values
             * equal to z.
             */
```

```
                while ((val_eq_z_samp1+1 < n_sample1)
                        && z == (*vec1)[val_eq_z_samp1+1] )
                        {
                        val_eq_z_samp1++;
                        }
                }
                else
                {
                val_eq_z_samp1 = 0;
                val_st_z_samp1 = n_sample1;

        // this should be val_eq_z_samp1 - 1 = n_sample1
                }

        if(val_eq_z_samp2 < n_sample2+1)
                {
                z_aux = (*vec2)[val_eq_z_samp2];;

        /* this while sequence calculates the number of values
         * equal to z_aux
         */

                while ((val_eq_z_samp2+1 < n_sample2)
                        && z_aux == (*vec2)[val_eq_z_samp2+1] )
                        {
                        val_eq_z_samp2++;
                        }

        /* the smaller of the two actual data values is picked
         * as the next zj.
         */

           if(z > z_aux)
                        {
                        z = z_aux;
                        next_z_sample2 = true;
                        }
                 else
                        {
                        if (z == z_aux)
                        {
                        equal_z_both_samples = true;
                        }

        /* This is the case, if the last value of column1 is
         * smaller than the remaining values of column2.
         */
                        if (val_eq_z_samp1 == 0)
```

```
                        {
                        z = z_aux;
                        next_z_sample2 = true;
                        }
                    }
                  }
                else
                  {
                  val_eq_z_samp2 = 0;
                  val_st_z_samp2 = n_sample2;

              // this should be val_eq_z_samp2 - 1 = n_sample2

                  }

              /* in the following, sum j = 1 to L is calculated for
               * sample 1 and sample 2.
               */

                 if (equal_z_both_samples)
                   {

              /* hj is the number of values in the combined sample
               * equal to zj
               */
                        hj = val_eq_z_samp1 - val_st_z_samp1
                     + val_eq_z_samp2 - val_st_z_samp2;

              /* H_j is the number of values in the combined sample
               * smaller than zj plus one half the the number of
               * values in the combined sample equal to zj
               * (that's hj/2).
               */

                        H_j = val_st_z_samp1 + val_st_z_samp2
                              + hj / 2;

              /* F1j is the number of values in the 1st sample
               * which are less than zj plus one half the number
               * of values in this sample which are equal to zj.
               */

                        F1j = val_st_z_samp1 + (double)
                    (val_eq_z_samp1 - val_st_z_samp1) / 2;

              /* F2j is the number of values in the 1st sample
               * which are less than zj     plus one half the number
               * of values in this sample which are equal to zj.
```

```
 */
                F2j = val_st_z_samp2 + (double)
        (val_eq_z_samp2 - val_st_z_samp2) / 2;

      /* set the line of values equal to zj to the
       * actual line of the last value picked for zj.
       */
                val_st_z_samp1 = val_eq_z_samp1;

      /* Set the line of values equal to zj to the actual
       * line of the last value picked for zjof each
       * sample. This is required as data smaller than zj
       * is accounted differently than values equal to zj.
       */

           val_st_z_samp2 = val_eq_z_samp2;

      /* next the lines of the next values z, ie. zj+1
       * are addressed.
       */

        val_eq_z_samp1++;

      /* next the lines of the next values z, ie.
       * zj+1 are addressed
       */

                val_eq_z_samp2++;
                }
           else
                {

      /* the smaller z value was contained in sample 2,
       * hence this value is the zj to base the following
       * calculations on.
       */
                if (next_z_sample2)
                {

      /* hj is the number of values in the combined
       * sample equal to zj, in this case these are
       * within sample 2 only.
       */
                hj = val_eq_z_samp2 - val_st_z_samp2;

      /* H_j is the number of values in the combined sample
       * smaller than zj plus one half the the number of
       * values in the combined sample equal to zj
```

```
             * (that's hj/2).
             */

                            H_j = val_st_z_samp1 + val_st_z_samp2
                          + hj / 2;

            /* F1j is the number of values in the 1st sample which
             * are less than zj plus one half the number of values in
             * this sample which are equal to zj.
             * As val_eq_z_samp2 < val_eq_z_samp1, these are the
             * val_st_z_samp1 only.
             */
                        F1j = val_st_z_samp1;

            /* F2j is the number of values in the 1st sample which
             * are less than zj plus one half the number of values in
             * this sample which are equal to zj. The latter are from
             * sample 2 only in this case.
             */

                 F2j = val_st_z_samp2 + (double)
                      (val_eq_z_samp2 - val_st_z_samp2) / 2;

            /* Set the line of values equal to zj to the actual line
             * of the last value picked for zj of sample 2 only in
             * this case.
             */
                            val_st_z_samp2 = val_eq_z_samp2;

            /* next the line of the next value z, ie. zj+1 is
             * addressed. Here, only sample 2 must be addressed.
             */

                 val_eq_z_samp2++;
                                if (val_eq_z_samp1 == 0)
                                {
                                val_eq_z_samp1 = stop_loop1;
                                }
                        }

            /* the smaller z value was contained in sample 2,
             * hence this value is the zj to base the following
             * calculations on.
             */

                        else
                        {
```

```
            /* hj is the number of values in the combined
             * sample equal to zj, in this case these are
             * within sample 1 only.
             */
                        hj = val_eq_z_samp1 - val_st_z_samp1;

            /* H_j is the number of values in the combined
             * sample smaller than zj plus one half the the number
             * of values in the combined sample equal to zj
             * (that's hj/2).
             */

                 H_j = val_st_z_samp1 + val_st_z_samp2
                       + hj / 2;

            /* F1j is the number of values in the 1st sample which
             * are less than zj plus, in this case these are within
             * sample 1 only one half the number of values in this
             * sample which are equal to zj. The latter are from
             * sample 1 only in this case.
             */

                 F1j = val_st_z_samp1 + (double)
                      (val_eq_z_samp1 - val_st_z_samp1) / 2;

            /* F2j is the number of values in the 1st sample which
             * are less than zj plus one half the number of values
             * in this sample which are equal to zj. As
             * val_eq_z_samp1 < val_eq_z_samp2, these are the
             * val_st_z_samp2 only.
             */

                        F2j = val_st_z_samp2;

            /* Set the line of values equal to zj to the actual line
             * of the last value picked for zj of sample 1 only in
             * this case
             */

                 val_st_z_samp1 = val_eq_z_samp1;

            /* next the line of the next value z, ie. zj+1 is
             * addressed. Here, only sample 1 must be addressed.
             */
                        val_eq_z_samp1++;

                        if (val_eq_z_samp2 == 0)
                                {
```

```
                                val_eq_z_samp2 = stop_loop2;
                                }
                        }
                        }

                denom_1_aux = n_total * F1j - n_sample1 * H_j;
                denom_2_aux = n_total * F2j - n_sample2 * H_j;

                sum_adk_samp1 = sum_adk_samp1 + hj
                        * (denom_1_aux * denom_1_aux) /
                                        (H_j * (n_total - H_j)
                        - n_total * hj / 4);
                sum_adk_samp2 = sum_adk_samp2 + hj
                        * (denom_2_aux * denom_2_aux) /
                                        (H_j * (n_total - H_j)
                        - n_total * hj / 4);

                next_z_sample2 = false;
                equal_z_both_samples = false;

        /* index to count the z. It is only required to prevent
         * the while slope to execute endless
         */
                j++;
                }

        // calculating the adk value is the final step.

        adk_result = (double) (n_total - 1) / (n_total
                * n_total * (k - 1))
                * (sum_adk_samp1 / n_sample1
                + sum_adk_samp2 / n_sample2);


