

IPPM
Internet-Draft
Intended status: Standards Track
Expires: April 18, 2013

Emily. Bi
K. Pentikousis
Yang. Cui
Huawei Technologies
October 15, 2012

Network Performance Measurement for IPsec
draft-bi-ippm-ipsec-00.txt

Abstract

IPsec is a mature technology with several interoperable implementations. Indeed, the use of IPsec tunnels is increasingly gaining popularity in several deployment scenarios, not the least in what used to be solely areas of traditional telecommunication protocols. Wider deployment calls for mechanisms and methods that enable tunnel end-users, as well as operators, to measure one-way and two-way network performance. Unfortunately, however, standard IP performance measurement mechanisms cannot be readily used with IPsec. This document makes the case for employing IPsec to protect O/TWAMP and proposes a method which combines IKEv2 and O/TWAMP as defined in RFC 4656 and RFC 5357, respectively. This specification aims, on the one hand, to ensure that O/TWAMP can be secured well, while on the other hand, it extends the applicability of O/TWAMP to networks that have already deployed IPsec.

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 18, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

Table of Contents

1. Introduction	4
2. Terminology used in this document	4
3. Motivation	4
3.1 O/TWAMP-Control Security	5
3.2 O/TWAMP-Test Security	6
3.3 O/TWAMP Security Root	7
3.4 Why IPPM cannot be employed over IPsec?	7
4. O/TWAMP over IPsec	8
5. Others	12
6. Security Considerations	12
7. IANA Considerations	12
8. Acknowledgments	12
Authors' Addresses	12

1. Introduction

The active measurement protocols OWAMP [RFC4656] and TWAMP [RFC5357] can be used to measure network performance parameters, such as latency, bandwidth, and packet loss by sending probe packets and monitoring their experience in the network. In order to guarantee the accuracy of network measurement results, security aspects must be considered, otherwise, attacks may occur and authenticity may be violated. For example, a man in the middle may modify packet timestamps if no protection is provided.

Cryptographic security mechanisms, such as IPsec, have been considered during the early stage of working towards the definition of the two protocols mentioned above. However, due to several reasons, it was preferred to avoid tying the development and deployment of O/TWAMP protocols to such security mechanisms. In practice, for many networks, the issues listed in [RFC4656], Sec. 6.6 with respect to IPsec are still valid. However, we expect that in the near future IPsec will be deployed in many more hosts and networks than today. Therefore, in this document we attempt to make the case that for networks where wide deployment of IPsec and other security mechanisms is mandatory for a variety of reasons, there are increasingly more use cases in which IPsec and O/TWAMP protocols are needed simultaneously. In other words, we argue that it is now time to specify how O/TWAMP can be used in a network environment where IPsec is already deployed. In such an environment, measuring IP performance over IPsec tunnels with O/TWAMP is an important tool for operators.

2. Terminology used in this document

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

3. Motivation

Let us first consider why the reasons listed [RFC4656] Sec. 6.6 may not apply in many cases. First, the argument made is that partial authentication in OWAMP authentication mode is not possible with IPsec. IPsec indeed cannot authenticate only a part of a packet. However, in an environment where IPsec is already deployed and actively used, partial authentication of O/TWAMP contradicts the operational reasons dictating the use of IPsec. At the same time, this limits the applicability and use of O/TWAMP in networks using IPsec.

The second argument made is the need to keep separate deployment paths between OWAMP and IPsec. In several currently deployed types of networks, IPsec is widely used to protect the data and signaling planes. For example, in mobile telecommunication networks, the deployment rate of IPsec exceeds 95% with respect to the LTE serving network. In older technology cellular networks, such as UMTS and GSM, this percentage is lower, but still quite significant. Additionally, there is a great number of IPsec-based VPN applications which are widely used in business applications to provide end-to-end, or host-to-host security. At the same time, lots of standardized protocols make use of IPsec/IKE, including MIPv4/v6, HIP, SCTP, BGP, NAT and SIP, just to name a few.

Third, with respect to the support of IPsec in lightweight embedded devices, nowadays, a large number of limited-resource and low-cost devices, such as Ethernet switches, DSL modems, and other such devices come with support for IPsec "out of the box". Therefore concerns about implementation, although likely valid a decade ago, are not well founded today.

Fourth, everyday use of IPsec applications by field technicians, on the one hand, and by good understanding of the IPsec API by many programmers, on the other, should not be anymore a reason for concern. On the contrary: By now, IPsec open source code is available for anyone who wants to use it. Therefore, although IPsec does need a certain level of expertise to deal with it, in practice, most competent technical personnel and programmers have no problems using it on a daily basis.

O/TWAMP actually consists of two inter-related protocols: O/TWAMP-Control and O/TWAMP-Test. O/TWAMP-Control is used to initiate, start, and stop test sessions and to fetch their results, whereas O/TWAMP-Test is used to exchange test packets between two measurement nodes. In the following subsections we consider security for each one separately and then make the case for using them over IPsec.

3.1 O/TWAMP-Control Security

O/TWAMP uses a simple cryptographic protocol which relies on AES-CBC for confidentiality and on HMAC-SHA1 truncated to 128 bits for message authentication. Three modes of operation are supported: unauthenticated, authenticated, and encrypted. The authenticated and encrypted modes require that endpoints possess a shared secret, typically a passphrase. The secret key is derived from the passphrase using a password-based key derivation function PBKDF2 (PKCS#5) [RFC2898].

In the unauthenticated mode, the security parameters are unused. In authenticated and encrypted mode, the security parameters will be negotiated during the control connection establishment. Before the client can send commands to a server, it has to establish a connection to the server. Then, the client opens a TCP connection to the server on the well-known port number 861. The server responds with a server greeting, which contains the Challenge, mode, Salt and Count. If the client wants to establish the connection, it responds with a Set-Up-Response message, wherein the KeyID, Token and Client IV are included. The Token is the concatenation of a 16-octet challenge, a 16-octet AES Session-key used for encryption, and a 32-octet HMAC-SHA1 Session-key used for authentication. The token itself is encrypted using AES in Cipher Block Chaining (AES-CBC).

Encryption is performed using a key derived from the shared secret associated with KeyID. In authenticated and encrypted modes, all further communications are encrypted using the AES Session-key and authenticated with HMAC Session-key. The client encrypts everything it sends through the just-established O/TWAMP-Control connection using stream encryption with Client-IV as the IV. Correspondingly, the server encrypts its side of the connection using Server-IV as the IV. The IVs themselves are transmitted in cleartext. Encryption starts with the block immediately following the block containing the IV.

The AES Session-key and HMAC Session-key are generated randomly by the client. The HMAC Session-key is communicated along with the AES Session-key during O/TWAMP-Control connection setup. The HMAC Session-key is derived independently of the AES Session-key.

3.2 O/TWAMP-Test Security

The O/TWAMP-Test protocol runs over UDP, using sender and receiver IP and port numbers negotiated during the Request-Session exchange. As with O/TWAMP-Control, O/TWAMP-Test has three modes: unauthenticated, authenticated, and encrypted. All O/TWAMP-Test sessions that are spawned by an O/TWAMP-Control session inherit its mode.

The O/TWAMP-Test packet layout is the same in authenticated and encrypted modes. The encryption and authentication operations are, however, different. Similarly with the respective O/TWAMP-Control session, each O/TWAMP-Test session has two keys: an AES Session-key and an HMAC Session-key. However, there is a difference in how the keys are obtained. In the case of O/TWAMP-Control, the keys are generated by the client and communicated (as part of the Token) during connection setup as part of Set-Up-Response message. In the case of O/TWAMP-Test, the keys are derived from the O/TWAMP-Control

keys and the session identifier (SID). The O/TWAMP-Test AES Session-key is obtained from the O/TWAMP-Control AES Session-key. The O/TWAMP-Control AES Session-key is encrypted using AES, with the 16-octet session identifier (SID) as the key. The result is the O/TWAMP-Test AES Session-key used to encrypt and decrypt the packets of the particular O/TWAMP-Test session. The O/TWAMP-Test HMAC Session-key is obtained from the O/TWAMP-Control HMAC Session-key. The O/TWAMP-Control HMAC Session-key is encrypted using AES, with the 16-octet session identifier (SID) as the key. The result is the O/TWAMP-Test HMAC Session-key used in authenticating the packets of the particular O/TWAMP-Test session.

3.3 O/TWAMP Security Root

As discussed above, the AES Session-key and HMAC Session-key used in the O/TWAMP-Test protocol are derived from the AES Session-key and HMAC Session-key which are used in O/TWAMP-Control protocol. The AES Session-key and HMAC Session-key used in the O/TWAMP-Control protocol are generated randomly by the client, and encrypted with the shared secret associated with KeyID. Therefore, the security root is the shared secret key.

3.4 Why IPPM cannot be employed over IPsec?

IPsec provides confidentiality and data integrity to IP datagrams. Three protocols are provided: Authentication Header (AH), Encapsulating Security Payload (ESP) and Internet Key Exchange (IKE v1/v2). Only integrity protection can be provided with AH. Both integrity and encryption can be provided with ESP. The IKE Protocol is used for dynamical key negotiation and automatic key management.

When the sender and receiver implement O/TWAMP over IPsec, the sender and the receiver will agree on a shared key during the establishment of IPsec, and all IP packets sent by the sender are protected with IPsec. If the AH protocol is used, IP packets are transmitted in plaintext. The authentication part covers the entire packet. So all test information, such as UDP port number, and the test results will be visible to any attacker, which can intercept these test packets, and introduce errors or forge packets that may be injected during the transmission. In order to avoid this attack, the receiver must validate the integrity of these packets with the negotiated secret key. If ESP is used, IP packets are encrypted, and hence no other than the receiver can use the IPsec secret key and decrypt the IP packet, and then it can obtain the test data to assess the IP network performance based on the measurements. So both the sender and receiver must support IPsec to generate the security secret key of

IPsec.

In the current implementation of O/TWAMP, after the test packets are received by the receiver, it cannot execute active measurement over IPsec. That is because the receiver knows only the shared secret key but not the IPsec key, while the test packets are protected with IPsec key ultimately. Therefore, it needs to be considered how to measure IP network performance over an IPsec tunnel with O/TWAMP. Without this functionality, the use of OWAMP and TWAMP over IPsec is hindered.

Of course, backward compatibility should be considered, as well. That is, the intrinsic security method based on shared key as specified in the O/TWAMP standards can also fit the other platforms. There should be no impact on the current security mechanisms defined in O/TWAMP for other use cases. This document describes a possible solution to this problem which takes advantage of the secret key derived by IPsec, to provision the key needed in RFC 4656 and RFC 5357.

4. O/TWAMP over IPsec

A security method based on a shared secret key has been defined in O/TWAMP [RFC 4656][RFC 5357]. In this section, in order to employ O/TWAMP over IPsec, a method of binding IPPM and IKEv2 is described, for those both the sender and receiver supporting the IPsec protocols. The shared key used in the security of O/TWAMP is derived from IPsec [RFC5996]. If the AH protocol is used, the IP packets are transmitted in plaintext. All of O/TWAMP is integrity-protected by IPsec. Even if the peers choose unauthenticated mode, IPsec integrity protection is provided to O/TWAMP. In authenticated and encrypted modes, the shared secret can be derived from IKE SA or IPsec SA. If the shared secret key is derived from IKE SA, SKEYSEED must be generated firstly. SKEYSEED and its derivatives are computed as the way in [RFC5996], where prf is a pseudorandom function:

$$\text{SKEYSEED} = \text{prf}(\text{Ni} \parallel \text{Nr}, g^{\text{ir}})$$

Ni and Nr are the nonces, negotiated during initial exchange. g^{ir} is the shared secret from the ephemeral Diffie-Hellman exchange and is represented as a string of octets. SKEYSEED can be used as the share secret key directly then the share key is equal to SKEYSEED. Alternatively, the shared secret key can be generated as follows: Shared secret key=PRF{ SKEYSEED, Session ID}, wherein the session ID is the SID agreed during the O/TWAMP-Test protocol.

If the shared secret key is derived from IPsec SA, the shared secret

key can be equal to KEYMAT, wherein $\text{KEYMAT} = \text{prf}+(\text{SK_d}, \text{Ni} \parallel \text{Nr})$ and the term "prf+" describes a function that outputs a pseudorandom stream based on the inputs to a prf [RFC5996]; or the shared secret key can be generated as follows: Shared secret key=PRF{ KEYMAT, Session ID} , wherein the session ID is the SID agreed during the O/TWAMP-Test protocol.

There are some cases for rekeying IKE SA and IPsec SA, after which the corresponding key of SA is updated. Generally ESP and AH SAs always exist in pairs, with one SA in each direction. If the SA is deleted, the key generated from IKE SA or IPsec SA should also be updated.

