A One-way Packet Loss Metric for IPPM

Status of this Memo

Copyright Notice

1. Introduction

   This memo defines a metric for one-way packet loss across Internet
   paths. It builds on notions introduced and discussed in the IPPM
   Framework document, RFC 2330 [1]; the reader is assumed to be
   familiar with that document.

   This memo is intended to be parallel in structure to a companion
   document for One-way Delay ("A One-way Delay Metric for IPPM") [2];
   the reader is assumed to be familiar with that 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 RFC 2119 [5].
   Although RFC 2119 was written with protocols in mind, the key words
   are used in this document for similar reasons. They are used to
   ensure the results of measurements from two different implementations
   are comparable, and to note instances when an implementation could
   perturb the network.

   The structure of the memo is as follows:

   + A 'singleton' analytic metric, called Type-P-One-way-Loss, is
      introduced to measure a single observation of packet transmission
      or loss.

+ Using this singleton metric, a 'sample', called Type-P-One-way-
   Loss-Poisson-Stream, is introduced to measure a sequence of
   singleton transmissions and/or losses measured at times taken from
   a Poisson process.

+ Using this sample, several 'statistics' of the sample are defined
   and discussed.

This progression from singleton to sample to statistics, with clear
separation among them, is important.

Whenever a technical term from the IPPM Framework document is first
used in this memo, it will be tagged with a trailing asterisk. For
example, "term*" indicates that "term" is defined in the Framework.

1.1. Motivation:

Understanding one-way packet loss of Type-P* packets from a source
host* to a destination host is useful for several reasons:

+ Some applications do not perform well (or at all) if end-to-end
   loss between hosts is large relative to some threshold value.

+ Excessive packet loss may make it difficult to support certain
   real-time applications (where the precise threshold of "excessive"
   depends on the application).

+ The larger the value of packet loss, the more difficult it is for
   transport-layer protocols to sustain high bandwidths.

+ The sensitivity of real-time applications and of transport-layer
   protocols to loss become especially important when very large
   delay-bandwidth products must be supported.

The measurement of one-way loss instead of round-trip loss is
motivated by the following factors:

+ In today's Internet, the path from a source to a destination may
   be different than the path from the destination back to the source
   ("asymmetric paths"), such that different sequences of routers are
   used for the forward and reverse paths. Therefore round-trip
   measurements actually measure the performance of two distinct
   paths together. Measuring each path independently highlights the
   performance difference between the two paths which may traverse
   different Internet service providers, and even radically different
   types of networks (for example, research versus commodity
   networks, or ATM versus packet-over-SONET).

   + Even when the two paths are symmetric, they may have radically
     different performance characteristics due to asymmetric queueing.

   + Performance of an application may depend mostly on the performance
     in one direction. For example, a file transfer using TCP may
     depend more on the performance in the direction that data flows,
     rather than the direction in which acknowledgements travel.

   + In quality-of-service (QoS) enabled networks, provisioning in one
     direction may be radically different than provisioning in the
     reverse direction, and thus the QoS guarantees differ. Measuring
     the paths independently allows the verification of both
     guarantees.

   It is outside the scope of this document to say precisely how loss
   metrics would be applied to specific problems.

1.2. General Issues Regarding Time

   {Comment: the terminology below differs from that defined by ITU-T
   documents (e.g., G.810, "Definitions and terminology for
   synchronization networks" and I.356, "B-ISDN ATM layer cell transfer
   performance"), but is consistent with the IPPM Framework document.
   In general, these differences derive from the different backgrounds;
   the ITU-T documents historically have a telephony origin, while the
   authors of this document (and the Framework) have a computer systems
   background. Although the terms defined below have no direct
   equivalent in the ITU-T definitions, after our definitions we will
   provide a rough mapping. However, note one potential confusion: our
   definition of "clock" is the computer operating systems definition
   denoting a time-of-day clock, while the ITU-T definition of clock
   denotes a frequency reference.}

   Whenever a time (i.e., a moment in history) is mentioned here, it is
   understood to be measured in seconds (and fractions) relative to UTC.