        /* if(adk_result <= adk_criterium)
         * adk_2_sample test is passed
         */
```

Figure 4

Appendix C. A tunneling set up for remote metric implementation testing

Parties interested in testing metric compliance is most convenient if all involved parties can stay in their local test laboratories. Figure 4 shows a test configuration which may enable remote metric compliance testing.

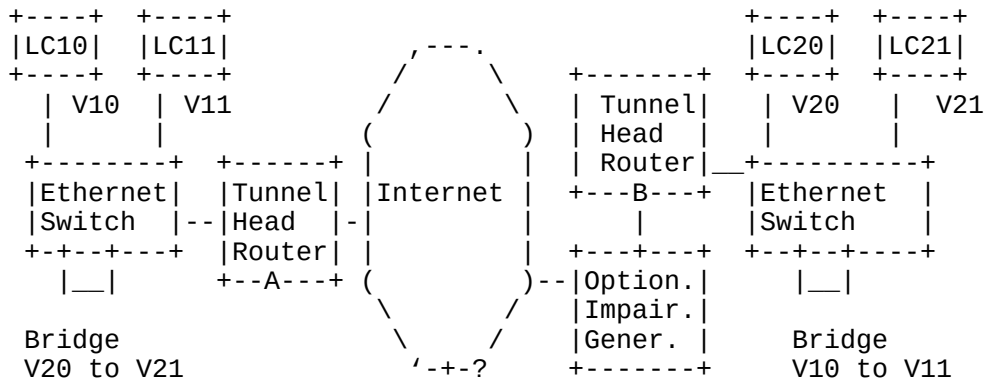


Figure 5

LC10 identify measurement clients /line cards. V10 and the others denote VLANs. All VLANs are using the same tunnel from A to B and in the reverse direction. The remote site VLANs are U-bridged at the local site Ethernet switch. The measurement packets of site 1 travel tunnel A->B first, are U-bridged at site 2 and travel tunnel B->A second. Measurement packets of site 2 travel tunnel B->A first, are U-bridged at site 1 and travel tunnel A->B second. So all measurement packets pass the same tunnel segments, but in different segment order. An experiment to prove or reject the above test set up shown in figure 4 has been agreed but not yet scheduled between Deutsche Telekom and RIPE.

Figure 4 includes an optional impairment generator. If this impairment generator is inserted in the IP path between the tunnel head end routers, it equally impacts all measurement packets and flows. Thus trouble with ensuring identical test set up by configuring two separated impairment generators identically is avoided (which was another proposal allowing remote metric compliance testing).

Appendix D. Glossary

ADK	Anderson-Darling K-Sample test, a test used to check whether two samples have the same statistical distribution.
ECMP	Equal Cost Multipath, a load balancing mechanism evaluating MPLS labels stacks, IP addresses and ports.
EDF	The "Empirical Distribution Function" of a set of scalar measurements is a function F(x) which for any x gives the fractional proportion of the total measurements that were smaller than or equal as x.
Metric	A measured quantity related to the performance and reliability of the Internet, expressed by a value. This could be a singleton (single value), a sample of single values or a statistic based on a sample of singletons.
OWAMP	One-way Active Measurement Protocol, a protocol for communication between IPPM measurement systems specified by IPPM.
OWD	One-Way Delay, a performance metric specified by IPPM.
Sample metric	A sample metric is derived from a given singleton metric by evaluating a number of distinct instances together.
Singleton metric	A singleton metric is, in a sense, one atomic measurement of this metric.
Statistical metric	A ’statistical’ metric is derived from a given sample metric by computing some statistic of the values defined by the singleton metric on the sample.
TWAMP	Two-way Active Measurement Protocol, a protocol for communication between IPPM measurement systems specified by IPPM.

Table 2

Authors' Addresses

Ruediger Geib (editor)
Deutsche Telekom
Heinrich Hertz Str. 3-7
Darmstadt, 64295
Germany

Phone: +49 6151 628 2747
Email: Ruediger.Geib@telekom.de

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/

Reza Fardid
Cariden Technologies
888 Villa Street, Suite 500
Mountain View, CA 94041
USA

Phone:
Email: rfardid@cariden.com

Alexander Steinmitz
HS Fulda
Marquardstr. 35
Fulda, 36039
Germany

Phone:
Email: steinionline@gmx.de

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 28, 2011

A. Morton
G. Ramachandran
G. Maguluri
AT&T Labs

October 25, 2010

Reporting Metrics: Different Points of View
draft-ietf-ippm-reporting-metrics-04

Abstract

Consumers of IP network performance metrics have many different uses in mind. The memo provides "long-term" reporting considerations (e.g, days, weeks or months, as opposed to 10 seconds), based on analysis of the two key audience points-of-view. It describes how the audience categories affect the selection of metric parameters and options when seeking info that serves their needs.

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 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 April 28, 2011.

Copyright Notice

Table of Contents

1. Introduction

When designing measurements of IP networks and presenting the results, knowledge of the audience is a key consideration. To present a useful and relevant portrait of network conditions, one must answer the following question:

"How will the results be used?"

There are two main audience categories:

1. Network Characterization - describes conditions in an IP network for quality assurance, troubleshooting, modeling, Service Level Agreements (SLA), etc. The point-of-view looks inward, toward the network, and the consumer intends their actions there.

2. Application Performance Estimation - describes the network conditions in a way that facilitates determining affects on user applications, and ultimately the users themselves. This point-of-view looks outward, toward the user(s), accepting the network as-is. This consumer intends to estimate a network-dependent aspect of performance, or design some aspect of an application's accommodation of the network. (These are *not* application metrics, they are defined at the IP layer.)

This memo considers how these different points-of-view affect both the measurement design (parameters and options of the metrics) and statistics reported when serving their needs.

The IPPM framework [RFC2330] and other RFCs describing IPPM metrics provide a background for this memo.

2. Purpose and Scope

The purpose of this memo is to clearly delineate two points-of-view (POV) for using measurements, and describe their effects on the test design, including the selection of metric parameters and reporting the results.

The scope of this memo primarily covers the design and reporting of the loss and delay metrics [RFC2680] [RFC2679]. It will also discuss the delay variation [RFC3393] and reordering metrics [RFC4737] where applicable.

With capacity metrics growing in relevance to the industry, the memo also covers POV and reporting considerations for metrics resulting from the Bulk Transfer Capacity Framework [RFC3148] and Network

Capacity Definitions [RFC5136]. These memos effectively describe two different categories of metrics,

- o [RFC3148] with congestion flow-control and the notion of unique data bits delivered, and
- o [RFC5136] using a definition of raw capacity without the restrictions of data uniqueness or congestion-awareness.

It might seem at first glance that each of these metrics has an obvious audience (Raw = Network Characterization, Restricted = Application Performance), but reality is more complex and consistent with the overall topic of capacity measurement and reporting. For example, TCP is usually used in Restricted capacity measurement methods, while UDP appears in Raw capacity measurement. The Raw and Restricted capacity metrics will be treated in separate sections, although they share one common reporting issue: representing variability in capacity metric results as part of a long-term report.

Sampling, or the design of the active packet stream that is the basis for the measurements, is also discussed.

3. Reporting Results

This section gives an overview of recommendations, followed by additional considerations for reporting results in the "long-term", based on the discussion and conclusions of the major sections that follow.

3.1. Overview of Metric Statistics

This section gives an overview of reporting recommendations for the loss, delay, and delay variation metrics.

The minimal report on measurements MUST include both Loss and Delay Metrics.

For Packet Loss, the loss ratio defined in [RFC2680] is a sufficient starting point, especially the guidance for setting the loss threshold waiting time. We have calculated a waiting time above that should be sufficient to differentiate between packets that are truly lost or have long finite delays under general measurement circumstances, 51 seconds. Knowledge of specific conditions can help to reduce this threshold, but 51 seconds is considered to be manageable in practice.

We note that a loss ratio calculated according to [Y.1540] would

exclude errored packets from the numerator. In practice, the difference between these two loss metrics is small if any, depending on whether the last link prior to the destination contributes errored packets.

For Packet Delay, we recommend providing both the mean delay and the median delay with lost packets designated undefined (as permitted by [RFC2679]). Both statistics are based on a conditional distribution, and the condition is packet arrival prior to a waiting time dT, where dT has been set to take maximum packet lifetimes into account, as discussed below. Using a long dT helps to ensure that delay distributions are not truncated.

For Packet Delay Variation (PDV), the minimum delay of the conditional distribution should be used as the reference delay for computing PDV according to [Y.1540] or [RFC5481] and [RFC3393]. A useful value to report is a pseudo range of delay variation based on calculating the difference between a high percentile of delay and the minimum delay. For example, the 99.9%-ile minus the minimum will give a value that can be compared with objectives in [Y.1541].

3.2. Long-Term Reporting Considerations

[I-D.ietf-ippm-reporting] describes methods to conduct measurements and report the results on a near-immediate time scale (10 seconds, which we consider to be "short-term").

Measurement intervals and reporting intervals need not be the same length. Sometimes, the user is only concerned with the performance levels achieved over a relatively long interval of time (e.g, days, weeks, or months, as opposed to 10 seconds). However, there can be risks involved with running a measurement continuously over a long period without recording intermediate results:

- Temporary power failure may cause loss of all the results to date.
- Measurement system timing synchronization signals may experience a temporary outage, causing sub-sets of measurements to be in error or invalid.
- Maintenance may be necessary on the measurement system, or its connectivity to the network under test.

For these and other reasons, such as

- the constraint to collect measurements on intervals similar to user session length, or

- o the dual-use of measurements in monitoring activities where results are needed on a period of a few minutes,

there is value in conducting measurements on intervals that are much shorter than the reporting interval.

There are several approaches for aggregating a series of measurement results over time in order to make a statement about the longer reporting interval. One approach requires the storage of all metric singletons collected throughout the reporting interval, even though the measurement interval stops and starts many times.

Another approach is described in [RFC5835] as "temporal aggregation". This approach would estimate the results for the reporting interval based on many individual measurement interval statistics (results) alone. The result would ideally appear in the same form as though a continuous measurement was conducted. A memo to address the details of temporal aggregation is yet to be prepared.