As discussed above, a binding association between the key generated from IPsec and the shared secret key needs to be considered. SA can be identified by SPI and protocol uniquely for a sender and a receiver. So these parameters should be agreed during the O/TWAMP protocol. When the sender and receiver execute O/TWAMP protocol to negotiate integrity key, the IPsec protocol and SPI should be checked. Only if two parameters are matched with the information of IPsec, should the O/TWAMP connection be established. As illustrated in Figure 1, the SPI and protocol type are included in the server greeting of the O/TWAMP-Control protocol. After the client receives the greeting, it closes the connection if it receives a greeting with an erroneous SPI and protocol value. Otherwise, the client responds with the following Set-Up-Response message and generate shared secret key. The message exchange flow is described as Figure 1:

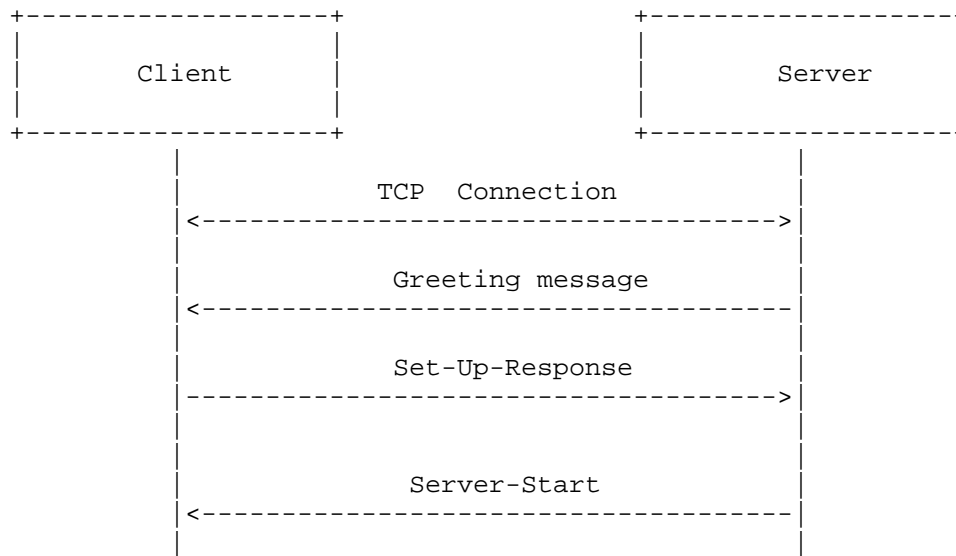


Figure 1. The procedure of O/TWAMP-Control

The format of server greeting is illustrated in Figure 2.

The unused 12 octets are used to carry the new parameter: protocol and SPIs.

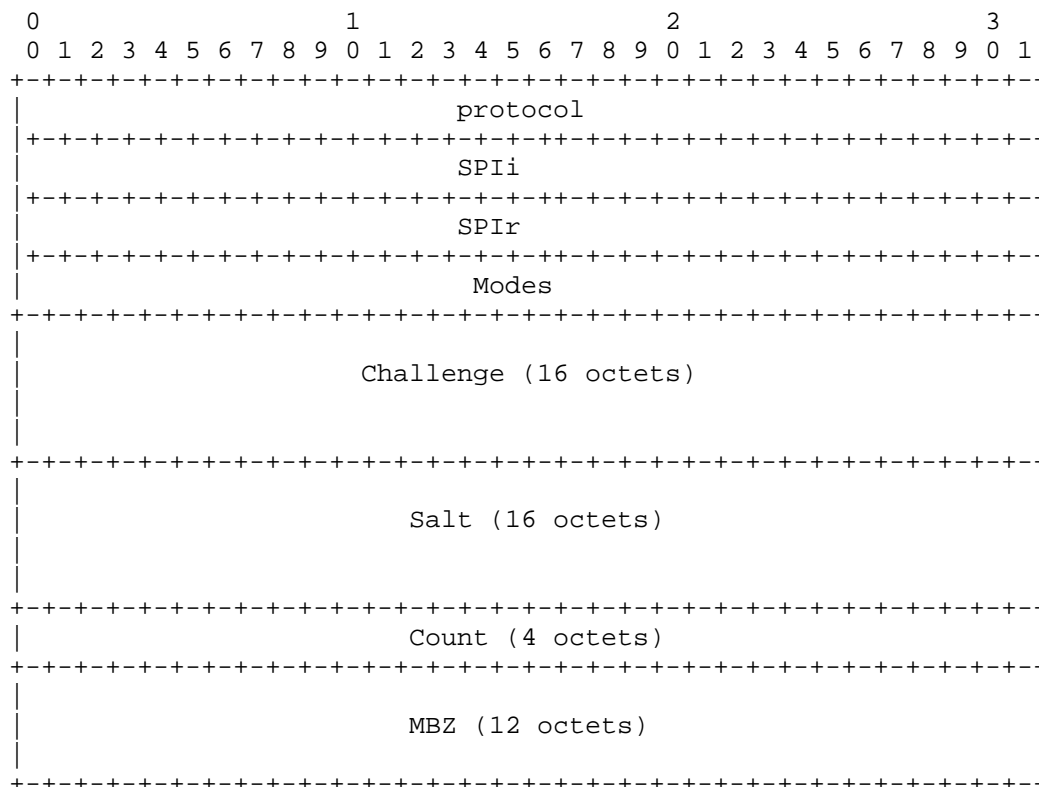


Figure 2. The format of server greeting

In ESP, when the IP packets are encrypted, no other than the receiver achieves the IPsec key and decrypted the IP packet. It gains the test data to process measurement IP performance. IPsec protocol between the sender and receiver provides additional security. Even if the peers choose unauthenticated mode, IPsec encryption and integrity protection is provided to O/TWAMP. If the sender and receiver also want to use authenticated or encrypted mode, the shared secret can be also derived from IKE SA or IPsec SA. The method of key generation and binding association is the same as AH protocol mode.

Besides, there is encryption-only configuration in ESP, though not recommended for its insecurity. Since it does not produce integrity key in this case, either encryption-only ESP should be prohibited for O/TWAMP, or a decryption failure should be distinguished due to possible integrity attack.

5. Others

To be added

6. Security Considerations

As the shared secret key is derived from IPsec, the key derived algorithm strength and limitations are as per [RFC5996]. The strength of a key derived from a Diffie-Hellman exchange using any of the groups defined here depends on the inherent strength of the group, the size of the exponent used, and the entropy provided by the random number generator used. The strength of all keys and implementation vulnerabilities, particularly DoS attacks are as defined in [RFC5996].

7. IANA Considerations

There may be IANA consideration for taking additional value for these options. The values of the protocol field needed to be assigned from the numbering space.

8. Acknowledgments

who contributed actively to this document.

Authors' Addresses

Emily Bi
Huawei Technologies
Huawei Building, Xinxu Road No.3
Haidian District, Beijing 100085
P. R. China

Phone: +86-10-60611962
Email: bixiaoyu@huawei.com

Kostas Pentikousis
Huawei Technologies
Carnotstr. 4
10587 Berlin
Germany

Email: k.pentikousis@huawei.com

Yang Cui
Huawei Technologies
Huawei Building, Xinxu Road No.3
Haidian District, Beijing 100085
P. R. China

Phone: +86-10-60611962
Email: cuiyang@huawei.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 26, 2012

A. Morton
AT&T Labs
June 24, 2012

Rate Measurement Problem Statement
draft-ietf-ippm-rate-problem-00

Abstract

There is a rate measurement scenario which has wide-spread attention of users and seemingly all industry participants, including regulators. This memo presents an access rate-measurement problem statement for IP Performance Metrics. Key aspects require the ability to control packet size on the tested path and enable asymmetrical packet size testing in a controller-responder architecture.

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 December 26, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal

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

Table of Contents

1. Introduction	3
2. Purpose and Scope	3
3. Active Rate Measurement	4
4. Measurement Method Categories	6
5. Test Protocol Control & Generation Requirements	7
6. Security Considerations	7
7. IANA Considerations	8
8. Acknowledgements	8
9. Appendix	8
10. References	8
10.1. Normative References	8
10.2. Informative References	9
Author's Address	9

1. Introduction

There are many possible rate measurement scenarios. This memo describes one rate measurement problem and presents a rate-measurement problem statement for IP Performance Metrics (IPPM).

The access-rate scenario or use case has wide-spread attention of users and seemingly all industry participants, including regulators. It is being approached with many different measurement methods.

2. Purpose and Scope

The scope and purpose of this memo is to define the measurement problem statement for access rate measurement on production networks. We characterize this scenario as follows:

- o The Access portion of the network is the focus of this effort. The user typically subscribes to a service with bi-directional access partly described by rates in bits per second.
- o Rates at the edge of the network are several orders of magnitude less than aggregation and core portions.
- o Asymmetrical ingress and egress rates are prevalent.
- o Extremely large scale of access services requires low complexity devices participating at the user end of the path.

Today, the majority of widely deployed access services achieve rates less than 100 Mbit/s, and this is the rate-regime for which a solution is sought now.

This problem statement assumes that the most-likely bottleneck device or link is adjacent to the remote (user-end) measurement device, or is within one or two router/switch hops of the remote measurement device.

Other use cases for rate measurement involve situations where the packet switching and transport facilities are leased by one operator from another and the actual capacity available cannot be directly determined (e.g., from device interface utilization). These scenarios could include mobile backhaul, Ethernet Service access networks, and/or extensions of a layer 2 or layer 3 networks. The results of rate measurements in such cases could be employed to select alternate routing, investigate whether capacity meets some previous agreement, and/or adapting the rate of certain traffic sources if a capacity bottleneck is found via the rate measurement.

In the case of aggregated leased networks, available capacity may also be asymmetric. In these cases, the tester is assumed to have a sender and receiver location under their control. We refer to this scenario below as the aggregated leased network case.

Only active measurement methods will be addressed here, consistent with the IPPM working group's current charter. Active measurements require synthetic traffic dedicated to testing, and do not use user traffic.

The actual path used may influence the rate measurement results for some forms of access, as it may differ between user and test traffic.

- o This issue requires further study to list the likely causes for this behavior. The possibilities include IP address assignment, transport protocol used (where TCP packets may be routed differently from UDP).

Although the user may have multiple instances of network access available to them, the primary intent is to measure one form of access at a time. It is plausible that a solution for the single access problem will be applicable to simultaneous measurement of multiple access instances, but this is beyond the current scope.

A key consideration is whether active measurements will be conducted with user traffic present (In-Service Testing), or not present (Out-of-Service Testing), such as during pre-service testing or maintenance that interrupts service temporarily. Out-of-Service testing includes activities described as "service commissioning", "service activation", and "planned maintenance". Both In-Service and Out-of-Service Testing are within the scope of this problem.

It is a non-goal to solve the measurement protocol specification problem in this memo.

It is a non-goal to standardize methods of measurement in this memo. However, the problem statement will mandate that support for one or more categories of rate measurement methods and adequate control features for the methods in the test protocol.

3. Active Rate Measurement

This section lists features of active measurement methods needed to measure access rates in production networks.

Test coordination between Source and Destination devices through control messages and other basic capabilities described in the

methods of IPPM RFCs [RFC2679][RFC2680] are taken as given (these could be listed later, if desired).

Most forms of active testing intrude on user performance to some degree. One key tenet of IPPM methods is to minimize test traffic effects on user traffic in the production network. Section 5 of [RFC2680] lists the problems with high measurement traffic rates, and the most relevant for rate measurement is the tendency for measurement traffic to skew the results, followed by the possibility to introduce congestion on the access link. Obviously, categories of rate measurement methods that use less active test traffic than others with similar accuracy SHALL be preferred for In-Service Testing.

On the other hand, Out-of-Service Tests where the test path shares no links with In-Service user traffic have none of the congestion or skew concerns, but must address other practical concerns such as conducting measurements within a reasonable time from the tester's point of view. Out-of-Service Tests where some part of the test path is shared with In-Service traffic MUST respect the In-Service constraints.

The ****intended metrics to be measured**** have strong influence over the categories of measurement methods required. For example, using the terminology of [RFC5136], a it may be possible to measure a Path Capacity Metric while In-Service if the level of background (user) traffic can be assessed and included in the reported result.

The measurement ***architecture*** MAY be either of one-way (e.g., [RFC4656]) or two-way (e.g., [RFC5357]), but the scale and complexity aspects of end-user or aggregated access measurement clearly favor two-way (with low-complexity user-end device and round-trip results collection, as found in [RFC5357]). However, the asymmetric rates of many access services mean that the measurement system MUST be able to assess each direction of transmission. In the two-way architecture, it is expected that both end devices MUST include the ability to launch test streams and collect the results of measurements in both (one-way) directions of transmission (this requirement is consistent with previous protocol specifications, it is not a unique problem for rate measurements).

The following paragraphs describe features for the roles of test packet SENDER, RECEIVER, and results REPORTER.

SENDER:

Ability to generate streams of test packets with various characteristics as desired (see Section 4). The SENDER may be

located at the user end of the access path, or may be located elsewhere in the production network, such as at one end of an aggregated leased network segment.

RECEIVER:

Ability to collect streams of test packets with various characteristics (as described above), and make the measurements necessary to support rate measurement at the other end of an end-user access or aggregated leased network segment.

REPORTER:

Ability to use information from test packets and local processes to measure delivered packet rates.

4. Measurement Method Categories

The design of rate measurement methods can be divided into two phases: test stream design and measurement (SENDER and RECEIVER), and a follow-up phase for analysis of the measurement to produce results (REPORTER). The measurement protocol that addresses this problem MUST only serve the test stream generation and measurement functions.

For the purposes of this problem statement, we categorize the many possibilities for rate measurement stream generation as follows:

1. Packet pairs, with fixed intra-pair packet spacing and fixed or random time intervals between pairs in a test stream.
2. Multiple streams of packet pairs, with a range intra-pair spacing and inter-pair intervals.
3. One or more packet ensembles in a test stream, using a fixed ensemble size in packets and one or more fixed intra-ensemble packet spacings (including zero).
4. One or more packet chirps, where intra-packet spacing typically decreases between adjacent packets in the same chirp and each pair of packets represents a rate for testing purposes.

For all categories, the test protocol MUST support:

1. Variable payload lengths among packet streams
2. Variable length (in packets) among packet streams or ensembles

3. Variable header markings among packet streams
4. Variable number of packets-pairs, ensembles, or streams used in a test session

are additional variables that the test protocol MUST be able to communicate.

