Network Working Group G. Almes, Advanced Network & Services
Internet Draft S. Kalidindi, Advanced Network & Services
Expiration Date: April 1998 November 1997

A One-way Delay Metric for IPPM
<draft-ietf-ippm-delay-01.txt>

1. Status of this Memo

2. Introduction

This memo defines a metric for one-way delay of packets across Internet paths. It builds on notions introduced and discussed in the IPPM Framework document (currently ''Framework for IP Performance Metrics'' <draft-ietf-ippm-framework-01.txt>); the reader is assumed to be familiar with that document.

This memo is intended to be very parallel in structure to a companion document for Packet Loss (''A Packet Loss Metric for IPPM'' <draft-ietf-ippm-loss-01.txt>).

The structure of the memo is as follows:

+ A 'singleton' analytic metric, called Type-P-One-way-Delay, will be introduced to measure a single observation of one-way delay.
+ Using this singleton metric, a 'sample', called Type-P-One-way-Delay-Stream, will be introduced to measure a sequence of singleton delays measured at times taken from a Poisson process.
+ Using this sample, several 'statistics' of the sample will be 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, as with >>term*<<.

2.1. Motivation:

One-way delay of a type-P packet 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 delay between hosts is large relative to some threshold value.
+ Erratic variation in delay makes it difficult (or impossible) to support many real-time applications.
+ The larger the value of delay, the more difficult it is for transport-layer protocols to sustain high bandwidths.
+ The minimum value of this metric provides an indication of the delay due only to propagation and transmission delay.
+ The minimum value of this metric provides an indication of the delay that will likely be experienced when the path* traversed is lightly loaded.
+ Values of this metric above the minimum provide an indication of the congestion present in the path.

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

2.2. General Issues Regarding Time

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
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.

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.

resolution
measures the precision of a given clock. For example, the clock on an old Unix host might tick only once every 10 msec, and thus have a resolution of only 10 msec.

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, and this threatens accuracy. We might also speak of the skew of one clock relative to another clock, and this threatens synchronization.

3. A Singleton Definition for One-way Delay

3.1. Metric Name:

Type-P-One-way-Delay

3.2. Metric Parameters:

+ Src, the IP address of a host
+ Dst, the IP address of a host
+ T, a time
+ Path, the path* from Src to Dst; in cases where there is only one path from Src to Dst, this optional parameter can be omitted

{Comment: the presence of path is motivated by cases such as with Merit's NetNow setup, in which a Src on one NAP can reach a Dst on another NAP by either of several different backbone networks. Generally, this optional parameter is useful only when several different routes are possible from Src to Dst. Using the loose source route IP option is avoided since it would often artificially worsen the performance observed, and since it might not be supported along some paths.}

3.3. Metric Units:

The value of a type-P-One-way-Delay is either a non-negative real number or an undefined (informally, infinite) number of seconds.

3.4. Definition:

For a non-negative real number dT, >>the *Type-P-One-way-Delay* from Src to Dst at T [via path] is dT<< means that Src sent the first bit of a type-P packet [via path] to Dst at wire-time T and that Dst received the last bit of that packet at wire-time T+dT.

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

3.5. Discussion:

Type-P-One-way-Delay is a relatively simple analytic metric, and one that we believe will afford effective methods of measurement.

The following issues are likely to come up in practice:

+ Since delay values will often be as low as the 100 usec to 10 msec range, it will be important for Src and Dst to synchronize very closely. GPS systems afford one way to achieve synchronization to within several 10s of usec. Ordinary application of NTP may allow synchronization to within several msec, but this depends on the stability and symmetry of delay properties among those NTP agents used, and this delay is what we are trying to measure. A combination of some GPS-based NTP servers and a conservatively designed and deployed set of other NTP servers should yield good results, but this is yet to be tested.

+ A given methodology will have to include a way to determine whether a delay value is infinite or whether it is merely very large (and the packet is yet to arrive at Dst). As noted by Mahdavi and Paxson, simple upper bounds (such as the 255 seconds theoretical upper bound on the lifetimes of IP packets [Postel: RFC 791]) 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, the harm in treating a large delay as infinite might be zero or very small. A TCP data packet, for example, that arrives only after several multiples of the RTT may as well have been lost.}
+ As with other 'type-P' metrics, the value of the metric may depend on such properties of the packet as protocol, (UDP or TCP) port number, size, and arrangement for special treatment (as with IP precedence or with RSVP).

+ 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, and the first copy to arrive determines the packet's one-way delay.
+ If the packet is fragmented and if, for whatever reason, reassembly does not occur, then the packet will be deemed lost.

3.6. 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, the methodology would proceed as follows:
+ Arrange that Src and Dst are synchronized; that is, that they have clocks that are very closely synchronized with each other and each fairly close to the actual time.
+ At the Src host, select Src and Dst IP addresses, and form a test packet of Type-P with these addresses. Any 'padding' portion of the packet needed only to make the test packet a given size should

be filled with randomized bits to avoid a situation in which the measured delay is lower than it would otherwise be due to compression techniques along the path.