Yet another approach requires a numerical objective for the metric, and the results of each measurement interval are compared with the objective. Every measurement interval where the results meet the objective contribute to the fraction of time with performance as specified. When the reporting interval contains many measurement intervals it is possible to present the results as "metric A was less than or equal to objective X during Y% of time.

NOTE that numerical thresholds of acceptability are not set in IETF performance work and are explicitly excluded from the IPPM charter.

In all measurement, it is important to avoid unintended synchronization with network events. This topic is treated in [RFC2330] for Poisson-distributed inter-packet time streams, and [RFC3432] for Periodic streams. Both avoid synchronization through use of random start times.

There are network conditions where it is simply more useful to report the connectivity status of the Source-Destination path, and to distinguish time intervals where connectivity can be demonstrated from other time intervals (where connectivity does not appear to exist). [RFC2678] specifies a number of one-way and two connectivity metrics of increasing complexity. In this memo, we RECOMMEND that long term reporting of loss, delay, and other metrics be limited to time intervals where connectivity can be demonstrated, and other intervals be summarized as percent of time where connectivity does not appear to exist. We note that this same approach has been adopted in ITU-T Recommendation [Y.1540] where performance parameters are only valid during periods of service "availability" (evaluated

according to a function based on packet loss, and sustained periods of loss ratio greater than a threshold are declared "unavailable").

4. Effect of POV on the Loss Metric

This section describes the ways in which the Loss metric can be tuned to reflect the preferences of the two audience categories, or different POV. The waiting time to declare a packet lost, or loss threshold is one area where there would appear to be a difference, but the ability to post-process the results may resolve it.

4.1. Loss Threshold

RFC 2680 [RFC2680] defines the concept of a waiting time for packets to arrive, beyond which they are declared lost. The text of the RFC declines to recommend a value, instead saying that "good engineering, including an understanding of packet lifetimes, will be needed in practice." Later, in the methodology, they give reasons for waiting "a reasonable period of time", and leaving the definition of "reasonable" intentionally vague.

4.1.1. Network Characterization

Practical measurement experience has shown that unusual network circumstances can cause long delays. One such circumstance is when routing loops form during IGP re-convergence following a failure or drastic link cost change. Packets will loop between two routers until new routes are installed, or until the IPv4 Time-to-Live (TTL) field (or the IPv6 Hop Limit) decrements to zero. Very long delays on the order of several seconds have been measured [Casner] [Cia03].

Therefore, network characterization activities prefer a long waiting time in order to distinguish these events from other causes of loss (such as packet discard at a full queue, or tail drop). This way, the metric design helps to distinguish more reliably between packets that might yet arrive, and those that are no longer traversing the network.

It is possible to calculate a worst-case waiting time, assuming that a routing loop is the cause. We model the path between Source and Destination as a series of delays in links (t) and queues (q), as these two are the dominant contributors to delay. The normal path delay across n hops without encountering a loop, D, is

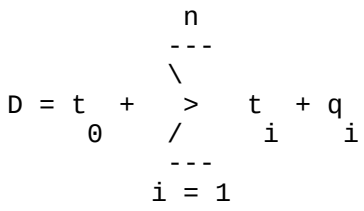


Figure 1: Normal Path Delay

and the time spent in the loop with L hops, is

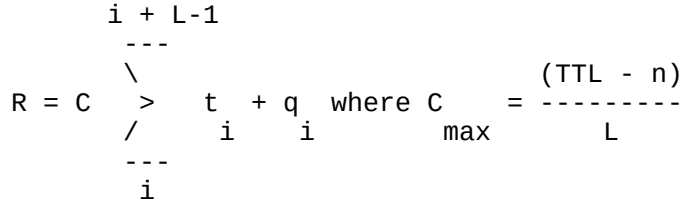


Figure 2: Delay due to Rotations in a Loop

and where C is the number of times a packet circles the loop.

If we take the delays of all links and queues as 100ms each, the TTL=255, the number of hops n=5 and the hops in the loop L=4, then

D = 1.1 sec and R ˜= 50 sec, and D + R ˜= 51.1 seconds

We note that the link delays of 100ms would span most continents, and a constant queue length of 100ms is also very generous. When a loop occurs, it is almost certain to be resolved in 10 seconds or less. The value calculated above is an upper limit for almost any realistic circumstance.

A waiting time threshold parameter, dT, set consistent with this calculation would not truncate the delay distribution (possibly causing a change in its mathematical properties), because the packets that might arrive have been given sufficient time to traverse the network.

It is worth noting that packets that are stored and deliberately forwarded at a much later time constitute a replay attack on the measurement system, and are beyond the scope of normal performance reporting.

4.1.2. Application Performance

Fortunately, application performance estimation activities are not adversely affected by the estimated worst-case transfer time. Although the designer's tendency might be to set the Loss Threshold at a value equivalent to a particular application's threshold, this specific threshold can be applied when post-processing the measurements. A shorter waiting time can be enforced by locating packets with delays longer than the application's threshold, and re-designating such packets as lost. Thus, the measurement system can use a single loss threshold and support both application and network performance POVs simultaneously.

4.2. Errored Packet Designation

RFC 2680 designates packets that arrive containing errors as lost packets. Many packets that are corrupted by bit errors are discarded within the network and do not reach their intended destination.

This is consistent with applications that would check the payload integrity at higher layers, and discard the packet. However, some applications prefer to deal with errored payloads on their own, and even a corrupted payload is better than no packet at all.

To address this possibility, and to make network characterization more complete, it is recommended to distinguish between packets that do not arrive (lost) and errored packets that arrive (conditionally lost).

4.3. Causes of Lost Packets

Although many measurement systems use a waiting time to determine if a packet is lost or not, most of the waiting is in vain. The packets are no-longer traversing the network, and have not reached their destination.

There are many causes of packet loss, including:

1. Queue drop, or discard

2. Corruption of the IP header, or other essential header info

3. TTL expiration (or use of a TTL value that is too small)

4. Link or router failure

After waiting sufficient time, packet loss can probably be attributed to one of these causes.

4.4. Summary for Loss

Given that measurement post-processing is possible (even encouraged in the definitions of IPPM metrics), measurements of loss can easily serve both points of view:

- Use a long waiting time to serve network characterization and revise results for specific application delay thresholds as needed.

- Distinguish between errored packets and lost packets when possible to aid network characterization, and combine the results for application performance if appropriate.

5. Effect of POV on the Delay Metric

This section describes the ways in which the Delay metric can be tuned to reflect the preferences of the two consumer categories, or different POV.

5.1. Treatment of Lost Packets

The Delay Metric [RFC2679] specifies the treatment of packets that do not successfully traverse the network: their delay is undefined.

" >>The *Type-P-One-way-Delay* from Src to Dst at T is undefined (informally, infinite)<< means that Src sent the first bit of a Type-P packet to Dst at wire-time T and that Dst did not receive that packet."

It is an accepted, but informal practice to assign infinite delay to lost packets. We next look at how these two different treatments align with the needs of measurement consumers who wish to characterize networks or estimate application performance. Also, we look at the way that lost packets have been treated in other metrics: delay variation and reordering.

5.1.1. Application Performance

Applications need to perform different functions, dependent on whether or not each packet arrives within some finite tolerance. In other words, a receivers' packet processing takes one of two directions (or "forks" in the road):

- Packets that arrive within expected tolerance are handled by processes that remove headers, restore smooth delivery timing (as in a de-jitter buffer), restore sending order, check for errors in

payloads, and many other operations.

- Packets that do not arrive when expected spawn other processes that attempt recovery from the apparent loss, such as retransmission requests, loss concealment, or forward error correction to replace the missing packet.

So, it is important to maintain a distinction between packets that actually arrive, and those that do not. Therefore, it is preferable to leave the delay of lost packets undefined, and to characterize the delay distribution as a conditional distribution (conditioned on arrival).

5.1.2. Network Characterization

In this discussion, we assume that both loss and delay metrics will be reported for network characterization (at least).

Assume packets that do not arrive are reported as Lost, usually as a fraction of all sent packets. If these lost packets are assigned undefined delay, then network's inability to deliver them (in a timely way) is captured only in the loss metric when we report statistics on the Delay distribution conditioned on the event of packet arrival (within the Loss waiting time threshold). We can say that the Delay and Loss metrics are Orthogonal, in that they convey non-overlapping information about the network under test.

However, if we assign infinite delay to all lost packets, then:

- The delay metric results are influenced both by packets that arrive and those that do not.

- The delay singleton and the loss singleton do not appear to be orthogonal (Delay is finite when Loss=0, Delay is infinite when Loss=1).

- The network is penalized in both the loss and delay metrics, effectively double-counting the lost packets.

As further evidence of overlap, consider the Cumulative Distribution Function (CDF) of Delay when the value positive infinity is assigned to all lost packets. Figure 3 shows a CDF where a small fraction of packets are lost.

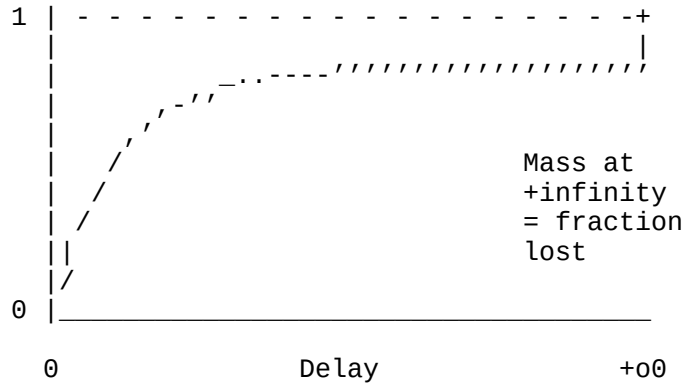


Figure 3: Cumulative Distribution Function for Delay when Loss = +Infinity

We note that a Delay CDF that is conditioned on packet arrival would not exhibit this apparent overlap with loss.

Although infinity is a familiar mathematical concept, it is somewhat disconcerting to see any time-related metric reported as infinity, in the opinion of the authors. Questions are bound to arise, and tend to detract from the goal of informing the consumer with a performance report.

5.1.3. Delay Variation