The test protocol SHALL support test packet ensemble generation (category 3), as this appears to minimize the demands on measurement accuracy. Other stream generation categories are OPTIONAL.

Measurements for each test packet transferred between SENDER and RECEIVER MUST be compliant with the singleton measurement methods described in IPPM RFCs [RFC2679][RFC2680] (these could be listed later, if desired). The time-stamp information or loss/arrival status for each packet MUST be available for communication to the protocol entity that collects results.

5. Test Protocol Control & Generation Requirements

Essentially, the test protocol MUST support the measurement features described in the sections above. This requires:

1. Communicating all test variables to the Sender and Receiver
2. Results collection in a one-way architecture
3. Remote device control for both one-way and two-way architectures
4. Asymmetric and/or pseudo-one-way test capability in a two-way measurement architecture

The ability to control packet size on the tested path and enable asymmetrical packet size testing in a two-way architecture are REQUIRED.

The test protocol SHOULD enable measurement of the [RFC5136] Capacity metric, either Out-of-Service, In-Service, or both. Other [RFC5136] metrics are OPTIONAL.

6. Security Considerations

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

There may be a serious issue if a proprietary Service Level Agreement involved with the access network segment provider were somehow leaked in the process of rate measurement. To address this, test protocols SHOULD NOT convey this information in a way that could be discovered by unauthorized parties.

7. IANA Considerations

This memo makes no requests of IANA.

8. Acknowledgements

Dave McDysan provided comments and text for the aggregated leased use case. Yaakov Stein suggested many considerations to address, including the in-service vs. out-of-service distinction and its implication on test traffic limits.

9. Appendix

This Appendix is intended to briefly summarize previous rate measurement experience. (There is a large body of research on rate measurement, so there is a question of what to include and what to omit.)

10. References

10.1. Normative References

- [RFC1305] Mills, D., "Network Time Protocol (Version 3) Specification, Implementation", RFC 1305, March 1992.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [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.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.

- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.
- [RFC5618] Morton, A. and K. Hedayat, "Mixed Security Mode for the Two-Way Active Measurement Protocol (TWAMP)", RFC 5618, August 2009.
- [RFC5938] Morton, A. and M. Chiba, "Individual Session Control Feature for the Two-Way Active Measurement Protocol (TWAMP)", RFC 5938, August 2010.
- [RFC6038] Morton, A. and L. Ciavattone, "Two-Way Active Measurement Protocol (TWAMP) Reflect Octets and Symmetrical Size Features", RFC 6038, October 2010.

10.2. Informative References

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

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/>

IP Performance Working Group
Internet-Draft
Intended status: Experimental
Expires: April 18, 2013

M. Mathis
Google, Inc
Oct 15, 2012

Curating Internet Measurement Data
draft-mathis-ippm-data-curation-00.txt

Abstract

This document describes the nomenclature, requirements and framework for handling per-sample metadata for a live archive of Internet measurements. By archive, we mean that once captured, the data MUST NOT be altered or deleted. By live, we mean that new data from ongoing measurements are being continuously appended to the archive.

The purpose of the metadata is to support the use of the Scientific Method in the study the archived data. Under the principle of full disclosure, the Scientific Method requires that later researchers be able to repeat earlier studies using the original data, but refined by new insights (such as improved calibration) gained from other intervening studies.

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 18, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

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

Table of Contents

1. Introduction	3
2. Background	3
3. Definitions and Assumptions	4
4. Requirements	6
5. References	7
5.1. Normative References	7
5.2. Informative References	7
Author's Address	7

1. Introduction

This [very preliminary Internet-Draft will eventually] describe the nomenclature, requirements and framework for handling per-sample metadata for a live archive of Internet measurements. By archive, we mean that once captured, the data MUST NOT be altered or deleted. By live, we mean that new data from ongoing measurement are being continuously appended to the archive. Without loss of generality, we model such an archive as a large sparse table where each row is a sample (likely but not required to be a singleton as defined in [RFC2330]) and each column is a measurement parameter or result.

The purpose of the metadata is to support the use of the Scientific Method in the study the live data archive. Under the principle of full disclosure, the Scientific Method requires that later researchers have access to the same tools and raw data set used by previous researchers. The difficulty being that the archive itself has been extended in the time between current researchers' access, and previous researchers' conclusions. A capability must exist to exclude all new data from these investigations. Furthermore, a great deal of knowledge is often gained by detecting systematic error in extant data and retroactively recalibrating well after it was collected. It is extremely important to be able to repeat earlier inquiry using legacy data refined by later calibration studies.

The metadata is to be placed in additional columns that are subject to the same rules as the data itself: columns can be added, but once added they MUST NOT be altered or deleted. This note describe the nomenclature, framework and requirements for handling the metadata.

[At this time our goal is to recruit additional authors.]

2. Background

The advent of large scale online storage and high performance cloud computing has created the opportunity for data mining Internet measurements. Data mining can be described as detecting patterns in large data sets to infer knowledge beyond the scope of the original data or measurement.

In order for inferred results to meet the formal criterion as a scientific endeavour, the data mining analysis MUST be repeatable by later researchers who MUST be able to re-examine the same raw data with the same or similar tools. The additional researcher can confirm the conclusions or consider the possibility of alternative explanations for the patterns detected in the data.

To support data mining as a scientific endeavour the data set must be archival: the only permitted alteration to the data set is to append more data: additional rows as more measurements are performed; additional columns as new annotations are added. Once captured, the data MUST NOT be deleted or altered under any conditions.

All data is subject to measurement error. One common use of data mining is detecting and retroactively compensating for measurement error. In the case of Internet measurement there is also the opportunity for operational events such as topology changes or routing updates to introduce subtle errors or changes in the calibration of the measurements. As a consequence, it is desirable to be able to tag rows with metadata indicating noteworthy operational events. Although most such annotations are entirely benign, occasionally it might be discovered that some of the data is "tainted" when something goes wrong such that the data can not be trusted without additional processing.

An important use case for an archived data set is supporting the following type of scenario: researcher A makes a claim about a pattern observed while mining the data. At a later time researcher B discovers some systematic error present in some of the data and develops a method to recalibrate the data to compensate for the systematic errors. Researcher C wants to reconfirm A's results by running the following series of experiments:

- A's original analysis of the same rows as A used, to confirm the procedure.

- A's original analysis of the same rows as A used, using data recalibrated by B.

- Differential version of A's algorithm to extrapolate how recalibration might affect other results.

- A's original analysis on the data to date, including new rows since A's conclusion.

- A's original analysis on the data to date, using data recalibrated by B.

This process of continuous re-evaluation is the foundation of the Scientific Method.

3. Definitions and Assumptions

Data is assumed to be named (indexed) by N-tuples. Without loss of generality, we further assume a 2 dimensional sparse table where each measurement is a single row, and columns that are parameters or results associated with each measurement, as might be implemented in an SQL like database. These assumptions give us some linguistic shorthand: each row is a singletons[RFC2330], and each column the

parameters, result and metadata data associated with each singleton. In a live archive, the number of rows continues to grow as more data is collected. This note describes conventions and practices for adding additional columns to support robust scientific method.

Other ways to organize the data are completely acceptable. Using some other data organization would change the terminology but not the principles outlined in this document.

Metadata can be added to each row, by adding additional columns to the table. Metadata MUST be subject to the same rules as the measurement data: once added it can not be deleted or altered, otherwise an experiment that used the metadata can not be repeated. To minimize the long term cost of the metadata, it should be designed carefully such that as much as possible each metadata column will remain useful for the life of the data set.

It is explicitly permitted for metadata to be added to existing rows after the data itself has been archived.

Each metadata column must be properly defined: at a minimum a textual description and date the column was first added. Other useful attributes include an indication of who or what authority or process is responsible for adding metadata to new rows, as the archive is extended by live measurement.

It is tempting to try to replace the textual description of the metadata by some formal language specification. However our intuition is that such an effort is likely to be unbound and potentially non-convergent. Therefore, we only provide a couple of examples.

A chronology is a metadata tag for which there are well defined "before" and "after" primitives. Examples include measurement time and archive time, as well as several platform properties: OS version and versions of all component software. The important property of a chronology is that it is useful to specify rows as ranges.

While it is tempting to think that measurement time might be sufficient metadata, consider the difficulty in representing a scenario where a software version change affected some detail of the measurement parameters. If this tool is deployed across a very large fleet of nodes, then making sense of the data requires being able to join the per node update logs with the archived data. In general it would be far easier to label the data with the measurement tool version used for measurement. The researcher then just has to filter the data by comparing tool version.

Another example of a chronology might be to tag measurements with pointers to entries in an operations log that might be tracking topology and configuration changes. This would permit fast exploration of questions such as, "did some network change affect performance in a detectable way?"

A another class of meta data is data recalibration, for instance using an independent means to compute systematic errors present in measurement parameters. Either updated values or error offsets could be stored as metadata. This gives all researchers access to both the raw, uncorrected values and the adjusted or recalibrated values.

Similar to recalibration is addressing tainted data, for example if some event made the data unreliable or inaccurate as pertains to determining some specific metric in a way that can not be calibrated. While it is tempting to entirely discard such data, doing so would invalidate the archive for other sorts of studies which might not be affected by inaccuracy, for example investigating questions of user self selection.

4. Requirements

This section defines a metadata formalism that would permit a data archive to implement computer science grade lock semantics on the data. This is likely to be overkill for nearly all applications, however it does permit an implementer clearly understand where there are assumption that might create potential for race conditions, for example by not having a strong way to assure that empty cells are not changed after they have been accessed.

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

Data in an archive MUST NOT be altered or deleted.

The Archive MUST make a distinction between an empty cell (never written) and a cell containing a null data. e.g. a null string or a null pointer. Each cell is permitted to have one transition from empty to non-empty, and no other transitions.

And many many more....

5. References

5.1. Normative References

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

5.2. Informative References

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

Author's Address

Matt Mathis
Google, Inc
1600 Amphitheater Parkway
Mountain View, California 93117
USA

Email: mattmathis@google.com

IP Performance Working Group
Internet-Draft
Intended status: Experimental
Expires: April 18, 2013

M. Mathis
Google, Inc
Oct 15, 2012

Model Based Internet Performance Metrics
draft-mathis-ippm-model-based-metrics-00.txt

Abstract

We introduce a new class of model based Internet metrics designed to determine if a long path can be expected to meet a predefined end-to-end application performance target by applying a suite of single property tests to successive sections of the long path. In many cases these single property tests are based on existing IPPM metrics, with the addition of success and validity criteria. The sub-path at a time tests are designed to eliminate all known conditions that might prevent the full path from meeting the target performance.

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 18, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Background	4
3. Common Models and Parameters	5
3.1. End-to-end parameters	5
3.2. Per sub-path parameters	7
3.3. Common Calculations for Single Property Tests	7
3.4. Parameter Derating	8
3.5. Single Property Tests Results	9
4. Single Property Tests	9
4.1. Verify the absence of cross traffic	9
4.1.1. Parameter Calculation	10
4.1.2. Cross traffic Measurement	10
4.2. Full Data Rate Loss Rate Tests	10
4.2.1. Loss Rate Measurement	11
4.3. Background Loss Rate Tests	11
4.3.1. Background Loss Rate Measurement	11
4.4. Queue Capacity Test	12
4.4.1. Model Calculation	12
4.4.2. Queue Capacity Measurement	12
4.5. AQM Test	12
4.5.1. Model Calculation	12
4.5.2. AQM Measurement	12
4.6. Reordering Test	12
4.6.1. Model Calculation	12
4.6.2. Reordering Measurement	12
5. Calibration	12
6. References	13
6.1. Normative References	13
6.2. Informative References	13
Appendix A. Model Derivations	13
Appendix B. Old text from an earlier document	13
Author's Address	15

1. Introduction

We introduce a new class of model based metrics designed to determine if a long path can be expected to meet a predefined application end-to-end performance target by applying a suite of single property tests to successive sections of the long path. In many cases these single property tests are based on existing IPPM metrics, with the addition of success and validity criteria. The sub-path at a time tests are designed to eliminate all known conditions that might prevent the full path from meeting the target performance. The end-to-end target performance must be specified in advance, in order to be able to open-loop the control systems (such as congestion control) that are present in all Internet transport protocols and applications. Since a singleton (see [RFC2330]) is only a pass/fail measurement of a sub-path, these metrics are most useful in composition over large pools of samples, such as across a collection of paths or a time interval [RFC5835].

For Bulk Transport Capacity (BTC) the target performance to be measured is a data rate. TCP's ability to compensate for less than ideal network conditions is fundamentally affected by the RTT and MTU of the end-to-end Internet path that it traverses. Since the minimum RTT and maximum MTU are both fixed properties of the path, they are also taken as parameters. The target values for these three parameters, Data Rate, RTT and MTU, are determined by the application, its intended use and the physical infrastructure over which it traverses. They are described in more detail in Section 3 together with the models used to infer the required performance of the underlying Internet fabric.

Traditional end-to-end BTC metrics have proven to be difficult or unsatisfactory for the reasons described in Section 2. Rather than testing the end-to-end path with TCP or other some other BTC, each sub-path is evaluated using suite of far simpler and more predictable single property tests described in Section 4. For BTC the following tests are sufficient: raw data rate, background loss rate, queue burst capacity, reordering extent, onset of congestion/AQM and return path quality. If every sub-path passes all of these tests, then an end-to-end application using any reasonably modern TCP or similar protocol should be able to attain the specified target data rate, over the full end-to-end path at the specified RTT and MTU.