   As described more fully in the Framework document, there are four
   distinct, but related notions of clock uncertainty:

   synchronization*

       Synchronization measures the extent to which two clocks agree on
       what time it is. For example, the clock on one host might be
       5.4 msec ahead of the clock on a second host. {Comment: A rough
       ITU-T equivalent is "time error".}

accuracy*

>   Accuracy measures the extent to which a given clock agrees with
>   UTC. For example, the clock on a host might be 27.1 msec behind
>   UTC. {Comment: A rough ITU-T equivalent is "time error from
>   UTC".}

resolution*

>   Resolution measures the precision of a given clock. For
>   example, the clock on an old Unix host might advance only once
>   every 10 msec, and thus have a resolution of only 10 msec.
>   {Comment: A very rough ITU-T equivalent is "sampling period".}

skew*

>   Skew measures the change of accuracy, or of synchronization,
>   with time. For example, the clock on a given host might gain
>   1.3 msec per hour and thus be 27.1 msec behind UTC at one time
>   and only 25.8 msec an hour later. In this case, we say that the
>   clock of the given host has a skew of 1.3 msec per hour relative
>   to UTC, which threatens accuracy. We might also speak of the
>   skew of one clock relative to another clock, which threatens
>   synchronization. {Comment: A rough ITU-T equivalent is "time
>   drift".}

2. A Singleton Definition for One-way Packet Loss

2.1. Metric Name:

Type-P-One-way-Packet-Loss

2.2. Metric Parameters:

+ Src, the IP address of a host

+ Dst, the IP address of a host

+ T, a time

2.3. Metric Units:

The value of a Type-P-One-way-Packet-Loss is either a zero
(signifying successful transmission of the packet) or a one
(signifying loss).

2.4. Definition:

   >>The *Type-P-One-way-Packet-Loss* from Src to Dst at T is 0<< means
   that Src sent the first bit of a Type-P packet to Dst at wire-time* T
   and that Dst received that packet.

   >>The *Type-P-One-way-Packet-Loss* from Src to Dst at T is 1<< means
   that Src sent the first bit of a type-P packet to Dst at wire-time T
   and that Dst did not receive that packet.

2.5. Discussion:

   Thus, Type-P-One-way-Packet-Loss is 0 exactly when Type-P-One-way-
   Delay is a finite value, and it is 1 exactly when Type-P-One-way-
   Delay is undefined.

   The following issues are likely to come up in practice:

   + A given methodology will have to include a way to distinguish
      between a packet loss and a very large (but finite) delay. As
      noted by Mahdavi and Paxson [3], simple upper bounds (such as the
      255 seconds theoretical upper bound on the lifetimes of IP
      packets [4]) could be used, but good engineering, including an
      understanding of packet lifetimes, will be needed in practice.
      {Comment: Note that, for many applications of these metrics, there
      may be no harm in treating a large delay as packet loss. An audio
      playback packet, for example, that arrives only after the playback
      point may as well have been lost.}

   + If the packet arrives, but is corrupted, then it is counted as
      lost. {Comment: one is tempted to count the packet as received
      since corruption and packet loss are related but distinct
      phenomena. If the IP header is corrupted, however, one cannot be
      sure about the source or destination IP addresses and is thus on
      shaky grounds about knowing that the corrupted received packet
      corresponds to a given sent test packet. Similarly, if other
      parts of the packet needed by the methodology to know that the
      corrupted received packet corresponds to a given sent test packet,
      then such a packet would have to be counted as lost. Counting
      these packets as lost but packet with corruption in other parts of
      the packet as not lost would be inconsistent.}

   + If the packet is duplicated along the path (or paths) so that
      multiple non-corrupt copies arrive at the destination, then the
      packet is counted as received.

   + If the packet is fragmented and if, for whatever reason,
      reassembly does not occur, then the packet will be deemed lost.

2.6. Methodologies:

   As with other Type-P-* metrics, the detailed methodology will depend
   on the Type-P (e.g., protocol number, UDP/TCP port number, size,
   precedence).

   Generally, for a given Type-P, one possible methodology would proceed
   as follows:

   + Arrange that Src and Dst have clocks that are synchronized with
      each other. The degree of synchronization is a parameter of the
      methodology, and depends on the threshold used to determine loss
      (see below).

   + At the Src host, select Src and Dst IP addresses, and form a test
      packet of Type-P with these addresses.

   + At the Dst host, arrange to receive the packet.

   + At the Src host, place a timestamp in the prepared Type-P packet,
      and send it towards Dst.

   + If the packet arrives within a reasonable period of time, the one-
      way packet-loss is taken to be zero.

   + If the packet fails to arrive within a reasonable period of time,
      the one-way packet-loss is taken to be one. Note that the
      threshold of "reasonable" here is a parameter of the methodology.