[RFC3393] excludes lost packets from samples, effectively assigning an undefined delay to packets that do not arrive in a reasonable time. Section 4.1 describes this specification and its rationale (ipdv = inter-packet delay variation in the quote below).

"The treatment of lost packets as having "infinite" or "undefined" delay complicates the derivation of statistics for ipdv. Specifically, when packets in the measurement sequence are lost, simple statistics such as sample mean cannot be computed. One possible approach to handling this problem is to reduce the event space by conditioning. That is, we consider conditional statistics; namely we estimate the mean ipdv (or other derivative statistic) conditioned on the event that selected packet pairs arrive at the destination (within the given timeout). While this itself is not without problems (what happens, for example, when every other packet is lost), it offers a way to make some (valid) statements about ipdv, at the same time avoiding events with undefined outcomes."

We note that the argument above applies to all forms of packet delay variation that can be constructed using the "selection function"

concept of [RFC3393]. In recent work the two main forms of delay variation metrics have been compared and the results are summarized in [RFC5481].

5.1.4. Reordering

[RFC4737]defines metrics that are based on evaluation of packet arrival order, and include a waiting time to declare a packet lost (to exclude them from further processing).

If packets are assigned a delay value, then the reordering metric would declare any packets with infinite delay to be reordered, because their sequence numbers will surely be less than the "Next Expected" threshold when (or if) they arrive. But this practice would fail to maintain orthogonality between the reordering metric and the loss metric. Confusion can be avoided by designating the delay of non-arriving packets as undefined, and reserving delay values only for packets that arrive within a sufficiently long waiting time.

5.2. Preferred Statistics

Today in network characterization, the sample mean is one statistic that is almost ubiquitously reported. It is easily computed and understood by virtually everyone in this audience category. Also, the sample is usually filtered on packet arrival, so that the mean is based a conditional distribution.

The median is another statistic that summarizes a distribution, having somewhat different properties from the sample mean. The median is stable in distributions with a few outliers or without them. However, the median’s stability prevents it from indicating when a large fraction of the distribution changes value. 50% or more values would need to change for the median to capture the change.

Both the median and sample mean have difficulty with bimodal distributions. The median will reside in only one of the modes, and the mean may not lie in either mode range. For this and other reasons, additional statistics such as the minimum, maximum, and 95%-ile have value when summarizing a distribution.

When both the sample mean and median are available, a comparison will sometimes be informative, because these two statistics are equal only when the delay distribution is perfectly symmetrical.

Also, these statistics are generally useful from the Application Performance POV, so there is a common set that should satisfy audiences.

Plots of the delay distribution may also be useful when single-value statistics indicate that new conditions are present. An empirically-derived probability distribution function will usually describe multiple modes more efficiently than any other form of result.

5.3. Summary for Delay

From the perspectives of:

1. application/receiver analysis, where subsequent processing depends on whether the packet arrives or times-out,

2. straightforward network characterization without double-counting defects, and

3. consistency with Delay variation and Reordering metric definitions,

the most efficient practice is to distinguish between truly lost and delayed packets with a sufficiently long waiting time, and to designate the delay of non-arriving packets as undefined.

6. Effect of POV on Raw Capacity Metrics

This section describes the ways that raw capacity metrics can be tuned to reflect the preferences of the two audiences, or different Points-of-View (POV). Raw capacity refers to the metrics defined in [RFC5136] which do not include restrictions such as data uniqueness or flow-control response to congestion.

In summary, the metrics considered are IP-layer Capacity, Utilization (or used capacity), and Available Capacity, for individual links and complete paths. These three metrics form a triad: knowing one metric constrains the other two (within their allowed range), and knowing two determines the third. The link metrics have another key aspect in common: they are single-measurement-point metrics at the egress of a link. The path Capacity and Available Capacity are derived by examining the set of single-point link measurements and taking the minimum value.

6.1. Type-P Parameter

The concept of "packets of type-P" is defined in [RFC2330]. The type-P categorization has critical relevance in all forms of capacity measurement and reporting. The ability to categorize packets based on header fields for assignment to different queues and scheduling mechanisms is now common place. When un-used resources are shared

across queues, the conditions in all packet categories will affect capacity and related measurements. This is one source of variability in the results that all audiences would prefer to see reported in a useful and easily understood way.

Type-P in OWAMP and TWAMP is essentially confined to the Diffserv Codepoint [ref]. DSCP is the most common qualifier for type-P.

Each audience will have a set of type-P qualifications and value combinations that are of interest. Measurements and reports SHOULD have the flexibility to per-type and aggregate performance.

6.2. a priori Factors

The audience for Network Characterization may have detailed information about each link that comprises a complete path (due to ownership, for example), or some of the links in the path but not others, or none of the links.

There are cases where the measurement audience only has information on one of the links (the local access link), and wishes to measure one or more of the raw capacity metrics. This scenario is quite common, and has spawned a substantial number of experimental measurement methods [ref to CAIDA survey page, etc.]. Many of these methods respect that their users want a result fairly quickly and in a one-trial. Thus, the measurement interval is kept short (a few seconds to a minute). For long-term reporting, a sample of short term results need to be summarized.

6.3. IP-layer Capacity

For links, this metric’s theoretical maximum value can be determined from the physical layer bit rate and the bit rate reduction due to the layers between the physical layer and IP. When measured, this metric takes additional factors into account, such as the ability of the sending device to process and forward traffic under various conditions. For example, the arrival of routing updates may spawn high priority processes that reduce the sending rate temporarily. Thus, the measured capacity of a link will be variable, and the maximum capacity observed applies to a specific time, time interval, and other relevant circumstances.

For paths composed of a series of links, it is easy to see how the sources of variability for the results grow with each link in the path. Results variability will be discussed in more detail below.

6.4. IP-layer Utilization

The ideal metric definition of Link Utilization [RFC5136] is based on the actual usage (bits successfully received during a time interval) and the Maximum Capacity for the same interval.

In practice, Link Utilization can be calculated by counting the IP-layer (or other layer) octets received over a time interval and dividing by the theoretical maximum of octets that could have been delivered in the same interval. A commonly used time interval is 5 minutes, and this interval has been sufficient to support network operations and design for some time. 5 minutes is somewhat long compared with the expected download time for web pages, but short with respect to large file transfers and TV program viewing. It is fair to say that considerable variability is concealed by reporting a single (average) Utilization value for each 5 minute interval. Some performance management systems have begun to make 1 minute averages available.

There is also a limit on the smallest useful measurement interval. Intervals on the order of the serialization time for a single Maximum Transmission Unit (MTU) packet will observe on/off behavior and report 100% or 0%. The smallest interval needs to be some multiple of MTU serialization time for averaging to be effective.

6.5. IP-layer Available Capacity

The Available Capacity of a link can be calculated using the Capacity and Utilization metrics.

When Available capacity of a link or path is estimated through some measurement technique, the following parameters SHOULD be reported:

- Name and reference to the exact method of measurement
- IP packet length, octets (including IP header)
- Maximum Capacity that can be assessed in the measurement configuration
- The time a duration of the measurement
- All other parameters specific to the measurement method

Many methods of Available capacity measurement have a maximum capacity that they can measure, and this maximum may be less than the actual Available capacity of the link or path. Therefore, it is important to know the capacity value beyond which there will be no

measured improvement.

The Application Design audience may have a target capacity value and simply wish to assess whether there is sufficient Available Capacity. This case simplifies measurement of link and path capacity to some degree, as long as the measurable maximum exceeds the target capacity.

6.6. Variability in Utilization and Avail. Capacity

As with most metrics and measurements, assessing the consistency or variability in the results gives a the user an intuitive feel for the degree (or confidence) that any one value is representative of other results, or the underlying distribution from which these singleton measurements have come.

Two questions are raised here for further discussion:

What ways can Utilization be measured and summarized to describe the potential variability in a useful way?

How can the variability in Available Capacity estimates be reported, so that the confidence in the results is also conveyed?

7. Effect of POV on Restricted Capacity Metrics

This section describes the ways that restricted capacity metrics can be tuned to reflect the preferences of the two audiences, or different Points-of-View (POV). Raw capacity refers to the metrics defined in [RFC3148] which include restrictions such as data uniqueness or flow-control response to congestion.

In primary metric considered is Bulk Transfer Capacity (BTC) for complete paths. [RFC3148] defines

BTC = data_sent / elapsed_time

for a connection with congestion-aware flow control, where data_sent is the total of unique payload bits (no headers).