There exists the potential that model based metric might fail, in the sense that every sub-path of an end-to-end path passes every single property test and yet a application might still fail to attain its performance target. If so, then a traditional BTC needs to be used to validate the tests for each sub-path, as described in Section 5.

Future text (or a more likely a future document) will describe model based metrics for real time traffic. The salient point will be that concurrently meeting the goals of both RT and throughput maximizing traffic implicitly requires some form of traffic segregation, such that the two traffic classes are not placed in the same queue. Some technique as simple as SFQ[SFQ] might be a sufficient alternative to full QoS.

TODO:

Add discussion of protocol overhead: MSS vs IP MTU vs link MTU

2. Background

(Fragments of earlier text)

The holy grail of IPPM has been BTC measurement, but it has proven to be a very hard problem for a number of reasons:

TCP is a control systems - everything affects performance, including components that are explicitly not part of the the test. Congestion control is an equilibrium process, transport protocols change the network (raise loss probability and/or RTT) to conform to their behavior.

TCP's ability to compensate for network flaws is directly proportional to the number of round trips per second (e.g. inversely proportional to the RTT). As a consequence a flawed link that passes a local test is likely to completely fail when the path is extended by a perfect network to some larger RTT. TCP has a meta Heisenberg problem - Measurement and cross traffic interact in unknown and ill defined ways. The situation is actually worse than the traditional physics problem where you can at least estimate the relative masses of the measurement and measured particles. For network measurement you can not in general determine the relative "masses" of the measurement traffic and cross traffic, so you can not even gage the relative magnitude of their effects on each other.

The new approach is to "open loop" congestion control. Defeat CC, typically by throttling TCP to a lower rate, such that it does not respond to network conditions. In this mode the measurement software explicitly controls TCP's state variables (e.g. cwnd) to create controlled traffic patterns, which are manipulated to measure the network.

Models are used to determine the actual test parameters (loss rate, etc) from the target parameters. The basic method is to use models to estimate simple network properties required to sustain a given transport flow (or set of flows), and using a suite of simpler

metrics to confirm that the network meets the required properties. For example a network can sustain a Bulk TCP flow of a given data rate, MTU and RTT when 4 (and probably more) conditions are met:

- The raw link rate is higher than the target data rate.

- The raw packet loss rate is lower than required by a suitable TCP performance model

- There is sufficient buffering at any bottleneck smooth bursts.

- When the link is overfilled (congested), the onset of packet loss is progressive.

These condition can all be verified with simple tests, using model parameters and acceptance thresholds derived from the target data rate, MTU and RTT. Note that this procedure is not invertible: a singleton measurement is a pass/fail evaluation of a given path or subpath at a given performance. Measurements to confirm that a link passes at one particular performance may not be generally be useful to predict if the link will pass at a different performance.

Although they are not invertible, they do have several other valuable properties, such as natural ways to define several different composition metrics.

3. Common Models and Parameters

Transport performance models are used to derive the test parameters for each single property test from the end-to-end target parameters and additional ancillary parameters.

It is envisioned that the modeling phase (to compute the test parameters) and testing phases will be decoupled. This section covers common derived parameters, used by multiple single property tests. For some tests, additional modeling is described with the tests.

Since some aspects of the models are very conservative, the modeling framework permits some latitude in derating test parameters, as described in Section 3.4.

For certain sub-paths (e.g. common types of access links) it would be appropriate for the single property test parameters to be documented as a "measurement profile" together with the modeling assumptions and derating factors described in Section 3.3 and Section 3.4.

3.1. End-to-end parameters

These parameters are determined by the needs of the application or the ultimate end user and the end-to-end Internet path.

Target Data Rate: The application or ultimate user's performance goal (in aggregate across all connections).

Permitted Number of Connections: The target rate can be more easily obtained by dividing the traffic across more than one connection. In general the number of concurrent connections is determined by the application, however see the comments below on multiple connections.

Target RTT (Round Trip Time): For fundamental reasons a long path makes it more difficult for TCP or other transport protocol to meet the target rate. The target RTT must be representative for the actual applications expected to use the network. This parameter may be subject to a future convention (e.g. continental scale paths should be assumed to be some fixed RTT, such as 100 ms) or alternatively be an property of an ISP's topology (e.g. a ISP with richer and better placed peering may actually have lower RTTs for typical users.)

Target MTU (Maximum Transmission Unit): Assume 1500 Bytes unless otherwise specified. If some sub-path forces a smaller MTU, then all sub-paths must be tested with the same smaller MTU.

The use of multiple connections has been very controversial since the beginning of the World-Wide-Web[first complaint]. Modern browsers open many connections [BScope]. Experts in the IETF transport area have frequently spoken against this practice [long list]. It is not inappropriate to assume some small number of concurrent connections (e.g. 4 or 6), to compensate for limitation in TCP. However, choosing too large a number is at risk of being taken as a signal by the web browser community that this practice has been embraced by the Internet community. It may not be desirable to send such a signal.

The following optional parameters apply for testing generalized end-to-end paths that include subpaths with known specific types of behaviors that are not well represented by simple queueing models:

Bottleneck link clock rate: This applies to links that are using virtual queues or other techniques to police or shape users traffic at lower rates full link rate. The bottleneck link clock rate should be representative of queue drain times for short bursts of packets on an otherwise unloaded link.

Channel hold time: For channels that have relatively expensive channel arbitration algorithms, this is the typical (maximum?) time that data and or ACKs are held pending acquiring the channel. While under heavy load, the RTT may be inflated by this parameter, unless it is built into the target RTT

Preload traffic volume: If the user's traffic is shaped on the basis of average traffic volume, this is volume necessary to invoke "heavy hitter" policies.

Unloaded traffic volume: If the user's traffic is shaped on the basis of average traffic volume, this is the maximum traffic volume that a test can use and stay within a "light user" policies.

Note on a ConEx enabled network [ConEx], the word "traffic" in the last two items should be replaced by "congestion" i.e. "preload congestion volume" and "unloaded congestion volume".

3.2. Per sub-path parameters

Some single parameter tests also need parameter of the sub-path.

sub-path RTT: RTT of the sub-path under test.

sub-path link clock rate: If different than the Bottleneck link clock rate

3.3. Common Calculations for Single Property Tests

The most important derived parameter is `target_pipe_size` (in packets), which is the number of packets needed exactly meet the target rate, with no cross traffic for the specified RTT and MTU. It is given by:

$$\text{target_pipe_size} = \text{target_rate} * \text{target_RTT} / \text{target_MTU}$$

If the transport protocol (e.g. TCP) average window size is smaller than this, the link will be under filled.

If `target_data_rate` is equal to `bottleneck link_data_rate`, then `target_pipe_size` also predicts the onset of queueing. If the transport protocol (e.g. TCP) average window size is larger than the `target_pipe_size`, the excess packets will be in a standing queue at the bottleneck.

If the transport protocol is using Reno congestion control [RFC5681], then there must be `target_pipe_size` roundtrips between losses. Otherwise the multiplicative window reduction triggered by a loss would cause the network to be underfilled. Following [MSM097], we derive the losses must be no more frequent than every 1 in $(3/2)(\text{target_pipe_size}^2)$ packets. This provides the reference value for `target_run_length` which is typically the number of packets that must be delivered between loss episodes in the tests below:

$$\text{reference_target_run_length} = (3/2)(\text{target_pipe_size}^2)$$

Note that this calculation is based on a number of assumptions that may not apply. Appendix A discusses these assumptions and provides

some alternative models. The actual method for computing `target_run_length` MUST be published along with the rationale for the underlying assumptions and the ratio of chosen `target_run_length` to `reference_target_run_length`.

Although this document gives a lot of latitude for calculating `target_run_length` people specifying profiles for suites of single property tests need to consider the effect of their choices on the ongoing conversation and tussle about the relevance of "TCP friendliness" as an appropriate model for capacity allocation. Choosing a `target_run_length` that is substantially smaller than `reference_target_run_length` is equivalent to saying that it is appropriate for the research community to abandon "TCP friendliness" as a fairness model and to develop more aggressive Internet transport protocols, and for applications to continue (or even increase) the number of connections that they open.

The calculations for individual parameters are presented with the each single property test. In general these calculations are permitted some as described in Section 3.4

3.4. Parameter Derating

Since some aspects of the models are very conservative, the modeling framework permits some latitude in derating some specific test parameters, as indicated in Section 4. For example classical performance models suggest that in order to be sure that a single TCP stream can fill a link, it needs to have a full bandwidth-delay-product worth of buffering at the bottleneck[`QueueSize`]. In real networks with real applications this is sometimes overly conservative. Rather than trying to formalize more complicated models we permit some test parameters to be relaxed as long as they meet some additional procedural constraints:

- The method used compute and justify the derated metrics is published in such a way that it becomes a matter of public record.
- The calibration procedures described in Section 5 are used to demonstrate the feasibility of meeting the performance targets with the derated test parameters.

- The calibration process itself is documented in such a way that other researchers can duplicate the experiments and validate the results.

Note that some single property test parameters are not permitted to be derated.

3.5. Single Property Tests Results

TBD Define: Pass, Fail and inconclusive test results.

The inconclusive outcome is needed to address the case where a test failed to attain the specified test conditions. This is important to the extent that the tests themselves have built in control systems which might interfere with some aspect of the test. It is required for example to use TCP for testing.

4. Single Property Tests

The single property tests confirm that each sub-path can sustain the normal traffic patterns caused by TCP running at the specified target performance. Specifically they confirm that each sub-path has: sufficient raw capacity (e.g. sufficient data rate); low enough background loss rate where mandatory congestion control stays out of the way; large enough queue space to absorb TCP's normal bursts; does not cause unreasonable packet reordering; progressive AQM to appropriately invoke congestion control. Appropriately invoking congestion control requires that packet losses or ECN marks start progressively before TCP creates an excessive sustained queues[BufferBloat] or excessively bursty losses. The return path must also subject to a similar suite of tests, although potentially with different test parameters.

Note that many of the sub-path tests resemble metrics that have already been defined in the IPPM context, with the addition of criteria for passing or failing the test. The models used to derive the test parameters make specific assumptions about network conditions, a test is deemed "inconclusive" (as opposed to failing) if tester does not meet the underlying assumption. For example a loss rate test at a specified data rate is inconclusive if the tester fails to send data at the specified rate for some reason. This concept of an inconclusive test is necessary to build tests out of protocols or technologies that they themselves have built in or implicit control systems.

Some single property test can be combined, since their parameters are not mutually exclusive.

4.1. Verify the absence of cross traffic

Use a passive packet or SNMP monitoring to verify that the traffic volume on the sub-path agrees with the traffic generated by each test. Ideally this should be performed before during and after each test.

The goal is provide quality assurance on the overall measurement process, and specifically to detect the following measurement failure: a user observes unexpectedly poor application performance, the ISP observes that the access link is running at the rated capacity. Both fail to observe that the user's computer has been infected by a virus which is spewing traffic as fast as it can.

Parameters:

Maximum Cross Traffic Data Rate The amount of excess traffic permitted. Note that this might be different for different tests.
Maximum Data Rate underage The permitted amount that the traffic can be less than predicted for the current test. Normally this would just be a statement of the maximum permitted measurement error, however it might also detect cases where the passive and active tests are misaligned: testing different subscriber lines. This is important because the vantage points are so different: in-band active measurement vs out-of-band passive measurement.

4.1.1. Parameter Calculation

TBA

4.1.2. Cross traffic Measurement

One possible method is an adaptation of: [www-didc.lbl.gov/papers/SCNM-PAM03.pdf](http://www.didc.lbl.gov/papers/SCNM-PAM03.pdf) D Agarwal etal. "An Infrastructure for Passive Network Monitoring of Application Data Streams". Use the same technique as that paper to trigger the capture of SNMP statistics for the link.

4.2. Full Data Rate Loss Rate Tests

We propose two versions of the loss rate test. One, performed at data full rate, is intrusive and recommend for infrequent testing, such as when a service is first turned up or as part of an auditing process. Note that this test also implicitly confirms that sub_path has sufficient capacity to carry the target_data_rate.

The second background loss rate, described below, is designed for ongoing monitoring for change in sub-path quality.

Parameters:

Run Length Same as target_run_lenght
Data Rate Same as target_data_rate

Note that these parameters MUST NOT be derated. If the default parameters are too stringent use an alternate model for target_data_rate as described in Appendix A.

4.2.1. Loss Rate Measurement

Data is sent at the specified `data_rate`. The receiver accumulates the total data delivered and packets lost [and ECN marks, which are nominally treated as losses by conforming transport protocols]. The observed `average_run_lenght` is computed from `total_data_delivered` divided by the `total_loss_rate`. A [TBD] statistical test is applied to determine when or if the `average_run_lenght` is larger than `target_run_lenght`.

TODO: add language about monitoring cross traffic.

The test is deemed to have passed only if the observed data rate matches the `target_data_rate` and it is statistically significant that the `average_run_lenght` is larger than `target_run_lenght`. It is deemed inconclusive if: the statistical test is inconclusive; there is too much background load; or the `target_data_rate` could not be attained.

4.3. Background Loss Rate Tests

The background loss rate is designed for ongoing monitoring for change in sub-path quality. It should be used in conjunction with the above full rate test.

Parameters:

Run Length Same as `target_run_lenght`

Data Rate Some small fraction of `target_data_rate`, such as 1%.