      {Comment: The definition of reasonable is intentionally vague, and
      is intended to indicate a value "Th" so large that any value in
      the closed interval [Th-delta, Th+delta] is an equivalent
      threshold for loss. Here, delta encompasses all error in clock
      synchronization along the measured path. If there is a single
      value after which the packet must be counted as lost, then we
      reintroduce the need for a degree of clock synchronization similar
      to that needed for one-way delay. Therefore, if a measure of
      packet loss parameterized by a specific non-huge "reasonable"
      time-out value is needed, one can always measure one-way delay and
      see what percentage of packets from a given stream exceed a given
      time-out value.}

   Issues such as the packet format, the means by which Dst knows when
   to expect the test packet, and the means by which Src and Dst are
   synchronized are outside the scope of this document. {Comment: We
   plan to document elsewhere our own work in describing such more
   detailed implementation techniques and we encourage others to as
   well.}

2.7. Errors and Uncertainties:

   The description of any specific measurement method should include an
   accounting and analysis of various sources of error or uncertainty.
   The Framework document provides general guidance on this point.

   For loss, there are three sources of error:

   + Synchronization between clocks on Src and Dst.

   + The packet-loss threshold (which is related to the synchronization
      between clocks).

   + Resource limits in the network interface or software on the
      receiving instrument.

   The first two sources are interrelated and could result in a test
   packet with finite delay being reported as lost. Type-P-One-way-
   Packet-Loss is 0 if the test packet does not arrive, or if it does
   arrive and the difference between Src timestamp and Dst timestamp is
   greater than the "reasonable period of time", or loss threshold. If
   the clocks are not sufficiently synchronized, the loss threshold may
   not be "reasonable" - the packet may take much less time to arrive
   than its Src timestamp indicates. Similarly, if the loss threshold
   is set too low, then many packets may be counted as lost. The loss
   threshold must be high enough, and the clocks synchronized well
   enough so that a packet that arrives is rarely counted as lost. (See
   the discussions in the previous two sections.)

   Since the sensitivity of packet loss measurement to lack of clock
   synchronization is less than for delay, we refer the reader to the
   treatment of synchronization errors in the One-way Delay metric [2]
   for more details.

   The last source of error, resource limits, cause the packet to be
   dropped by the measurement instrument, and counted as lost when in
   fact the network delivered the packet in reasonable time.

   The measurement instruments should be calibrated such that the loss
   threshold is reasonable for application of the metrics and the clocks
   are synchronized enough so the loss threshold remains reasonable.

   In addition, the instruments should be checked to ensure the that the
   possibility a packet arrives at the network interface, but is lost
   due to congestion on the interface or to other resource exhaustion
   (e.g., buffers) on the instrument is low.

2.8. Reporting the metric:

The calibration and context in which the metric is measured MUST be
carefully considered, and SHOULD always be reported along with metric
results. We now present four items to consider: Type-P of the test
packets, the loss threshold, instrument calibration, and the path
traversed by the test packets. This list is not exhaustive; any
additional information that could be useful in interpreting
applications of the metrics should also be reported.

2.8.1. Type-P

As noted in the Framework document [1], the value of the metric may
depend on the type of IP packets used to make the measurement, or
"Type-P". The value of Type-P-One-way-Delay could change if the
protocol (UDP or TCP), port number, size, or arrangement for special
treatment (e.g., IP precedence or RSVP) changes. The exact Type-P
used to make the measurements MUST be accurately reported.

2.8.2. Loss threshold

The threshold (or methodology to distinguish) between a large finite
delay and loss MUST be reported.

2.8.3. Calibration results

The degree of synchronization between the Src and Dst clocks MUST be
reported. If possible, possibility that a test packet that arrives
at the Dst network interface is reported as lost due to resource
exhaustion on Dst SHOULD be reported.

2.8.4. Path

Finally, the path traversed by the packet SHOULD be reported, if
possible. In general it is impractical to know the precise path a
given packet takes through the network. The precise path may be
known for certain Type-P on short or stable paths. If Type-P
includes the record route (or loose-source route) option in the IP
header, and the path is short enough, and all routers* on the path
support record (or loose-source) route, then the path will be
precisely recorded. This is impractical because the route must be
short enough, many routers do not support (or are not configured for)
record route, and use of this feature would often artificially worsen
the performance observed by removing the packet from common-case
processing. However, partial information is still valuable context.
For example, if a host can choose between two links* (and hence two
separate routes from Src to Dst), then the initial link used is
valuable context. {Comment: For example, with Merit's NetNow setup,

a Src on one NAP can reach a Dst on another NAP by either of several
different backbone networks.}

3. A Definition for Samples of One-way Packet Loss

Given the singleton metric Type-P-One-way-Packet-Loss, we now define
one particular sample of such singletons. The idea of the sample is
to select a particular binding of the parameters Src, Dst, and Type-
P, then define a sample of values of parameter T. The means for
defining the values of T is to select a beginning time T0, a final
time Tf, and an average rate lambda, then define a pseudo-random
Poisson process of rate lambda, whose values fall between T0 and Tf.
The time interval between successive values of T will then average
1/lambda.