We note that this definition *differs* from the raw capacity definition in Section 2.3.1 of [RFC5136], where IP-layer Capacity *includes* all bits in the IP header and payload. This means that Restricted Capacity BTC is already operating at a disadvantage when compared to the raw capacity at layers below TCP. Further, there are cases where "THE IP-layer" is encapsulated in another IP-layer or other form of tunneling protocol, designating more and more of the

fundamental transport capacity as header bits that are pure overhead to the BTC measurement.

When thinking about the triad of raw capacity metrics, BTC is most akin to the "IP-Type-P Available Path Capacity", at least in the eyes of a network user who seeks to know what transmission performance a path might support.

7.1. Type-P Parameter and Type-C Parameter

The concept of "packets of type-P" is defined in [RFC2330]. The considerations for Restricted Capacity are identical to the raw capacity section on this topic, with the addition that the various fields and options in the TCP header MUST be included in the description.

The vast array of TCP flow control options are not well-captured by Type-P, because they do not exist in the TCP header bits. Therefore, we introduce a new notion here: TCP Configuration of "Type-C". The elements of Type-C describe all of the settings for TCP options and congestion control algorithm variables, including the main form of congestion control in use.

7.2. a priori Factors

The audience for Network Characterization may have detailed information about each link that comprises a complete path (due to ownership, for example), or some of the links in the path but not others, or none of the links.

There are cases where the measurement audience only has information on one of the links (the local access link), and wishes to measure one or more BTC metrics. This scenario is quite common, and has spawned a substantial number of experimental measurement methods [ref to CAIDA survey page, etc.]. Many of these methods respect that their users want a result fairly quickly and in a one-trial. Thus, the measurement interval is kept short (a few seconds to a minute). For long-term reporting, a sample of short term results need to be summarized.

7.3. Measurement Interval

There are limits on a useful measurement interval for BTC. Three factors that influence the interval duration are listed below:

1. Measurements may choose to include or exclude the 3-way handshake of TCP connection establishment, which requires at least 1.5 * RTT and contains both the delay of the path and the host

processing time for responses. However, user experience includes the 3-way handshake for all new TCP connections.

2. Measurements may choose to include or exclude Slow-Start, preferring instead to focus on a portion of the transfer that represents "equilibrium" <<<< which needs a definition for this purpose >>>>. However, user experience includes the Slow-Start for all new TCP connections.

3. Measurements may choose to use a fixed block of data to transfer, where the size of the block has a relationship to the file size of the application of interest. This approach yields variable size measurement intervals, where a path faster BTC is measured for less time than a slower path, an this has implications when path impairments are time-varying, or transient. Users are likely to turn their immediate attention elsewhere when a very large file must be transferred, thus they do not directly experience such a long transfer -- they see the result (success or fail) and possibly an objective measurement of the transfer time (which will likely include the 3-way handshake, Slow-start, and application file management processing time as well as the BTC).

Individual measurement intervals may be short or long, but there is a need to report the results on a long-term basis that captures the BTC variability experienced between each interval. Consistent BTC is a valuable commodity along with the value attained.

7.4. Bulk Transfer Capacity Reporting

When BTC of a link or path is estimated through some measurement technique, the following parameters SHOULD be reported:

- Name and reference to the exact method of measurement
- Maximum Transmission Unit (MTU)
- Maximum BTC that can be assessed in the measurement configuration
- The time and duration of the measurement
- The number of BTC connections used simultaneously
- *All* other parameters specific to the measurement method, especially the Congestion Control algorithm in use

See also
[http://tools.ietf.org/wg/ippm/draft-ietf-ippm-tcp-throughput-tm/]

Many methods of Bulk Transfer Capacity measurement have a maximum capacity that they can measure, and this maximum may be less than the available capacity of the link or path. Therefore, it is important to specify the measured BTC value beyond which there will be no measured improvement.

The Application Design audience may have a target capacity value and simply wish to assess whether there is sufficient BTC. This case simplifies measurement of link and path capacity to some degree, as long as the measurable maximum exceeds the target capacity.

7.5. Variability in Bulk Transfer Capacity

As with most metrics and measurements, assessing the consistency or variability in the results gives a the user an intuitive feel for the degree (or confidence) that any one value is representative of other results, or the underlying distribution from which these singleton measurements have come.

Two questions are raised here for further discussion:

What ways can BTC be measured and summarized to describe the potential variability in a useful way?

How can the variability in BTC estimates be reported, so that the confidence in the results is also conveyed?

8. Test Streams and Sample Size

This section discusses two key aspects of measurement that are sometimes omitted from the report: the description of the test stream on which the measurements are based, and the sample size.

8.1. Test Stream Characteristics

Network Characterization has traditionally used Poisson-distributed inter-packet spacing, as this provides an unbiased sample. The average inter-packet spacing may be selected to allow observation of specific network phenomena. Other test streams are designed to sample some property of the network, such as the presence of congestion, link bandwidth, or packet reordering.

If measuring a network in order to make inferences about applications or receiver performance, then there are usually efficiencies derived from a test stream that has similar characteristics to the sender. In some cases, it is essential to synthesize the sender stream, as with Bulk Transfer Capacity estimates. In other cases, it may be

sufficient to sample with a "known bias", e.g., a Periodic stream to estimate real-time application performance.

8.2. Sample Size

Sample size is directly related to the accuracy of the results, and plays a critical role in the report. Even if only the sample size (in terms of number of packets) is given for each value or summary statistic, it imparts a notion of the confidence in the result.

In practice, the sample size will be selected taking both statistical and practical factors into account. Among these factors are:

1. The estimated variability of the quantity being measured

2. The desired confidence in the result (although this may be dependent on assumption of the underlying distribution of the measured quantity).

3. The effects of active measurement traffic on user traffic

4. etc.

A sample size may sometimes be referred to as "large". This is a relative, and qualitative term. It is preferable to describe what one is attempting to achieve with their sample. For example, stating an implication may be helpful: this sample is large enough such that a single outlying value at ten times the "typical" sample mean (the mean without the outlying value) would influence the mean by no more than X.

9. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

10. Security Considerations

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

11. Acknowledgements

The authors thank: Phil Chimento for his suggestion to employ conditional distributions for Delay, Steve Konish Jr. for his careful review and suggestions, Dave Mcdysan and Don McLachlan for useful comments based on their long experience with measurement and reporting, and Matt Zekauskas for suggestions on organizing the memo for easier consumption.

12. References

12.1. Normative References

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

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

[RFC2678] Mahdavi, J. and V. Paxson, "IPPM Metrics for Measuring Connectivity", RFC 2678, September 1999.

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

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

[RFC3148] Mathis, M. and M. Allman, "A Framework for Defining Empirical Bulk Transfer Capacity Metrics", RFC 3148, July 2001.

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

[RFC3432] Raisanen, V., Grotefeld, G., and A. Morton, "Network performance measurement with periodic streams", RFC 3432, November 2002.

[RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.

[RFC4737] Morton, A., Ciavattone, L., Ramachandran, G., Shalunov, S., and J. Perser, "Packet Reordering Metrics", RFC 4737,

November 2006.

[RFC5136] Chimento, P. and J. Ishac, "Defining Network Capacity", RFC 5136, February 2008.

12.2. Informative References

[Casner] "A Fine-Grained View of High Performance Networking, NANOG 22 Conf.; http://www.nanog.org/mtg-0105/agenda.html", May 20-22 2001.

[Cia03] "Standardized Active Measurements on a Tier 1 IP Backbone, IEEE Communications Mag., pp 90-97.", June 2003.

[I-D.ietf-ippm-reporting] Shalunov, S. and M. Swany, "Reporting IP Performance Metrics to Users", draft-ietf-ippm-reporting-05 (work in progress), July 2010.

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

[RFC5835] Morton, A. and S. Van den Berghe, "Framework for Metric Composition", RFC 5835, April 2010.

[Y.1540] ITU-T Recommendation Y.1540, "Internet protocol data communication service - IP packet transfer and availability performance parameters", December 2002.

[Y.1541] ITU-T Recommendation Y.1540, "Network Performance Objectives for IP-Based Services", February 2006.

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/

Gomathi Ramachandran
AT&T Labs
200 Laurel Avenue South
Middletown, New Jersey 07748
USA

Phone: +1 732 420 2353
Fax:
Email: gomathi@att.com
URI:

Ganga Maguluri
AT&T Labs
200 Laurel Avenue
Middletown, New Jersey 07748
USA

Phone: 732-420-2486
Fax:
Email: gmaguluri@att.com
URI:

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 9, 2011

A. Morton
AT&T Labs

October 6, 2010

IMIX Genome: Specification of variable packet sizes for additional testing
draft-morton-bmwg-imix-genome-00

Abstract