4.3.1. Background Loss Rate Measurement

The receiver accumulates the total data delivered and packets losses [and ECN marks, which are nominally treated as losses by conforming transport protocols]. The observed `average_run_lenght` is computed from `total_data_delivered` divided by the `total_loss_rate`. A [TBD] statistical test is applied to determine when or if the `average_run_lenght` is larger than `target_run_lenght`.

TODO: add language about monitoring cross traffic.

The test is deemed to have passed if it is statistically significant that the `average_run_lenght` is larger than `target_run_lenght`. It is deemed inconclusive if there is too much background traffic or the statistical test is inconclusive.

4.4. Queue Capacity Test

Parameters:

TBA TBA

4.4.1. Model Calculation

TBA

4.4.2. Queue Capacity Measurement

TBA

4.5. AQM Test

Parameters:

TBA TBA

4.5.1. Model Calculation

TBA

4.5.2. AQM Measurement

TBA

4.6. Reordering Test

Parameters:

TBA TBA

4.6.1. Model Calculation

TBA

4.6.2. Reordering Measurement

TBA

5. Calibration

If using derated metrics, or when something goes wrong, the results must be calibrated against a traditional BTC.....

6. References

6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

6.2. Informative References

- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, May 1998.
 - [RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, September 2009.
 - [RFC5835] Morton, A. and S. Van den Berghe, "Framework for Metric Composition", RFC 5835, April 2010.
 - [MSMO97] Mathis, M., Semke, J., Mahdavi, J., and T. Ott, "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm", Computer Communications Review volume 27, number3, July 1997.
 - [BScope] Browserscope, "Browserscope Network tests", Sept 2012, <<http://www.browserscope.org/?category=network>>.
- See Max Connections column

Appendix A. Model Derivations

This appendix describes several different ways to calculate `target_run_length` and the implication of the chosen calculation.

Rederive MSMO97 under two different assumptions: `target_rate = link_rate` and `target_rate < 2 * link_rate`.

Show equivalent derivation for CUBIC.

Commentary on the consequence of the choice.

Appendix B. Old text from an earlier document

To be moved, removed or absorbed

Step 0: select target end-to-end parameters: a target rate and target RTT. The primary test will be to confirm that the link quality is sufficient to meet the specified target rate for the link under test, when extended to the target RTT by an ideal network. The target rate must be below the actual link rate and nominally the target RTT would be longer than the link RTT. There should probably be a convention for the relationship between link and target rates (e.g. 85%).

For example on a 10 Mb/s link, the target rate might be 1 MBytes/s, at an RTT of 100 mS (a typical continental scale path).

Step 1: On the basis of the target rate and RTT and your favorite TCP performance model, compute the "required run length", which is the required number of consecutive non-losses between loss episodes. The run length resembles one over the loss probability, if clustered losses only count as a single event. Also select "test duration" and "test rate". The latter would nominally be the same as the target rate, but might be different in some situations. There must be documentation connecting the test rate, duration and required run length, to the target rate and RTT selected in step 0.

Continuing the above example: Assuming a 1500 Byte MTU. The calculated model loss rate for a single TCP stream is about 0.01% (1 loss in 1E4 packets).

Step 2, the actual measurement proceeds as follows: Start an unconstrained bulk data flow using any modern TCP (with large buffers and/or autotuning). During the first interval (no rate limits) observe the slowstart (e.g. tcpdump) and measure: Peak burst size; link clock rate (delivery rate for each round); peak data rate for the fastest single RTT interval; fraction of segments lost at the end of slow start. After the flow has fully recovered from the slowstart (details not important) throttle the flow down to the test rate (by clamping cwnd or application pacing at the sender or receiver). While clamped to the test rate, observe the losses (run length) for the chosen test duration. The link passes the test if the slowstart ends with less than approximately 50% losses and no timeouts, the peak rate is at least the target rate, and the measured run length is better than the required run length. There will also need to be some ancillary metrics, for example to discard tests where the receiver closes the window, invalidating the slowstart test. [This needs to be separated into multiple subtests]

Optional step 3: In some cases it might make sense to compute an "extrapolated rate", which is the minimum of the observed peak rate, and the rate computed from the specified target RTT and the observed run length by using a suitable TCP performance model. The extrapolated rate should be annotated to indicate if it was run

length or peak rate limited, since these have different predictive values.

Other issues:

If the link RTT is not substantially smaller than the target RTT and the actual run length is close to the target rate, a standards compliant TCP implementation might not be effective at accurately controlling the data rate. To be independent of the details of the TCP implementation, failing to control the rate has to be treated as a spoiled measurement, not a infrastructure failure. This can be overcome by "stiffening" TCP by using a non-standard congestion control algorithm. For example if the rate controlling by clamping cwnd then use "relentless TCP" style reductions on loss, and lock ssthresh to the cwnd clamp. Alternatively, implement an explicit rate controller for TCP. In either case the test must be abandoned (aborted) if the measured run length is substantially below the target run length.

If the test is run "in situ" in a production environment, there also needs to be baseline tests using alternate paths to confirm that there are no bottlenecks or congested links between the test end points and the link under test.

It might make sense to run multiple tests with different parameters, for example infrequent tests with test rate equal to the target rate, and more frequent, less disruptive tests with the same target rate but the test rate equal to 1% of the target rate. To observe the required run length, the low rate test would take 100 times longer to run.

Returning to the example: a full rate test would entail sending 690 pps (1 MByte/s) for several tens of seconds (e.g. 50k packets), and observing that the total loss rate is below 1:1e4. A less disruptive test might be to send at 6.9 pps for 100 times longer, and observing

Formatted: Mon Oct 15 16:00:51 PDT 2012

Author's Address

Matt Mathis
Google, Inc
1600 Amphitheater Parkway
Mountain View, California 93117
USA

Email: mattmathis@google.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 11, 2013

J. Fabini
Vienna University of Technology
A. Morton
AT&T Labs
October 8, 2012

Advanced Stream and Sampling Framework for IPPM
draft-morton-ippm-2330-update-00

Abstract

To obtain repeatable results in modern networks, test descriptions need an expanded stream parameter framework that also augments aspects specified as Type-P for test packets. This memo proposes to update the IP Performance Metrics (IPPM) Framework with advanced considerations for measurement methodology and testing. The existing framework mostly assumes deterministic connectivity, and that a single test stream will represent the characteristics of the path when it is aggregated with other flows. Networks have evolved and test stream descriptions must evolve with them, otherwise unexpected network features may dominate the measured performance. This memo describes new stream parameters for both network characterization and support of application design using IPPM metrics.

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 11, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
2. Scope	3
3. New Stream Parameters	3
3.1. Test Packet Type-P	5
3.1.1. Test Packet Length	5
3.1.2. Test Packet Payload Content Optimization	5
3.2. Packet History	6
3.3. Access Technology Change	6
3.4. Time-Slotted Network Paths	6
4. Conclusions	7
5. Security Considerations	8
6. IANA Considerations	8
7. Acknowledgements	8
8. References	8
8.1. Normative References	8
8.2. Informative References	9
Authors' Addresses	10

1. Introduction

The IETF IP Performance Metrics (IPPM) working group first created a framework for metric development in [RFC2330]. This framework has stood the test of time and enabled development of many fundamental metrics, while only being updated once in a specific area [RFC5835].

The IPPM framework [RFC2330] generally relies on several assumptions, one of which is not explicitly stated but assumed: the network behaves (halfway) deterministic and without state/history-less (with some exceptions, firewalls are mentioned). However, this does not hold true for many modern network technologies, such as reactive networks (those with demand-driven resource allocation) and links with time-slotted operation. Per-flow state can be observed on test packet streams, and such treatment will influence network characterization if it is not taken into account. Flow history will also affect the performance of applications and be perceived by their users.

Moreover, Sections 4 and 6.2 of [RFC2330] explicitly recommend repeatable measurement metrics and methodologies. Measurements in today's access networks illustrate that methodological guidelines of [RFC2330] must be extended to capture the reactive nature of these networks. Although the proposed extensions can support methodologies to fulfill the continuity requirement stated in section 6.2 of [RFC2330], there is no guarantee. Practical measurements confirm that some link types exhibit distinct responses to repeated measurements with identical stimulus, i.e., identical traffic patterns. If feasible, appropriate fine-tuning of measurement traffic patterns can improve measurement continuity and repeatability for these link types as shown in [IBD].

2. Scope

The scope of this memo is to describe useful stream parameters in addition to the information in Section 11.1 of [RFC2330] and described in [RFC3432] for periodic streams. The purpose is to foster repeatable measurement results in modern networks by highlighting the key aspects of test streams and packets and make them part of the IPPM performance metric framework.

3. New Stream Parameters

There are several areas where measurement methodology definition and test result interpretation will benefit from an increased understanding of the stream characteristics and the (possibly

unknown) network condition that influence the measured metrics.

1. Network treatment depends on the fullest extent on the "packet of Type-P" definition in [RFC2330], and has for some time.
 - * State is often maintained on the per-flow basis at various points in the network, where "flows" are determined by IP and other layers. Significant treatment differences occur with the simplest of Type-P parameters: packet length.
 - * Payload content optimization (compression or format conversion) in intermediate segments. This breaks the convention of payload correspondence when correlating measurements made at different points in a path.
2. Packet history (instantaneous or recent test rate or inactivity, also for non-test traffic) profoundly influences measured performance, in addition to all the Type-P parameters described in [RFC2330].
3. Access technology may change during testing. A range of transfer capacities and access methods may be encountered during a test session. When different interfaces are used, the host seeking access will be aware of the technology change which differentiates this form of path change from other changes in network state. Section 14 of [RFC2330] treats the possibility that a host may have more than one attachment to the network, and also that assessment of the measurement path (route) is valid for some length of time (in Section 5 and Section 7 of [RFC2330]). Here we combine these two considerations under the assumption that changes may be more frequent and possibly have greater consequences on performance metrics.
4. Paths including links or nodes with time-slotted service opportunities represent several challenges to measurement (when service time period is appreciable):
 - * Random/unbiased sampling is not possible beyond one such link in the path.
 - * The above encourages a segmented approach to end to end measurement, as described in [RFC6049] for Network Characterization (as defined in [RFC6703]) to understand the full range of delay and delay variation on the path. Alternatively, if application performance estimation is the goal (also defined in [RFC6703]), then a stream with un-biased or known-bias properties [RFC3432] may be sufficient.

- * Multi-modal delay variation makes central statistics unimportant, others must be used instead.

Each of these topics is treated in detail below.

3.1. Test Packet Type-P

We recommend two Type-P parameters to be added to the factors which have impact on network performance measurements, namely packet length and payload type. Carefully choosing these parameters can improve measurement methodologies in their continuity and repeatability when deployed in reactive networks.

3.1.1. Test Packet Length

Many instances of network characterization using IPPM metrics have relied on a single test packet length. When testing to assess application performance or an aggregate of traffic, benchmarking methods have used a range of fixed lengths and frequently augmented fixed size tests with a mixture of sizes, or IMIX as described in [I-D.ietf-bmwg-imix-genome].

Test packet length influences delay measurements, in that the IPPM one-way delay metric [RFC2679] includes serialization time in its first-bit to last bit time stamping requirements. However, different sizes can have a larger effect on link delay and link delay variation than serialization would explain alone. This effect can be non-linear and change instantaneous or future network performance.

Repeatability is a main measurement methodology goal as stated in section 6.2 of [RFC2330]. To eliminate packet length as a potential measurement uncertainty factor, successive measurements must use identical traffic patterns. In practice a combination of random payload and random start time can yield representative results as illustrated in [IRR].

3.1.2. Test Packet Payload Content Optimization

The aim for efficient network resource use has resulted in a series of "smart" networks to deploy server-only or client-server lossless or lossy payload compression techniques on some links or paths. These optimizers attempt to compress high-volume traffic in order to reduce network load. Files are analyzed by application-layer parsers and parts (like comments) might be dropped. Although typically acting on HTTP or JPEG files, compression might affect measurement packets, too. In particular measurement packets are qualified for efficient compression when they use standard plain-text payload.

IPPM-conforming measurements should add packet payload content as a Type-P parameter which can help to improve measurement determinism. Some packet payloads are more susceptible to compression than others, but optimizers in the measurement path can be out ruled by using incompressible packet payload. This payload content could be either generated by a random device or by using part of a compressed file (e.g., a part of a ZIP compressed archive).

3.2. Packet History

Recent packet history and instantaneous data rate influence measurement results for reactive links supporting on-demand capacity allocation. Measurement uncertainty may be reduced by knowledge of measurement packet history and total host load. Additionally, small changes in history, e.g., because of lost packets along the path, can be the cause of large performance variations.

For instance delay in reactive 3G networks like High Speed Packet Access (HSPA) depends to a large extent on the test traffic data rate. The reactive resource allocation strategy in these networks affects the uplink direction in particular. Small changes in data rate can be the reason of more than 200% increase in delay, depending on the specific packet size.

3.3. Access Technology Change

[RFC2330] discussed the scenario of multi-homed hosts. If hosts become aware of access technology changes (e.g., because of IP address changes or lower layer information) and make this information available, measurement methodologies can use this information to improve measurement representativeness and relevance.

However, today's various access network technologies can present the same physical interface to the host. A host may or may not become aware when its access technology changes on such an interface. Measurements for networks which support on-demand capacity allocation are therefore challenging in that it is difficult to differentiate between access technology changes (e.g., because of mobility) and reactive network behavior (e.g., because of data rate change).