{Comment: Note that Poisson sampling is only one way of defining a
sample. Poisson has the advantage of limiting bias, but other
methods of sampling might be appropriate for different situations.
We encourage others who find such appropriate cases to use this
general framework and submit their sampling method for
standardization.}

3.1. Metric Name:

Type-P-One-way-Packet-Loss-Poisson-Stream

3.2. Metric Parameters:

+ Src, the IP address of a host

+ Dst, the IP address of a host

+ T0, a time

+ Tf, a time

+ lambda, a rate in reciprocal seconds

3.3. Metric Units:

A sequence of pairs; the elements of each pair are:

+ T, a time, and

+ L, either a zero or a one

The values of T in the sequence are monotonic increasing. Note that
T would be a valid parameter to Type-P-One-way-Packet-Loss, and that
L would be a valid value of Type-P-One-way-Packet-Loss.

3.4. Definition:

Given T0, Tf, and lambda, we compute a pseudo-random Poisson process
beginning at or before T0, with average arrival rate lambda, and
ending at or after Tf. Those time values greater than or equal to T0
and less than or equal to Tf are then selected. At each of the times
in this process, we obtain the value of Type-P-One-way-Packet-Loss at
this time. The value of the sample is the sequence made up of the
resulting <time, loss> pairs. If there are no such pairs, the
sequence is of length zero and the sample is said to be empty.

3.5. Discussion:

The reader should be familiar with the in-depth discussion of Poisson
sampling in the Framework document [1], which includes methods to
compute and verify the pseudo-random Poisson process.

We specifically do not constrain the value of lambda, except to note
the extremes. If the rate is too large, then the measurement traffic
will perturb the network, and itself cause congestion. If the rate
is too small, then you might not capture interesting network
behavior. {Comment: We expect to document our experiences with, and
suggestions for, lambda elsewhere, culminating in a "best current
practices" document.}

Since a pseudo-random number sequence is employed, the sequence of
times, and hence the value of the sample, is not fully specified.
Pseudo-random number generators of good quality will be needed to
achieve the desired qualities.

The sample is defined in terms of a Poisson process both to avoid the
effects of self-synchronization and also capture a sample that is
statistically as unbiased as possible. The Poisson process is used
to schedule the delay measurements. The test packets will generally
not arrive at Dst according to a Poisson distribution, since they are
influenced by the network.

{Comment: there is, of course, no claim that real Internet traffic
arrives according to a Poisson arrival process.

It is important to note that, in contrast to this metric, loss rates
observed by transport connections do not reflect unbiased samples.
For example, TCP transmissions both (1) occur in bursts, which can

induce loss due to the burst volume that would not otherwise have
been observed, and (2) adapt their transmission rate in an attempt to
minimize the loss rate observed by the connection.}

All the singleton Type-P-One-way-Packet-Loss metrics in the sequence
will have the same values of Src, Dst, and Type-P.

Note also that, given one sample that runs from T0 to Tf, and given
new time values T0' and Tf' such that T0 <= T0' <= Tf' <= Tf, the
subsequence of the given sample whose time values fall between T0'
and Tf' are also a valid Type-P-One-way-Packet-Loss-Poisson-Stream
sample.

3.6. Methodologies:

The methodologies follow directly from:

+   the selection of specific times, using the specified Poisson
    arrival process, and

+   the methodologies discussion already given for the singleton Type-
    P-One-way-Packet-Loss metric.

Care must be given to correctly handle out-of-order arrival of test
packets; it is possible that the Src could send one test packet at
TS[i], then send a second one (later) at TS[i+1], while the Dst could
receive the second test packet at TR[i+1], and then receive the first
one (later) at TR[i].