Benchmarking Methodologies have always relied on test conditions with constant packet sizes, with the goal of understanding what network device capability has been tested. Constant packets sizes differ significantly from the conditions encountered in operational deployment, and so additional tests are sometimes conducted with a mixture of packet sizes, or "IMIX". The mixture of sizes a networking device will encounter is highly variable and depends on many factors. An IMIX suited for one networking device and deployment will not be appropriate for another. However, the mix of sizes may be known and the tester may be asked to augment the fixed size tests. To address this need, and the additional goal of repeatable test conditions, this draft proposes a way to specify the exact repeating sequence of packet sizes from the usual set of fixed sizes.

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 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 April 9, 2011.

Table of Contents

1. Introduction

This memo defines a method to unambiguously specify the sequence of packet sizes used in a load test.

Benchmarking Methodologies [RFC2544] have always relied on test conditions with constant packet sizes, with the goal of understanding what network device capability has been tested. Tests with the smallest size stress the header processing capacity, and tests with the largest size stress the overall bit processing capacity. Tests with sizes in-between may determine the transition between these two capacities.

Constant packets sizes differ significantly from the conditions encountered in operational deployment, and so additional tests are sometimes conducted with a mixture of packet sizes. The set of sizes used is often called an Internet Mix, or "IMIX" [Spirent], [IXIA], [Agilent].

The mixture of sizes a networking device will encounter is highly variable and depends on many factors. An IMIX suited for one networking device and deployment will not be appropriate for another. However, the mix of sizes may be known and the tester may be asked to augment the fixed size tests.

To address this need, and the additional goal of repeatable test conditions, this draft proposes a way to specify the exact repeating sequence of packet sizes from the usual set of fixed sizes: the IMIX Genome.

1.1. First Draft

In this first draft, some section are very short or to-be-provided (TBP), and there are several questions identified for further discussion.

2. Scope and Goals

This memo defines a method to unambiguously specify the sequence of packet sizes that have been used in a load test, assuming that a relevant mix of sizes is known to the tester and the length of the repeating sequence is not very long (<30 packets).

The IMIX Genome will allow an exact sequence of packet sizes to be communicated as a single-line name, resolving the current ambiguity with results that simply refer to "IMIX".

While documentation of the exact sequence is ideal, the memo also covers the case where the sequence of sizes is very long or may be generated by a pseudo-random process.

It is a colossal non-goal to standardize one or more versions of the IMIX. This topic has been discussed on many occasions on the bmwg-list[IMIXonList]. The goal is to enable customization with minimal constraints while fostering repeatable testing once the fixed size testing is complete.

3. Specification of the IMIX Genome

The IMIX Genome is specified in the following format:

IMIX - 123456...x

where each number is replaced by the letter corresponding to the packet size of the packet at that position in the sequence. The following table gives the letter encoding for the [RFC2544] standard sizes (64, 128, 256, 512, 1024, 1280, and 1518 bytes).

Size, bytes	Genome Code Letter
64	a
128	b
256	c
512	d
1024	e
1280	f
1518	g
MTU ??	h

For example: a five packet sequence with sizes 64,64,64,1280,1518 would be designated:

IMIX - aaafg

While this approach allows some flexibility, there are also constraints.

- Non-RFC2544 packet sizes would need to be approximated by those available in the table.
- The Genome for very long sequences can become undecipherable by humans.

- o Whether h=MTU is useful/desirable is TBD.

- o Whether more tabulated packet sizes would be useful is TBD.

Some open issues with this format are:

1. Multiple Source-Destination Address Pairs: is the IMIX sequence applicable to each pair, across multiple pairs in sets, or across all pairs?

2. Multiple Tester Ports:is the IMIX sequence applicable to each port, across multiple ports in sets, or across all ports?

4. Reporting Long or Pseudo-Random Packet Sequences

When the IMIX-Genome cannot be used (when the sheer length of the sequence would make the genome unmanageable) or when the sequence is designed to vary within some proportional constraints, a table is necessary.

IP Length	Percentage of Total	Other Length(s)
64	23	82
128	67	146
1000	10	1018

Note that this approach also allows non-standard packet sizes, but trades the short genome specification and ability to specify the exact sequence for other flexibilities.

>>> Specification for psuedo-random size generation here? <<<

5. Security Considerations

Benchmarking activities as described in this memo are limited to technology characterization using controlled stimuli in a laboratory environment, with dedicated address space and the other constraints [RFC2544].

The benchmarking network topology will be an independent test setup and MUST NOT be connected to devices that may forward the test traffic into a production network, or misroute traffic to the test management network.

Further, benchmarking is performed on a "black-box" basis, relying solely on measurements observable external to the DUT/SUT.

Special capabilities SHOULD NOT exist in the DUT/SUT specifically for benchmarking purposes. Any implications for network security arising from the DUT/SUT SHOULD be identical in the lab and in production networks.

6. IANA Considerations

This memo makes no requests of IANA, and hopes that IANA will leave it alone as well.

7. Acknowledgements

8. References

8.1. Normative References

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

[RFC2544] Bradner, S. and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices", RFC 2544, March 1999.

8.2. Informative References

[Agilent] http://www.ixiacom.com/pdfs/test_plans/agilent_journal_of_internet_test_methodologies.pdf, "The Journal of Internet Test Methodologies", 2007.

[IMIXonList] http://www.ietf.org/mail-archive/web/bmwg/current/msg00691.html, "Discussion on IMIX", 2003.

[IXIA] http://www.ixiacom.com/library/test_plans/display?skey=testing_pppox, "Library: Test Plans", 2010.

[Spirent] http://gospirent.com/whitepaper/IMIX%20Test%20Methodolgy%20Journal.pdf, "Test Methodology Journal: IMIX (Internet Mix) Journal", 2006.

Author's Address

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
Obsoletes: 4148 (if approved)
Updates: 4737, 5560, 5644, 6049
(if approved)
Intended status: Informational
Expires: July 14, 2011

A. Morton
AT&T Labs

January 10, 2011

RFC 4148 and the IPPM Metrics Registry are Obsolete
draft-morton-ippm-rfc4148-obsolete-03

Abstract

This memo reclassifies RFC 4148, the IP Performance Metrics (IPPM) Registry as Obsolete, and withdraws the IANA IPPM Metrics Registry itself from use because it is obsolete. The current registry structure has been found to be insufficiently detailed to uniquely identify IPPM metrics. Despite apparent efforts to find current or even future users, no one has responded to the call for interest in the RFC 4148 registry during the second half of 2010.

Table of Contents

1. Introduction

The IP Performance Metrics (IPPM) framework [RFC2330] describes several ways to record options and metric parameter settings, in order to account for sources of measurement variability. For example, Section 13 of[RFC2330] describes the notion of "Type P" so that metrics can be specified in general, but the specifics (such as payload length in octets and protocol type) can replace P to disambiguate the results.

When the IPPM Metric Registry [RFC4148] was designed, the variability of the Type P notion, and the variability possible with the many metric parameters (see Section 4.1 of [RFC2679]) was not fully appreciated. Further, some of the early metric definitions only indicate Poisson streams [RFC2330] (see the metrics in [RFC2679], [RFC2680], and [RFC3393]), but later work standardized the methods for Periodic Stream measurements [RFC3432], adding to the variability possible when characterizing a metric exactly.

It is not believed to be feasible or even useful to register every possible combination of Type P, metric parameters, and Stream parameters using the current structure of the IPPM Metric Registry.

The IPPM Metrics Registry is believed to have very few users, if any. Evidence of this provided by the fact that one registry entry was syntactically incorrect for months after [RFC5644] was published. The text ":=" was used for the metrics in that document instead of "::=". It took eight months before someone offered that a parser found the error. Even the original registry author agrees that the current registry is not efficient, and has submitted a proposal to effectively create a new registry.

Despite apparent efforts to find current or even future users, no one has responded to the second half of 2010 call for interest in the RFC 4148 registry. Therefore, the IETF now declares the registry Obsolete without any further reservations.

When a registry is designated Obsolete, it simply prevents IANA from registering new objects, in this case new metrics. So, even if a registry user was eventually found, they could continue to use the current registry and its contents will continue to be available.

The most recently published memo that added metrics to the registry is [RFC6049]. This memo updates all previous memos that registered new metrics, including [RFC4737] and [RFC5560], so that the registry’s Obsolete status will be evident.

2. Action to Reclassify RFC 4148 and the corresponding IANA registry as Obsolete

Due to the ambiguities between the current metrics registrations and the metrics used, and the apparent minimal adoption of the registry in practice, it is required that:

- the IETF reclassify [RFC4148] as Obsolete.
- the IANA withdraw the current IPPM Metrics Registry from further updates and note that it too is Obsolete.

It is assumed that parties who wish to establish a replacement registry function will work to specify such a registry.

3. Security Considerations

This memo and its recommendations have no known impact on the security of the Internet (especially if there is a zombie apocalypse on the day it is published; humans will have many more security issues to worry about stemming from the rise of the un-dead).

4. IANA Considerations

Metrics defined in IETF have been typically registered in the IANA IPPM METRICS REGISTRY as described in initial version of the registry [RFC4148]. However, areas for improvement of this registry have been identified, and the registry structure has to be revisited when there is working group consensus to do so.

The current consensus is to designate the IPPM Metrics Registry, originally described in [RFC4148], as Obsolete.

The DESCRIPTION of the registry MIB should be modified as follows, and the first two sentences should be included on any IANA-maintained web-page describing this registry or its contents (with the RFC number of this memo replacing "XXXX"):

DESCRIPTION

"With the approval and publication of RFC XXXX, this module is designated Obsolete.