3.4. Time-Slotted Network Paths

Time-Slotted operation of network entities - interfaces, routers or links - in a network path is a particular challenge for measurements, especially if the time slot period is substantial. The central observation as an extension to Poisson stream sampling in [RFC2330] is that the first such time-slotted component cancels unbiased measurement stream sampling. In the worst case, time-slotted

operation converts an unbiased, random measurement packet stream into a periodic packet stream. Being heavily biased, these packets may interact with periodic network behavior of subsequent time-slotted network entities.

Practical measurements confirm that such interference limits delay measurement variation to a sub-set of theoretical value range. Measurement samples for such cases can aggregate on artificial limits, generating multi-modal distributions as demonstrated in [IRR]. In this context, the desirable measurement sample statistics differentiate between multi-modal delay distributions caused by reactive network behavior and the ones due to time-slotted interference.

The amount of measurement bias is determined by the relative offset between allocated time-slots in subsequent network entities, delay variation in these networks, and other sources of variation. Measurement results might change over time, depending on how accurately the sending host, receiving host, and time-slotted components in the measurement path are synchronized to each other and to global time. If network segments maintain flow state, flow parameter change or flow re-allocations can cause substantial variation in measurement results.

Measurement methodology selection for time-slotted paths depends to a large extent on the respective viewpoint. End-to-end metrics can provide accurate measurement results for short-term sessions and low likelihood of flow state modifications. Applications or services which aim at approximating network performance for a short time interval (in the order of minutes) and expect stable network conditions should therefore prefer end-to-end metrics. Here stable network conditions refer to any kind of global knowledge concerning measurement path flow state and flow parameters.

However, if long-term forecast of time-slotted network performance is the main measurement goal, a segmented approach relying on hop-by-hop metrics is preferred. Re-generating unbiased measurement traffic at any hop can help to unleash the true range of network performance for all network segments.

4. Conclusions

Safeguarding continuity and repeatability as key properties of measurement methodologies is highly challenging and sometimes impossible in reactive networks. Measurements in networks with demand-driven allocation strategies must use a prototypical application packet stream to infer a specific application's

performance. Measurement repetition with unbiased network and flow states (e.g., by rebooting measurement hosts) can help to avoid interference with periodic network behavior, randomness being a mandatory feature for avoiding correlation with network timing. Inferring from one measurement session or packet stream onto network performance for alternative streams is highly discouraged in reactive networks because of the huge set of global parameters which influence on instantaneous network performance.

5. Security Considerations

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

6. IANA Considerations

This memo makes no requests of IANA.

7. Acknowledgements

The authors thank folks for reading and commenting on this draft.

8. References

8.1. Normative References

- [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.
- [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.
- [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.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.
- [RFC5657] Dusseault, L. and R. Sparks, "Guidance on Interoperation and Implementation Reports for Advancement to Draft Standard", BCP 9, RFC 5657, September 2009.
- [RFC5835] Morton, A. and S. Van den Berghe, "Framework for Metric Composition", RFC 5835, April 2010.
- [RFC6049] Morton, A. and E. Stephan, "Spatial Composition of Metrics", RFC 6049, January 2011.
- [RFC6576] Geib, R., Morton, A., Fardid, R., and A. Steinmitz, "IP Performance Metrics (IPPM) Standard Advancement Testing", BCP 176, RFC 6576, March 2012.
- [RFC6703] Morton, A., Ramachandran, G., and G. Maguluri, "Reporting IP Network Performance Metrics: Different Points of View", RFC 6703, August 2012.

8.2. Informative References

- [I-D.ietf-bmwg-imix-genome] Morton, A., "IMIX Genome: Specification of variable packet sizes for additional testing", draft-ietf-bmwg-imix-genome-02 (work in progress), July 2012.
- [IBD] Fabini, J., "The Illusion of Being Deterministic - Application-Level Considerations on Delay in 3G HSPA Networks", Lecture Notes in Computer Science, Springer, Volume 5550, 2009, pp 301-312 , May 2009.
- [IRR] Fabini, J., "The Importance of Being Really Random: Methodological Aspects of IP-Layer 2G and 3G Network Delay Assessment", ICC'09 Proceedings of the 2009 IEEE International Conference on Communications, doi: 10.1109/ICC.2009.5199514, June 2009.

Authors' Addresses

Joachim Fabini
Vienna University of Technology
Favoritenstrasse 9/E389
Vienna, 1040
Austria

Phone: +43 1 58801 38813
Fax: +43 1 58801 38898
Email: Joachim.Fabini@tuwien.ac.at
URI: <http://www.tc.tuwien.ac.at/about-us/staff/joachim-fabini/>

Al Morton
AT&T Labs
200 Laurel Avenue South
Middletown, NJ 07748
USA

Phone: +1 732 420 1571
Fax: +1 732 368 1192
Email: acmorton@att.com
URI: <http://home.comcast.net/~acmacm/>

IPPM
Internet-Draft
Intended status: Informational
Expires: April 15, 2013

L. Sun
BUPT
F. Yu
Huawei Technologies
W. Wang
BUPT
October 12, 2012

Flow-based Performance Measurement
draft-sun-ippm-flowbased-pm-00

Abstract

The performance measurements of service flow are becoming significant important for administrators monitoring the fitness of the network. This memo defines an end-to-end flow-based performance measurement method, which is achieved by generating synthetic measurement packets, injecting them to the network and analyzing the statistics carried in the measurement packets. This measurement method can measure flow characteristics such as delay, ipdv (IP Packet Delay Variation) and packet loss.

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 15, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

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

Table of Contents

1. Introduction	4
2. Problems statement	4
3. Conventions and Terminology	6
3.1. Conventions Used in This Document	6
3.2. Terminology	6
4. Overview	6
4.1. Goals and Motivation	6
4.2. Protocol overview	7
4.3. Logical Model	8
5. Connection Control	9
5.1. Connection Activation	9
5.2. Connection Deactivation	11
6. Measurement Process	12
6.1. FPM Initiator behavior	12
6.2. FPM Responder behavior	14
7. Metrics	15
7.1. Example of loss rate	16
8. Exception Handling	17
8.1. FM/BR Packet Loss	17
8.2. Packet Reordering	17
9. Use Case	18
10. Security Considerations	19
11. IANA Considerations	19
12. Acknowledgments	20
13. References	20
13.1. Normative Reference	20
13.2. Informative References	20
Authors' Addresses	21

1. Introduction

The IETF IP Performance Metrics (IPPM) working group has defined a series of standard metrics that can be applied to the quality, performance and reliability of Internet data delivery services. The WG has produced protocols to enable communication among test equipment that implements the one- and two-way metrics (OWAMP and TWAMP respectively).

This memo introduces a new measurement method which is called FPM. It continues as follows. Section 2 discusses the existing problems and puts forward the motivation of FPM. Section 3 introduces terminology, followed by the overview of the FPM in Section 4. Section 5 and Section 6 introduce connection control and measurement process of the FPM respectively. Section 7 describes the usage of metrics in FPM which are defined in the current IPPM working group. Section 8 describes some exceptions and handling. At last it introduces a use case, which describes the deployment, characteristics and applications of FPM.

2. Problems statement

The TWAMP protocol proposed by IPPM WG provides a simple and useful network performance measurement method. It aims at using a safe and effective way to measure the performance of the IP network. TWAMP uses TWAMP-Control protocol to initiate, start, and stop test sessions, making the measurement process with more flexibility and security. It uses TWAMP-Test protocol for the actual network test by injecting test packets into network, so that various properties of the network can be measured offline effectively. However, while TWAMP is able to achieve most network measurement situations, it does not work well in some cases on the performance testing for real time business.

In some cases, it needs to monitor the various time-varying performance indexes of the IP network, the performance measurement should be based on real service stream and reflect the real performance of the network. For measuring the performance of the real service stream, TWAMP has the following defects due to the limit of measurement parameters and its framework.

- o TWAMP is mainly used to measure the performance of the network. It cannot be well applied to the real-time performance measurement of a particular or one kind of applications.
- o In the case of real time measurement of network performance, if the test packets are sent too frequently, the network load will be

increased and application flows will be affected. Otherwise, if the number of test packets is small, the performance of the network cannot be reflected accurately by the measurement.

For example, for real time streaming media services such as IPTV and video conference, packets carry QoS parameters to ensure service quality for different application flows. Network needs to real time monitor the performance of these application flows, in order to adjust the allocation of resources in network according to the monitoring results in real time, thereby ensuring the QoS requirements of different applications. For the performance measurement of such business discussed above, TWAMP cannot meet all the requirements well as a result of the above defects.

For the problems discussed above, it is required to define a new measurement framework, which is able to meet the demand for real time measurement of application flows, and does not have much impact on the data flow itself. At the same time, this new measurement method will be able to meet the following goals of the IPPM measurement: guarantee the Service Level Agreement (SLA) provided to the customers, detect/locate the network performance defects, react in response to performance degradation or the failures promptly, and optimize the network resources utilization.

In the following section, we discuss the requirements of IP performance measurement in IP mobile backhaul network.

In mobile operator's backhaul network, there must be a performance monitoring mechanism to check the traffic status in the network. With the status information, some strategies can be implemented on entities. For example, eNodeB can implement online congestion control and bandwidth adjustment strategy based on the performance monitoring result. Hence, IPPM mechanism is required to provide a reasonable estimate of the amount of delay, ipdv (IP Packet Delay Variation) and packet loss in the backhaul network connecting it to the GW.

In order to avoid adding superfluous traffic to the backhaul and leading to the increase of the load level in the network, it is necessary that the frequency of measurement packets generation is kept at a minimum. Moreover, these packets should not lead to an excessive computational overload on the eNodeB and the GW. In other words, the process of generation of these probe packets should be simple and must not overload the transport interface. The packets must be small and infrequent so as to not cause un-necessary overload on the backhaul bandwidth.

Applications or traffic in mobile backhaul network are divided into

multiple bearers with proper mobile QoS parameters (e.g. QCI). If the mobile network would manage bearers as QoS and applications, then the performance of backhaul is more like to be based on applications or QoS. The currently active measurement method (e.g. TWAMP) may be able to do flow-based measurement by specifying DSCP for the TWAMP-test packets. But it can not well support the online measurement and the length of test packet is changeless and not varying as the real service packets. The average performance indexes measured by the active measurement method may not be suitable in these cases.

A new measurement method can be applied into backhaul network, by deploying an end-to-end performance monitor on eNodeB to assist eNodeB to execute the congestion control and flow scheduling. Played as sender entity, eNodeB sends out the OAM packets periodically to trigger the other end (e.g. another eNodeB or an SGW) replying acknowledgement packets, then to estimate the delay, ipdv (IP Packet Delay Variation) and packet loss of each application flow by collecting and calculating status information.

3. Conventions and Terminology

3.1. Conventions Used in This Document

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

3.2. Terminology

FPM: Flow-based Performance Measurement

FM:Forward Monitoring

BR:Backward Reporting

4. Overview

4.1. Goals and Motivation

It is required to provide a reasonable estimation measurement of delay, ipdv (IP Packet Delay Variation) and packet loss in IP network (such as backhaul network). The above parameters are functions of time, which are stochastic in nature. Therefore, the mechanism is required to provide statistical condition estimations of the IP link status. Since measurement injects some OAM packets to the network, it is necessary that the frequency of packet generation is kept at a

minimum. Moreover, these packets should not lead to an excessive computational overload on the measure device. In other words, the generation process of these measurement packets should be simple and must not overload the transport interface. The packets must be small and infrequent so as not to cause unnecessary overload on the network bandwidth or influence the service running on the network.

4.2. Protocol overview

Firstly we make some statement for FPM. We define a measurement method that can be used in some scene. The method proposed here is an end-to-end measurement method over IP layer; it can be implemented under the tunnel mode of IPSec. Two types of logical entities are defined, the FPM Initiator and the FPM Responder. The FPM process consists of two parts: Connection control and Measurement Process.

During Connection control phase, the FPM initiator sends a request to the FPM Responder on a random port to set up the FPM connection activation. The FPM Responder SHOULD listen to a well-known port (This port number is introduced to be assigned by the IANA.). Second the FPM Responder responds with an ACK packet including some parameters based on the request. When the FPM Initiator receives the ACK, it will prepare for starting the measurement process.

After the measurement, FPM Initiator sends Connection Deactivation request packet called DEA to the FPM Responder. The FPM Responder sends DEA-ACK packets back to the FPM Initiator after it receives the DEA packets to stop the measurement.

During the measurement process, the FPM Initiator periodically generates Forward Monitor (FM) packets with the source and destination IP addresses, and other classification information (for example DSCP class) of the service packets, which are sent to the FPM Responder. The generation and transmission of FM packets can be periodical with a specific time interval, or a certain number of business packets should be sent between two contiguous FM packets. The FPM Responder receives the FM packets and sends Backward Reporting (BR) packets which are constructed according to the FM packets. The path performance such as the delay, ipdv (IP Packet Delay Variation) and loss rate etc. are calculates by the FPM Initiator according to the information in the BR packets.

The FM packets have the same source and destination IP addresses, even the same DSCP class in some cases with the business packets. So they are carried through the transport network just most like the business packets, and delay, ipdv(IP Packet Delay Variation) and packet loss encountered by them resembles the performance as seen by the packets of the actual business flows. The FM and BR packets used

in FPM are small enough to produce influence to actual service flow as little as possible.

Essential differences exist between FPM and IPPM WG-defined measurement, such as TWAMP and OWAMP. The solutions adopted in TWAMP and OWAMP are counting the information of measurement packets sent to network by Sender, and actively obtaining the measurement results. FPM is based on the running traffic of applications, and it collects the statistics of real business flow. Additional OAM packets are sent among business flow. Those OAM packets can be small and the inserted frequency can be lower. The OAM packets are used to carry flow/application statistics, which can be used to measure and estimate the business flow performance.