3.7. Errors and Uncertainties:

In addition to sources of errors and uncertainties associated with
methods employed to measure the singleton values that make up the
sample, care must be given to analyze the accuracy of the Poisson
arrival process of the wire-times of the sending of the test packets.
Problems with this process could be caused by several things,
including problems with the pseudo-random number techniques used to
generate the Poisson arrival process. The Framework document shows
how to use the Anderson-Darling test verify the accuracy of the
Poisson process over small time frames. {Comment: The goal is to
ensure that the test packets are sent "close enough" to a Poisson
schedule, and avoid periodic behavior.}

3.8. Reporting the metric:

The calibration and context for the underlying singletons MUST be
reported along with the stream. (See "Reporting the metric" for
Type-P-One-way-Packet-Loss.)

4. Some Statistics Definitions for One-way Packet Loss

   Given the sample metric Type-P-One-way-Packet-Loss-Poisson-Stream, we
   now offer several statistics of that sample. These statistics are
   offered mostly to be illustrative of what could be done.

4.1. Type-P-One-way-Packet-Loss-Average

   Given a Type-P-One-way-Packet-Loss-Poisson-Stream, the average of all
   the L values in the Stream. In addition, the Type-P-One-way-Packet-
   Loss-Average is undefined if the sample is empty.

   Example: suppose we take a sample and the results are:

      Stream1 = <
      <T1, 0>
      <T2, 0>
      <T3, 1>
      <T4, 0>
      <T5, 0>
      >

   Then the average would be 0.2.

   Note that, since healthy Internet paths should be operating at loss
   rates below 1% (particularly if high delay-bandwidth products are to
   be sustained), the sample sizes needed might be larger than one would
   like. Thus, for example, if one wants to discriminate between
   various fractions of 1% over one-minute periods, then several hundred
   samples per minute might be needed. This would result in larger
   values of lambda than one would ordinarily want.

   Note that although the loss threshold should be set such that any
   errors in loss are not significant, if the possibility that a packet
   which arrived is counted as lost due to resource exhaustion is
   significant compared to the loss rate of interest, Type-P-One-way-
   Packet-Loss-Average will be meaningless.

5. Security Considerations

   Conducting Internet measurements raises both security and privacy
   concerns. This memo does not specify an implementation of the
   metrics, so it does not directly affect the security of the Internet
   nor of applications which run on the Internet. However,
   implementations of these metrics must be mindful of security and
   privacy concerns.

There are two types of security concerns: potential harm caused by
the measurements, and potential harm to the measurements. The
measurements could cause harm because they are active, and inject
packets into the network. The measurement parameters MUST be
carefully selected so that the measurements inject trivial amounts of
additional traffic into the networks they measure. If they inject
"too much" traffic, they can skew the results of the measurement, and
in extreme cases cause congestion and denial of service.

The measurements themselves could be harmed by routers giving
measurement traffic a different priority than "normal" traffic, or by
an attacker injecting artificial measurement traffic. If routers can
recognize measurement traffic and treat it separately, the
measurements will not reflect actual user traffic. If an attacker
injects artificial traffic that is accepted as legitimate, the loss
rate will be artificially lowered. Therefore, the measurement
methodologies SHOULD include appropriate techniques to reduce the
probability measurement traffic can be distinguished from "normal"
traffic. Authentication techniques, such as digital signatures, may
be used where appropriate to guard against injected traffic attacks.

The privacy concerns of network measurement are limited by the active
measurements described in this memo. Unlike passive measurements,
there can be no release of existing user data.

6. Acknowledgements

Thanks are due to Matt Mathis for encouraging this work and for
calling attention on so many occasions to the significance of packet
loss.

Thanks are due also to Vern Paxson for his valuable comments on early
drafts, and to Garry Couch and Will Leland for several useful
suggestions.

7. References

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

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

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

   [4] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

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

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

8. Authors' Addresses

   Guy Almes
   Advanced Network & Services, Inc.
   200 Business Park Drive
   Armonk, NY 10504
   USA

   Phone: +1 914 765 1120
   EMail: almes@advanced.org

   Sunil Kalidindi
   Advanced Network & Services, Inc.
   200 Business Park Drive
   Armonk, NY 10504
   USA

   Phone: +1 914 765 1128
   EMail: kalidindi@advanced.org

   Matthew J. Zekauskas
   Advanced Network & Services, Inc.
   200 Business Park Drive
   Armonk, NY 10504
   USA

   Phone: +1 914 765 1112
   EMail: matt@advanced.org

9. Full Copyright Statement

Acknowledgement