The registry will no longer be updated, and the current contents will be maintained as-is on the day that RFC XXXX was published.

The original Description text follows below:

This module defines a registry for IP Performance Metrics.

... "

5. Acknowledgements

Henk Uijterwaal suggested additional rationale for the recommendation in this memo.

6. References

6.1. Normative References

[RFC4148] Stephan, E., "IP Performance Metrics (IPPM) Metrics Registry", BCP 108, RFC 4148, August 2005.

6.2. Informative References

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

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

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

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

[RFC3432] Raisanen, V., Grotefeld, G., and A. Morton, "Network performance measurement with periodic streams", RFC 3432, November 2002.

[RFC4737] Morton, A., Ciavattone, L., Ramachandran, G., Shalunov, S., and J. Perser, "Packet Reordering Metrics", RFC 4737, November 2006.

[RFC5560] Uijterwaal, H., "A One-Way Packet Duplication Metric", RFC 5560, May 2009.

[RFC5644] Stephan, E., Liang, L., and A. Morton, "IP Performance

Metrics (IPPM): Spatial and Multicast", RFC 5644, October 2009.

[RFC6049] Morton, A. and E. Stephan, "Spatial Composition of Metrics", RFC 6049, January 2011.

Author's Address

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: Standards Track
Expires: April 9, 2011

A. Morton
AT&T Labs

October 6, 2010

Round-trip Loss Metrics
draft-morton-ippm-rt-loss-01

Abstract

Many user applications and the transport protocols that make them possible require two-way communications. To address this need, and also for system simplicity, round-trip loss measurements are frequently conducted in practice. The Two-Way Active Measurement Protocol specified in RFC 5357 establishes a round-trip loss measurement capability for the Internet. However, there is currently no metric specified according to the RFC 2330 framework.

This memo proposes/adds round-trip loss to the set of IP Performance Metrics (IPPM).

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 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 April 9, 2011.

Copyright Notice

Table of Contents

1. Introduction

This memo defines a metric for round-trip loss on Internet paths. It builds on the notions and conventions introduced in the IP Performance Metrics (IPPM) framework [RFC2330]. Also, the specifications of the One-way Loss metric [RFC2680] and the Round-trip Delay metric [RFC2681] are frequently referenced and modified to match the round-trip circumstances addressed here. However, this memo assumes that the reader is familiar with the references, and does not repeat material as was done in [RFC2681].

This memo uses the terms "two-way" and "round-trip" synonymously.

1.1. Motivation

Many user applications and the transport protocols that make them possible require two-way communications. For example, the TCP SYN->, <-SYN-ACK, ACK-> three-way handshake attempted billions of times each day cannot be completed without two-way connectivity in a near-simultaneous time interval. Thus, measurements of Internet round-trip loss performance provide a basis to infer application performance more easily.

Measurement system designers have also recognized advantages of system simplicity when one host simply echoes or reflects test packets to the sender. Round-trip loss measurements are frequently conducted and reported in practice. The Two-Way Active Measurement Protocol specified in [RFC5357] establishes a round-trip loss measurement capability for the Internet. However, there is currently no round-trip loss metric specified according to the [RFC2330] framework.

[RFC2681] indicates that round-trip measurements may sometimes encounter "asymmetric" paths. When loss is observed using a round-trip measurement, there is often a desire to ascertain which of the two directional paths "lost" the packet. Under some circumstances, it is possible to make this inference. The round-trip measurement method raises a few complications when interpreting the embedded one-way results, and the user should be aware of them.

[RFC2681] also points out that loss measurement conducted sequentially in both directions of a path and reported as a round-trip result may be exactly the desired metric. On the other hand, it may be difficult to derive the state of round-trip loss from one-way measurements conducted in each direction unless a method to match the appropriate one-way measurements has pre-arranged.

Finally, many measurement systems report statistics on a conditional

delay distribution, where the condition is packet arrival at the destination. This condition is encouraged in [RFC3393], [RFC5481], and [draft-ietf-ippm-reporting-metrics]. As a result, lost packets need to be reported separately, according to a standardized metric. This memo defines such a metric.

See Section 1.1 of[RFC2680] for additional motivation of the packet loss metric.

2. Scope

This memo defines a round-trip loss metric using the conventions of the IPPM framework [RFC2330].

The memo defines a singleton metric, a sample metric, and a statistic, as per [RFC2330].

The memo also investigates the topic of one-way loss inference from a two-way measurement, and lists some key considerations.

3. Common Specifications for Round-trip Metrics

To reduce the redundant information presented in the detailed metrics sections that follow, this section presents the specifications that are common to two or more metrics. The section is organized using the same subsections as the individual metrics, to simplify comparisons.

3.1. Name: Type-P-*

All metrics use the Type-P convention as described in [RFC2330]. The rest of the name is unique to each metric.

3.2. Metric Parameters

- Src, the IP address of a host
- Dst, the IP address of a host
- T, a time (start of test interval)
- Tf, a time (end of test interval)
- lambda, a rate in reciprocal seconds (for Poisson Streams)

- incT, the nominal duration of inter-packet interval, first bit to first bit (for Periodic Streams)

- T0, a time that MUST be selected at random from the interval [T, T+dT] to start generating packets and taking measurements (for Periodic Streams)

- TstampSrc, the wire time of the packet as measured at MP(Src) as it leaves for Dst.

- TstampDst, the wire time of the packet as measured at MP(Dst), assigned to packets that arrive within a "reasonable" time.

- Tmax, a maximum waiting time for packets to arrive, set sufficiently long to disambiguate packets with long delays from packets that are discarded (lost).

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

- N, the total number of packets received at Dst (sent between T0 and Tf)

- Type-P, as defined in [RFC2330], which includes any field that may affect a packet's treatment as it traverses the network

3.3. Metric Definition

This section is specific to each metric.

3.4. Metric Units

The metric units are logical (1 or 0) when describing a single packet's loss performance, where a 0 indicates successful packet transmission and a 1 indicates packet loss.

Units of time are as specified in [RFC2330].

Other units used are defined in the associated section.

4. A Singleton Round-trip Loss Metric

4.1. Name: Type-P-Round-trip-Loss

4.2. Metric Parameters

See section 3.2.

4.3. Definition and Metric Units

Type-P-Round-trip-Loss SHALL be represented by the binary logical values (or their equivalents) when the following conditions are met:

Type-P-Round-trip-Loss = 0:

- Src sent the first bit of a Type-P packet to Dst at wire-time TstampSrc,
- that Dst received that packet,
- the Dst immediately sent a Type-P packet back to the Src, and
- that Src received the last bit of the reflected packet at wire-time TstampSrc + Tmax.

Type-P-Round-trip-Loss = 1:

- Src sent the first bit of a Type-P packet to Dst at wire-time TstampSrc,
- that Src did not receive the last bit of the reflected packet before the waiting time lapsed at TstampSrc + Tmax
- (possibly because that Dst did not receive that packet,
- the Dst did not immediately sent a Type-P packet back to the Src, or
- the Src did not receive a reflected Type-P packet sent from the Dst).

Following the precedent of[RFC2681], we make the simplifying assertion:

Type-P-Round-trip-Loss(Src->Dst) = Type-P-Round-trip-Loss(Dst->Src)

(and agree with the rationale, that the ambiguity introduced is a small price to pay for measurement efficiency).

Therefore, each singleton can be represented by pairs of elements as follows:

- o TstampSrc, the wire time of the packet at the Src (beginning the round-trip journey).

- o L, either zero or one (or some logical equivalent), where L=1 indicates loss and L=0 indicates successful round-trip arrival prior to TstampSrc + Tmax.

4.4. Discussion and other details

See [RFC2680] and [RFC2681] for extensive discussion, methods of measurement, errors and uncertainties, and other fundamental considerations that need not be repeated here.

5. A Sample Round-trip Loss Metric

Given the singleton metric Type-P-Round-trip-Loss, we now define one particular sample of such singletons. The idea of the sample is to select a particular binding of the parameters Src, Dst, and Type-P, then define a sample of values of parameter TstampSrc. This can be done in several ways, including:

1. Poisson: a pseudo-random Poisson process of rate lambda, whose values fall between T and Tf. The time interval between successive values of TstampSrc will then average 1/lambda, as per [RFC2330].

2. Periodic: a periodic stream process with pseudo-random start time T0 between T and dT, and nominal inter-packet interval incT, as per [RFC3432].

In the metric name, the variable <Stream> should be replaced with the process used to define the sample, using one of the above processes (or other process, the details of which MUST be specified if used).

5.1. Name: Type-P-Round-trip-Loss-<Sample>-Stream

5.2. Metric Parameters

See section 3.2.

5.3. Definition and Metric Units

Given one of the methods for defining the test interval, the sample of times (TstampSrc) and other metric parameters, we obtain a sequence of Type-P-Round-trip-Loss singletons as defined in section 4.3.

Type-P-Round-trip-Loss-<Sample>-Stream SHALL be a sequence of pairs with elements as follows:

- o TstampSrc, as above
- o L, either zero or one (or some logical equivalent), where L=1 indicates loss and L=0 indicates successful round-trip arrival prior to TstampSrc + Tmax.

where <Sample> SHALL be replaced with "Poisson", "Periodic", or an appropriate term to designate another sample method meeting the criteria of [RFC2330].

5.4. Discussion and other details

See [RFC2680] and [RFC2681] for extensive discussion, methods of measurement, errors and uncertainties, and other fundamental considerations that need not be repeated here.

6. Round-trip Loss Statistic

This section gives the primary and overall statistic for loss performance. Additional statistics and metrics originally prepared for One-way loss MAY also be applicable.

6.1. Type-P-Round-trip-Loss-<Sample>-Ratio

Given a Type-P-Round-trip-Loss-<Sample>-Stream, the average of all the logical values, L, in the Stream is the Type-P-Round-trip-Loss-<Sample>-Ratio. This ratio is in units of lost packets per round-trip transmissions attempted.