4.3. Logical Model

The role and definition of the logical entities and measurement packets in FPM are defined as follows.

FPM is an end-to-end measurement, so two logical entities are defined.

- o FPM Initiator: FPM Initiator serves as the sending endpoint, and charges for generating and sending the request to initiate a FPM connection. It could also send FM packets to collect measurement data and generate statistical report.
- o FPM Responder: FPM Responder serves as the data receiving endpoint, and charges for responding the request of initiating a link. It could also send back a BR packet to the sending endpoint once it receives the FM packet from the FPM Initiator.

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

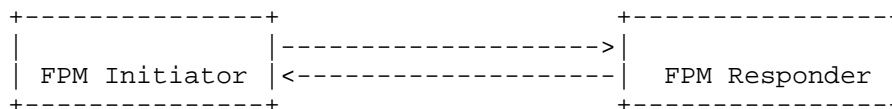


Figure 1: One possible relationship between FPM Initiator and FPM Responder

Note that the FPM Initiator can also serve as the data receiving endpoint, and the FPM Responder serves as the data sending endpoint. In this case the FM packets are sent by the FPM Responder and the BR packet is sent by the corresponding FPM Initiator. In later sections

the method is described for the first case. To avoid repetition, detail of this case is not described.

There are six types of packets in total, which include four types of control packets and two types of measurement packets.

Control packet:

- o ACT: It is sent from the FPM Initiator to a specific UDP port on FPM Responder, carries parameters used in negotiation process when initiating a FPM connection.
- o ACT-ACK: It is a response for ACT sent by the FPM Responder to the FPM Initiator.
- o DEA: It is sent by the FPM Initiator to the FPM Responder for disconnecting the FPM connection.
- o DEA-ACK: It is a response for DEA sent by the FPM Responder to the FPM Initiator.

Measurement packet:

- o FM (Forward Monitoring): It is sent by the FPM Initiator. The format of FM packet payload as defined by this document will be shown below.
- o BR (Backward Reporting): It is sent by the FPM Responder. The format of BR packet payload as defined by this document will be shown below. It is a response for FM.

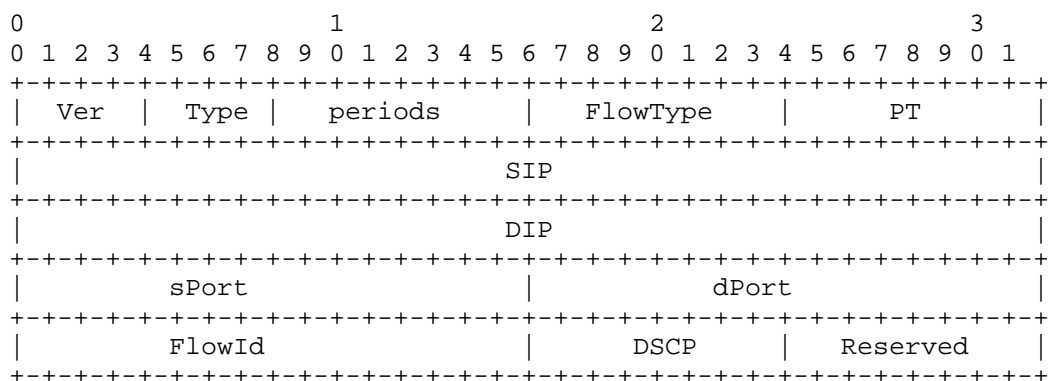
5. Connection Control

5.1. Connection Activation

In the FPM connection activation process, a Flow ID is assigned for a defined flow (the Flow ID is unique in a connection between the FPM Initiator and the FPM Responder). It should specify how to define the Flow corresponding to the measurement instance. Flow can be defined by different combinations of source IP address (SIP), destination IP address (DIP), protocol type (PT), DSCP, source port number (sPort) and destination port number (dPort). Three types of combinations are suggested: (SIP, DIP, PT), or (SIP, DIP, PT, DSCP) or (SIP, DIP, PT, sPort, dPort). The more the combinational dimensions are, the more fine-grained can be the monitoring of business flow.

Before starting the measurement, a connection should be established. When the FPM Initiator wants to start the measurement process, it enables the measurement capabilities to the FPM Responder by sending ACT packet to the specific UDP port on the FPM Responder. When the FPM Responder receives the ACT, it enables its measurement function and responses to the FPM Initiator with ACT-ACK packet. The connection activation process is finished after the FPM Initiator receiving the ACT-ACK packet from the FPM Responder, then the FPM Initiator can send FM packet after one cycle. The definition of flow, FlowID, and the sending period of FM packets must be consulted by two ends during the connection activation process.

The format of ACT packet is defined as follows:



Ver and Type existed in all packets in this memo indicate the version and type of packet. Type in these packets MUST be 0x1 indicates that this is an ACT packet.

Periods defined by FPM Initiator indicates the sending period of FM packets.

FlowType indicates how a flow is defined. 0x0 in this field is for (SIP, DIP, PT, sPort, dPort), while 0x1 is for (SIP, DIP, PT, DSCP) and 0x2 is for (SIP, DIP, PT). The other values are not defined.

PT is the protocol type value of the service flow needed to be measured. It may be UDP, TCP, SCTP or other types.

SIP is the source IP address of the service flow, and DIP is the destination IP address of the service flow. SPort and DPort which are valid only when the flow is defined by (SIP, DIP, PT, sPort, dPort) indicate source/destination port number of the ACT packets.

If the FlowType is not defined by (SIP, DIP, PT, sPort, dPort), this field is 0xFF.

FlowId is the flow id assigned for the defined flow. It is defined by the FPM Initiator. The FlowId field in others control packets and measurement packets have the same meaning.

DSCP is valid only when the flow is defined by (SIP, DIP, PT, DSCP). It indicates the value of the DSCP field in IP header of service flow.

Reserved is reserved for extensions in future and MUST be set to 0x0 currently.

The format of ACT-ACK packet is defined as follows:

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Ver										Type										periods										FlowId									
Accept										Reserved																													

Type of 0X2 indicates ACT-ACK.

Accept is 0x0 means Connection Activation is OK. Accept is 0x01 means Connection Activation is failure and the reason is unspecified.

5.2. Connection Deactivation

When the FPM Initiator wants to stop the measurement, it sends Connection Deactivation request packet called DEA to the FPM Responder. The FPM Responder sends DEA-ACK packets back to the FPM Initiator after it receives the DEA packets.

The format of DEA packet is defined as follows:

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Ver										Type										Reserved										FlowId									

Type of 0X5 indicates DEA.

The format of DEA-ACK packet is defined as follows:

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| Ver  | Type |  Reserved  |                               FlowId |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Type of 0X6 indicates DEA-ACK.

6. Measurement Process

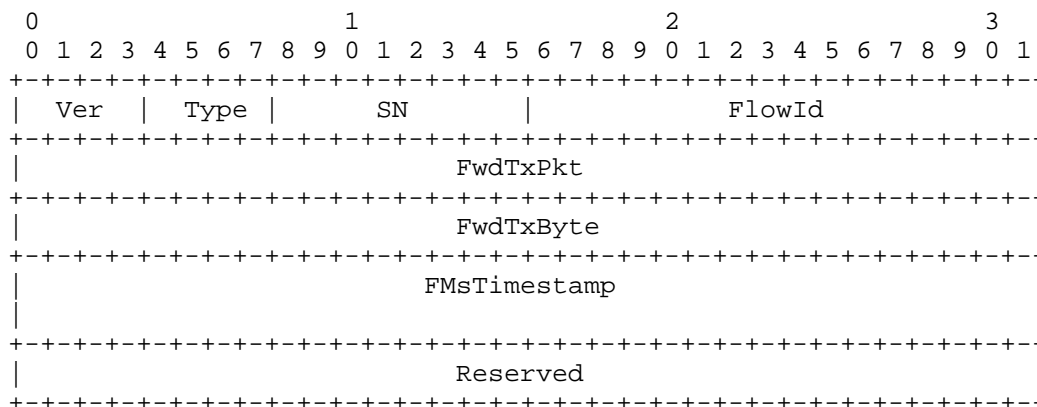
This section describes FPM Measurement process. It runs over UDP, and its packet header is constructed in accordance with the business packet except the source port number and destination port number. The destination port number is a well-known port number, and the source port number can be assigned a random port number. Its packets function is similar to the OAM packets, they can be small and the inserted frequency can be lower.

6.1. FPM Initiator behavior

In the measurement phase, FPM Initiator major responsibility is to structure and send FM measurement packet, as well as receive and process BR measurement packet.

When the connection is established successfully, the FPM Initiator sends FM packets according to the given time-interval. Regardless of any scheduling delays, each packet that is actually sent MUST have the best possible approximation of its real time of departure as its timestamp (in the packet).

The format of FM packet is defined as follows:



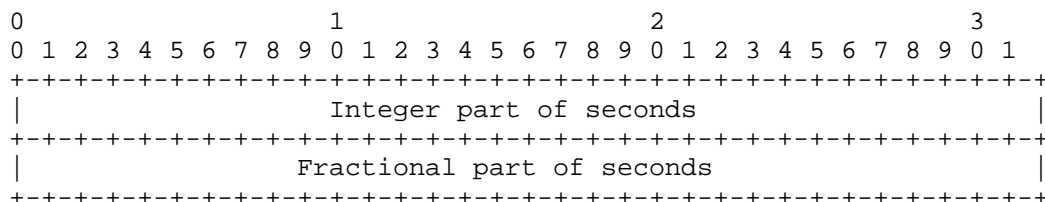
Type of 0X3 indicates FM.

SN is the sequence number of the flow, which distinguishes the different FM packets and indicates the correspondence between FM packets and BR packets. Each FPM flow SHOULD maintain a set of sequence numbers (SN).

FwdTxPkt is the accumulation of the number of the packets sent by the FPM Initiator. FwdTxByte is the accumulation of the number of bytes sent by the FPM Initiator. In order to determine the value of the fields of FwdTxPkt and FwdTxByte, the FPM Initiator maintains two counters, SPN and SBN, for each FPM flow that is incremented every time a business packet is sent. When the FM packets are to be sent, the FwdTxPkt and FwdTxByte are set to the then value of the counters respectively.

FMstimestamp is the timestamp when the FPM Initiator sends the first bit of the FM packets.

The format of the FMstimestamp is the same as in [RFC5905] and is as follows: the first 32 bits represent the unsigned integer number of seconds elapsed since 0h on 1 January 1900; the next 32 bits represent the fractional part of a second that has elapsed since then. The timestamp follows the above format in the below sections.

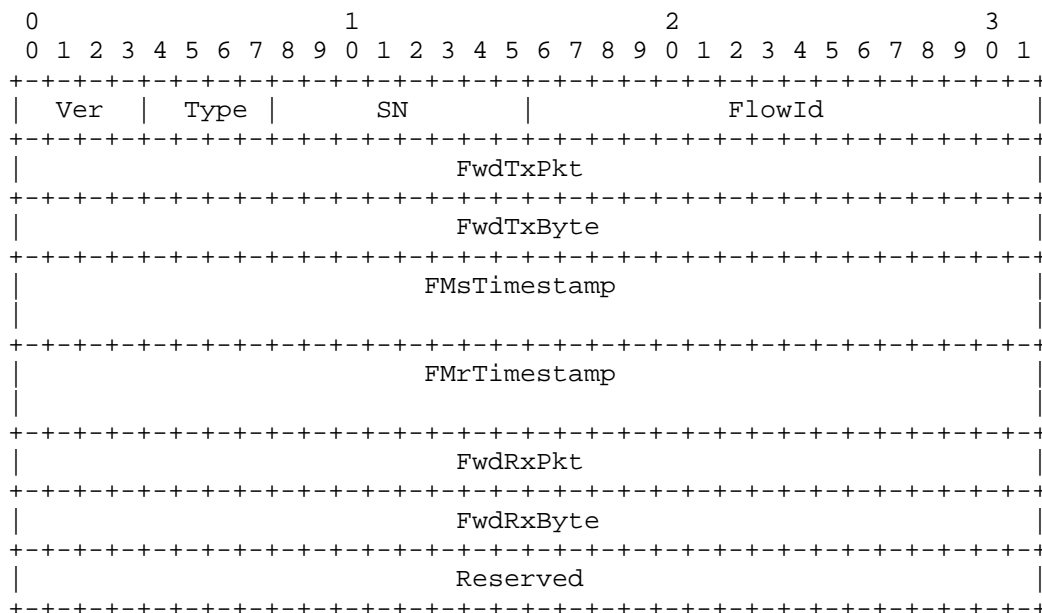


6.2. FPM Responder behavior

FPM requires the FPM Responder to transmit a packet to the FPM Initiator in response to each FM packet it receives.

When the FPM Responder receives a FM packet, it copies the value of FwdTxPkt, FwdTxByte and Timestamp in FM packet into the corresponding fields of the BR packet, and sets the fields of Sender Timestamp, FwdRxPkt and FwdRxByte and sends the BR packet.

The format of BR packet is defined as follows:



Type of 0x4 indicates BR.

SN is copied from the SN field of the corresponding FM packet.

The FwdTxPkt and FwdTxByte are copied from the corresponding FM packet.

FwdRxPkt is the accumulation of the number of the packets received by the FPM Responder.

FwdRxByte is the accumulation of the number of bytes received by the FPM Responder.