In addition, the Type-P-Round-trip-Loss-<Sample>-Ratio is undefined if the sample is empty.

7. Round-trip Testing and One-way Reporting

This section raises considerations for results collected using a round-trip measurement architecture, such as in TWAMP [RFC5357].

The sampling process for the return path (Dst->Src) is a conditional process that depends on successful packet arrival at the Dst and correct operation at the Dst to generate the reflected packet. Therefore, the sampling process for the return path will be significantly affected when appreciable loss occurs on the Src->Dst path, making an attempt to assess the return path performance invalid

(for loss or possibly any metric).

Further, the sampling times for the return path (Dst->Src) are a random process that depends on the original sample times (TstampSrc), the one-way-delay for successful packet arrival at the Dst, and time taken at the Dst to generate the reflected packet. Therefore, the sampling process for the return path will be significantly affected when appreciable delay variation occurs on the Src->Dst path, making an attempt to assess the return path performance invalid (for loss or possibly any metric).

8. Security Considerations

8.1. Denial of Service Attacks

This metric requires a stream of packets sent from one host (source) to another host (destination) through intervening networks, and back. This method could be abused for denial of service attacks directed at the destination and/or the intervening network(s).

Administrators of source, destination, and the intervening network(s) should establish bilateral or multi-lateral agreements regarding the timing, size, and frequency of collection of sample metrics. Use of this method in excess of the terms agreed between the participants may be cause for immediate rejection or discard of packets or other escalation procedures defined between the affected parties.

8.2. User Data Confidentiality

Active use of this method generates packets for a sample, rather than taking samples based on user data, and does not threaten user data confidentiality. Passive measurement must restrict attention to the headers of interest. Since user payloads may be temporarily stored for length analysis, suitable precautions MUST be taken to keep this information safe and confidential. In most cases, a hashing function will produce a value suitable for payload comparisons.

8.3. Interference with the metrics

It may be possible to identify that a certain packet or stream of packets is part of a sample. With that knowledge at the destination and/or the intervening networks, it is possible to change the processing of the packets (e.g. increasing or decreasing delay) that may distort the measured performance. It may also be possible to generate additional packets that appear to be part of the sample metric. These additional packets are likely to perturb the results of the sample measurement.

To discourage the kind of interference mentioned above, packet interference checks, such as cryptographic hash, may be used.

9. IANA Considerations

Metrics defined in IETF are typically registered in the IANA IPPM METRICS REGISTRY as described in initial version of the registry [RFC4148]. However, areas for improvement of this registry have been identified, and the registry structure has to be revisited when there is consensus to do so.

Therefore, the metrics in this draft may be considered for registration in the future, and no IANA Action is requested at this time.

10. Acknowledgements

11. References

11.1. Normative References

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

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

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

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

[RFC2681] Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, September 1999.

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

[RFC3432] Raisanen, V., Grotefeld, G., and A. Morton, "Network performance measurement with periodic streams", RFC 3432, November 2002.

[RFC4148] Stephan, E., "IP Performance Metrics (IPPM) Metrics Registry", BCP 108, RFC 4148, August 2005.

[RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.

[RFC5835] Morton, A. and S. Van den Berghe, "Framework for Metric Composition", RFC 5835, April 2010.

11.2. Informative References

[RFC5474] Duffield, N., Chiou, D., Claise, B., Greenberg, A., Grossglauser, M., and J. Rexford, "A Framework for Packet Selection and Reporting", RFC 5474, March 2009.

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

[Stats] McGraw-Hill NY NY, "Introduction to the Theory of Statistics, 3rd Edition,", 1974.

Author's Address

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: Standards Track
Expires: January 2, 2011

E. Stephan
France Telecom Orange

july 1, 2010

IPPM Metrics Registry Extension
draft-stephan-ippm-registry-ext-00

Abstract

The current IANA IPPM Metrics Registry [RFC4148] only assigns an identifier to each IP Performance Metrics (IPPM) defined in the IETF. This document extends this registry for enabling the registration of fine-grained information on each metric.

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 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 2, 2011.

Copyright Notice

Table of Contents

1. Introduction

The current IANA IPPM Metrics Registry [RFC4148] assigns an identifier to each IP Performance Metrics (IPPM) defined in the IETF. This document extends this registry for enabling the registration of fine-grained information on each metric.

2. Overview

To facilitate the understanding of the changes this document reuse mostly the structure of [RFC4148].

The current version assumes that IESG will request backward compatibility with the existing registry.

This memo suggest to extend the current registry for the following reasons:

- The current registry is designed as a MIB extension which may be used by other MIB modules to identify specific IP Performance Metrics. This precludes the usage of the registry by other management frameworks like those based on XML. The new registry should be easely parsable by other management frameworks.

- parameters: It should capture information to distinguish flavors of a metric when a metric have optional parameters.

- results: It should register parameters for easing the comparison of metrics. As a example an ouput parameter should be registered with clear units (time, number of packet, bytes...) or default value (e.g. milliseconds, kbytes...);

3. IPPM Registry Extension Framework

The new registry should preserve the compatibilty with the existing one because MIB compilers already import this as a MIB module. Nevertheless the extension part does not inherit of this constraint. In brief the new registry is made of 2 legs the existing one and a new one which should be readable by non SMI network management frameworks.

3.1. Leg1, existing registry

Leg1 corresponds the the current SMIv2 module. Its behavior is unchanged. New metrics are still identified in 'ianaIppmMetrics' subtree.

Furthermore the number assigned to a metric is copied in the table of the metrics of the Leg2.

3.2. Leg2, metrics parameters and options

To capture the characterization of each metric the Leg2 has the following structure :

- One table of metric names and identifiers given by the Leg1;
- A list of metrics flavors

The table of metric names copies the metric names, id and reference from the 'ianaIppmMetrics' subtree of the Leg1 (pratically this is done by IANA):

MetricName	MetricId
ietfInstantUnidirConnectivity	1
ietfInstantBidirConnectivity	2
...	...
ietfOneToGroupRangeDelayVariation	70

Metrics Table

Then metrics flavors are defined separatly after this table.

Each metric flavor is introduced with its name and fields like the MetricName it is based on and a brief description. Then the parameters of the metric flavor are listed in a dedicaced table described below.

Name	Unit	Cardinality	Description	Type
the name of the metric	The default unit	The parameter is mandatory or optional	Text precising the meaning of the parameter	Input or output

Metric flavor table

4. Discussion and Open issues

Complexity: The new registry will probably have 2 legs, a SMI leg and the extension leg. Is this too complex ?

Duplication of works: Having 2 legs means duplicating the metric identifier to provide natural access to SMI and non SMI frameworks. It is the price to have the metric identifiers to be shared amongs SMI and non SMI management frameworks.

Security considerations: Diff with v1 of the registry: Security considerations differ from the initial registry because the new registry exposes detailed information on the metrics.

Do we keep the retro compatibily with the initial registry ? IESG will probably say 'Yes', I made this asumption and may be wrong.

Initial content: Do we initiate the extension of the registry with content ?

Reporting metrics: This document does not specify a management interface. Nevertheless it may be somewhat tied with the work on the reporting of metrics the IPPM WG is currently addressing. How to benefit from that ?

5. IANA Considerations

This section describes the rules for the management of the registry by IANA.

The management of the ianaIppmMetrics subtree (existing registry) is inchanged. The rules below include these rules . Several are common to the 2 legs.

5.1. New Registry Management rules

Registrations are done sequentially by IANA on the bases of 'Specification Required' as defined in [RFC2434]. The number and the name identifying a metric is the same in the 2 legs.

The reference of the specification must point to a stable document including a title, a revision and a date.

The name always starts with the name of the organization and must respect the SMIv2 rules for descriptors defined in the section 3.1 of [RFC2578];

A document that creates new metrics would have an IANA considerations section in which it would describe new metrics to register.

Additional documents may add new metric flavors in the registry on the bases of 'Specification Required' as defined in [RFC2434].

5.1.1. ianaIppmMetrics subtree (SMI leg)

Registrations are done sequentially by IANA in the ianaIppmMetrics subtree. The number and the name identifying the metric is reused i the leg2.

An OBJECT IDENTITY assigned to a metric is definitive and cannot be reused. If a new version of a metric is produced then it is assigned with a new name and a new identifier.

5.1.2. Leg2

see section 3.2

6. Security Considerations

FIXME: Security considerations differ from the initial registry.

7. Acknowledgements

8. References

8.1. Normative References

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

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC4148] Stephan, E., "IP Performance Metrics (IPPM) Metrics Registry", BCP 108, RFC 4148, August 2005.

8.2. Informative References

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

Appendix A. An Appendix

Author's Address

Stephan Emile
France Telecom Orange
2 avenue Pierre Marzin
Lannion F-22307
France

Email: emile.stephan@orange-ftgroup.com