In order to determine the value of the fields of FwdRxPkt and FwdRxByte, the FPM Responder maintains two counters, RPN and RBN, for each FPM flow that is incremented every time a business packet is received. When the BR packets are to be sent, the FwdRxPkt and FwdRxByte are set to the then value of the counters respectively.

FMsTimestamp is copied from the Timestamp field of the FM packets; FMrTimestamp is the timestamp when the FPM Responder receives the last bit of the FM packets. The format of the timestamp is the same as in FM packets.

Note that the FPM Initiator could start multiple measurement engines; each engine is corresponding to an active logical path (with a different Flow). These measurement engines operate in parallel, and send FM packets with the flow id of the logical path, collect the corresponding BR packets, and maintain the collected statistical values.

7. Metrics

FPM method can measure most of the Metrics defined by IPPM WG. We assume that the reader is familiar with IPPM working group document, some terms in these documents may be used as described below.

[RFC2679] describes the one-way delay metrics between the IP hosts. Delay is dT means that Src sent the first bit of a Type-P packet to Dst at time T and that Dst received the last bit of that packet at time T+dT[RFC2679]. In FPM, FMsTimestamp in FM packet is the timestamp when the FPM Initiator sent the first bit of the FM packet, and FPM Responder records the timestamp when the FPM Responder received the last bit of the FM packets. Assuming that time in both ends are synchronous, one-way delay in one measurement can be derived through FMrTimestamp-FMsTimestamp.

[RFC2679]also describes a pseudo-random Poisson sampling method for sampling T between the designated T0 and Tf (T0 and Tf is the sampling time boundary value). FPM can sample the set of results measured in FPM Initiator, and the sampling method can be the pseudo-

random Poisson sampling or other methods. Error handling and other operations are described in [RFC2679].

[RFC2680] defines the one-way packet loss metrics between the IP hosts. If the packet can reach the destination host, the data loss value is 0. If the packet is lost during transmission, the data loss value is 1[RFC2680]. The host records whether the business packet is lost or not during the measurement cycle and it can calculate the unidirectional IP packet loss for this period of time. FPM uses this method to get statistics of the business packet loss. In the measurement process, FPM Initiator uses a counter to count the actual number of packets sent by it. FPM Responder records whether business packet has arrived or not, if the packet reaches the FPM Responder, business packet loss value is 0; if the packet is lost during transmission, the business packet loss value is 1. FPM Responder cumulates the loss value, and calculates the number of packets actually received. The above parameters, carried by measurement packets, are exchanged between both sides, used to calculate packet loss and loss rate in FM Initiator ultimately.

As described above, measurement packet only needs to carry statistics information of both sides in the FPM. The measurement packet size and the transmission frequency are relatively small, so FPM will not cause too much impact on the original business.

Similarly, the FPM methods can also measure metrics defined in [RFC3393] and [RFC4737].

Note that the statistics is carried in the IP layer. They are calculated before packet fragmentation at the FPM Initiator and after packet reassembly at the FPM Responder. If both the FPM Initiator and the FPM Responder support IPSec, the parameter statistics, including number of bytes/packets, Delay and ipdv (IP Packet Delay Variation), are carried before IPSec process is executed. If IPHC (IP Header Compression) is used at the two ends, the parameter statistics should be carried before IPHC.

7.1. Example of loss rate

Using the statistics of loss data, we can calculate the value of loss rate. The flowing is an example.

When the d th BR packet is received at the FPM Initiator, the loss rate plr based on the d th BR packet and $(d-1)$ th BR packet is calculated as:

$$\text{plrd} = \frac{(\text{SPN}(d) - \text{SPN}(d-1)) - (\text{RPN}(d) - \text{RPN}(d-1))}{\text{SPN}(d) - \text{SPN}(d-1)}$$

(SPN(d)-SPN(d-1)) indicates the number of service packets sent by the FPM Initiator during dth measurement, and (RPN(d)-RPN(d-1)) indicates the actual number of service packets received by the FPM Responder during dth measurement.

The loss rate needs to be aggregated over the reporting interval. Let's assume that N BR packets were received during the dth reporting interval. Therefore, the packet loss rate for that interval can be calculated as:

$$\text{PLRd} = \frac{1}{N} * \sum_{d=1}^N \text{plrd}$$

8. Exception Handling

8.1. FM/BR Packet Loss

In some cases the FM or BR packet may be lost in transit, then no statistics can be obtained from this round of measurement.

So the loss rate of the mth measurement can be calculated as:

$$\text{plrd} = \frac{(\text{SPN}(m) - \text{SPN}(n)) - (\text{RPN}(m) - \text{RPN}(n))}{\text{SPN}(m) - \text{SPN}(n)}$$

where m is the SN of the BR packet currently received and n is the SN of the latest BR received.

8.2. Packet Reordering

In the receive side if the received packets are out of order, the FM packet may arrive earlier than the last service packet sent before it, or later than the first service packet sent after it. Then statistical error of packet loss will be result in.

There are several reasons for packet reordering. When a network node

receives a fragmented IP packet, it has to reassemble the datagram and the extra time spent on the IP fragment reassembly may cause packet reordering; Some load sharing schemes for network (e.g. ECMP, ML-PPP) may create multipath for packets, which can also cause packet reordering; Multi-core CPU processing and multi-threading of packets in the sender and receiver may also lead to packet reordering.

In the simplest case that data transmits along a single path, DSCP can be used to classify the flow in order to avoid the packet reordering.

Note that the packet loss calculation is based on sample statistic, and by increasing the monitoring period, the error caused by the occasional packet reordering can be smoothed.

9. Use Case

This section describes a typical scene using the measurement method. The wireless mobile backhaul networks based on IP, share the available capacity between the connected eNodeBs. Compared to the traditional SDH/ATM transport network, in IP-RAN, the data transfer speed is unstable and data transfer lacks of QoS guarantee and there is no perfect testing method on packet loss, delay and ipdv (IP Packet Delay Variation). So it is necessary for the nodes in the RAN side to detect the network quality of the connection between RNC and NodeB or eNodeB and SAE.

Take the eNodeB and SAE for example; in order to make sure that the amount of generated traffic is aligned with the available capacity, it is important that the eNodeB probes the backhaul network to determine the actual delay, jitter and packet loss encountered by typical packets. The proposed method in this document can be used to detect the IP Performance of the connections between the eNB and S-GW.

As shown below, FPM Initiator is deployed in eNodeB, and FPM Responder is deployed in S-GW.

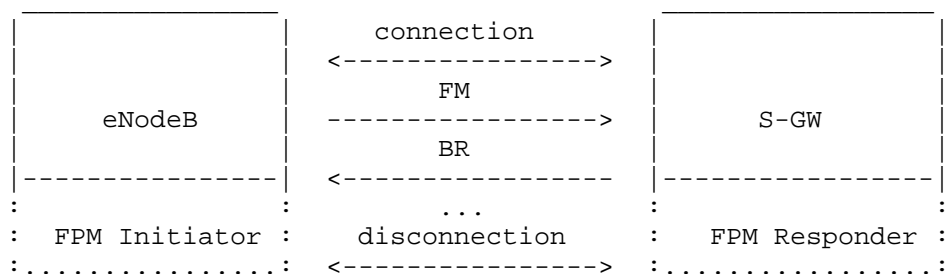


Figure 2: Example of FPM in backhaul network

At the eNodeB, FM packets are generated periodically with the source and destination IP addresses, and DSCP class. At the S-GW, after receiving the FM packets, BR packets are constructed. They are then forwarded back to the eNodeB.

We sent a similar packet of OAM, packet size and the transmission frequency is relatively small, so FPM will not cause too much impact on the original business between eNodeB and S-GW. Since the measurement packet is constructed according to the business packet, the network path of measurement packet and business packet is the same (Tunneling is used for the data transmission between eNodeB and S-GW, the parameter statistics of FPM should be carried before the tunnel. Measurement packet and business packet are encapsulated into a same tunnel and they are passed in the same path). The data obtained through FPM can represent the performance of the business flows between the eNodeB and the S-GW accurately.

Upon receiving the BR packets at the logical port the eNodeB exactly knows the current congestion extent in transport network. The bandwidth of the logical port is reduced if congestion is detected according to the measurement result; otherwise, the bandwidth is increased slowly.

10. Security Considerations

To be defined.

11. IANA Considerations

The destination port number of the newly defined packets for measurement needs to be assigned by the IANA.

12. Acknowledgments

The authors gratefully acknowledge reviews and contributions from Peter McCann.

The authors would like thank Xiangyang Gong and Xirong Que for their technical guidance towards to this draft.

The authors would like to thank Yuehui Ding for editing problem statement.

13. References

13.1. Normative Reference

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [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.
- [RFC5905] Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.

13.2. Informative References

- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, May 1998.
- [RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, 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.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J.

Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)",
RFC 5357, October 2008.

Authors' Addresses

Lishun Sun
Beijing University of Posts and Telecommunications
Xitucheng road 10
Haidian District, Beijing 100876
P. R. China

Email: lishunsun@Gmail.com

Fang Yu
Huawei Technologies
Huawei Building, Q20 No.156 Beiqing Rd.Z-park
Haidian District, Beijing 100095
P. R. China

Email: grace.yufang@huawei.com

Wendong Wang
Beijing University of Posts and Telecommunications
Xitucheng road 10
Haidian District, Beijing 100876
P. R. China

Email: wdwang@bupt.edu.cn

IP Performance Measurement (ippm)
Internet-Draft
Intended status: Informational
Expires: April 15, 2013

B. Trammell
ETH Zurich
October 12, 2012

Hybrid Measurement using IPPM Metrics
draft-trammell-ippm-hybrid-ps-00.txt

Abstract

Hybrid measurement is the combination of metrics derived from passive and active measurement to produce a measurement result. This document discusses use cases for hybrid measurement using metrics defined within the IPPM framework

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 15, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

1. Introduction

Hybrid measurement is the combination of metrics derived from passive and active measurement to produce a measurement result. This combination can be either spatial or temporal. For example, one way delay to a given endpoint could be derived from passive measurements from a sample of remote endpoints with which traffic is frequently exchanged, and supplemented with active measurements from endpoints with less frequent traffic, to build a "delay map" to a certain point in the network. On the temporal side, loss or delay metrics could be passively measured and stored over time to provide a baseline against which actively-measured loss or delay metrics could be compared during troubleshooting, in order to determine whether a specific path or path segment is contributing to an observed performance problem.

The IPPM working group has produced a framework [RFC2330] for and rich set of well-defined metrics (e.g. [RFC2679], [RFC2680]) for IP performance measurement using active methods, and protocols for measuring them. These metrics could form the basis of a platform for hybrid measurement, provided that passively-derived metrics were defined to be compatible with the corresponding actively-derived metrics; or alternately, provided that methodologies for passive measurement can be defined for each of the existing active metrics to be used, such that those methodologies produce values for the metrics equivalent to the active methodology for the same metric parameters, given some assumptions about the packet stream to be observed to perform the passive measurement, and given tolerances for uncertainty in the results.

2. Problem Statement

Complicating the definition of hybrid measurements is that passive measurement must make do with the traffic that is observable, while active measurement has some control over the traffic observed. Measurements for some set of parameters are not possible if no suitable traffic is observed, and the timing of the measurement cannot be controlled. Placement of the observation points for passive measurement along a path additionally introduces uncertainty in the results. For example, passive one-way delay measurement could be performed using two measurement points, one close to each endpoint, with synchronized clocks, comparing the observation times of packets via their hashes. This will not produce a value which is directly comparable to a Type-P-One-way-Delay measured as specified in section 3.6 of [RFC2679], because it will not account for the one-way-delay from the source to the source-side observation point, or from the destination-side observation point to the destination. Any specification of hybrid measurement using IPPM metrics must handle

these complications.

The proposed specification entails:

- o Definition of scenarios and requirements for hybrid measurement.
- o Selection of existing IPPM metrics to be used for the active side of hybrid measurements to meet these requirements.
- o Definition of equivalent passive measurement methodologies for each selected metric, specifically addressing the assumptions about the observed packet stream which must hold for the metric to be valid, and with a specific allowance for the measurement and/or estimation of uncertainty due to uncontrollable conditions or observation point placement.
- o Definition of metrics based on these passive methodologies, or modification of the definition of existing metrics to add passive methodologies.
- o Definition of methods for comparison between active and passive metrics allowing for estimated uncertainty.
- o Definition of methods for spatial and temporal composition of active and passive metrics together allowing for estimated uncertainty.

3. Scenarios and Requirements

[EDITOR'S NOTE: This section will contain scenarios and requirements for hybrid measurement. Candidate scenarios include (1) use of passive measurement to measure delay, loss, etc. on paths on which traffic is frequently sent, supplemented with active measurement on low-traffic paths or during low-traffic times and (2) use of passive measurement to establish a long-term baseline against which active measurements can be compared to detect and isolate anomalies, as in the introduction.]

4. Selected IPPM Metrics

[EDITOR'S NOTE: this section will contain information on the metrics selected for passive measurement, and initial discussion of passive measurement methodologies for them. Metric definition will presumably be left for a future document.]

5. Security Considerations

[EDITOR'S NOTE: this section will discuss general security considerations of using passive measurement for performance, both on the potential for attacks against the measurement system as well as the potential for privacy or security threats posed by the measurement system itself.]

6. IANA Considerations

This document contains no considerations for IANA.

7. References

7.1. Normative References

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

7.2. Informative References

[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.

Author's Address

Brian Trammell
Swiss Federal Institute of Technology Zurich
Gloriastrasse 35
8092 Zurich
Switzerland

Phone: +41 44 632 70 13
Email: trammell@tik.ee.ethz.ch