+ Optionally, select a specific path and arrange for Src to send the packet to that path. {Comment: This could be done, for example, by installing a temporary host-route for Dst in Src's routing table.}
+ 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 [via path].
+ If the packet arrives within a reasonable period of time, take a timestamp as soon as possible upon the receipt of the packet. By subtracting the two timestamps, an estimate of one-way delay can be computed. Error analysis of a given implementation of the method must take into account the closeness of synchronization between Src and Dst. If the delay between Src's timestamp and the actual sending of the packet is known, then the estimate could be adjusted by subtracting this amount; uncertainty in this value must be taken into account in error analysis. Similarly, if the delay between the actual receipt of the packet and Dst's timestamp is known, then the estimate could be adjusted by subtracting this amount; uncertainty in this value must be taken into account in error analysis.

+ If the packet fails to arrive within a reasonable period of time, the one-way delay is taken to be undefined (informally, infinite). Note that the threshold of 'reasonable' here is a parameter of the methodology.

Issues such as the packet format, the means by which the path is ensured, 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.}

3.7. Errors and Uncertainties:

The description of any specific measurement method should include an accounting and analysis of various sources of error/uncertainty. The Framework document provides general guidence on this point, but we note here the following specifics related to delay metrics:

+ Errors/uncertainties due to uncertainties in the clocks of the Src and Dst hosts.
+ Errors/uncertainties due to the difference between 'wire time' and 'host time'.

Each of these are discussed in more detail below.

3.7.1. Errors/uncertainties related to Clocks

The uncertainty in a measurement of one-way delay is related, in part, to uncertainties in the clocks of the Src and Dst hosts. In the following, we refer to the clock used to measure when the packet was sent from Src as the source clock, we refer to the clock used to measure when the packet was received by Dst as the dest clock, we refer to the observed time when the packet was sent by the source clock as Tsource, and the observed time when the packet was received by the dest clock as Tdest. Alluding to the notions of synchronization, accuracy, resolution, and skew mentioned in the Introduction, we note the following:

+ Any error in the synchronization between the source clock and the dest clock will contribute to error in the delay measurement. We say that the source clock and the dest clock have a synchronization error of Tsynch if the source clock is Tsynch ahead of the dest clock. Thus, if we know the value of Tsynch exactly, we could correct for clock synchronization by adding Tsynch to the uncorrected value of Tdest-Tsource.

+ The accuracy of a clock is important only in identifying the time at which a given delay was measured. Accuracy, per se, has no importance to the accuracy of the measurement of delay. This is because, when computing delays, we are interested only in the differences between clock values.
+ The resolution of a clock adds to uncertainty about any time measured with it. Thus, if the source clock has a resolution of 10 msec, then this adds 10 msec of uncertainty to any time value mea-

sured with it. We will denote the resolution of the source clock and the dest clock as Rsource and Rdest, respectively.

+ The skew of a clock is not so much an additional issue as it is a realization of the fact that Tsynch is itself a function of time. Thus, if we attempt to measure or to bound Tsynch, this needs to be done periodically. Over some periods of time, this function can be approximated as a linear function plus some higher order terms; in these cases, one option is to use knowledge of the linear component to correct the clock. Using this correction, the residual Tsynch is made smaller, but remains a source of uncertainty that must be accounted for. We use the function Esynch(t) to denote an upper bound on the uncertainty in synchronization. Thus, |Tsynch(t)| <= Esynch(t).

Taking these items together, we note that naive computation Tdest-Tsource will be off by Tsynch(t) +/- (|Rsource|+|Rdest|). Using the notion of Esynch(t), we note that these clock-related problems introduce a total uncertainty of Esynch(t)+|Rsource|+|Rdest|. This estimate of total clock-related uncertainty should be included in the error/uncertainty analysis of any measurement implementation.

3.7.2. Errors/uncertainties related to Wire-time vs Host-time

As we've defined one-way delay, we'd like to measure the time between when the test packet leaves the network interface of Src and when it (completely) arrives at the network interface of Dst, and we refer to this as 'wire time'. If the timings are themselves performed by software on Src and Dst, however, then this software can only directly measure the time between when Src grabs a timestamp just prior to sending the test packet and when Dst grabs a timestamp just after having received the test packet, and we refer to this as 'host time'.

To the extent that the difference between wire time and host time is accurately known, this knowledge can be used to correct for host time measurements and the corrected value more accurately estimates the desired (wire time) metric.

To the extent, however, that the difference between wire time and host time is uncertain, this uncertainty must be accounted for in an

analysis of a given measurement method. We denote by Hsource an upper bound on the uncertainty in the difference between wire time and host time on the Src host, and similarly define Hdest for the Dst host. We then note that these problems introduce a total uncertainty of Hsource+Hdest. This estimate of total wire-vs-host uncertainty should be included in the error/uncertainty analysis of any measurement implementation.

4. A Definition for Samples of One-way Delay

Given the singleton metric Type-P-One-way-Delay, 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, path, 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 arrival process of rate lambda, whose values fall between T0 and Tf. The time interval between successive values of T will then average 1/lambda.

4.1. Metric Name:

Type-P-One-way-Delay-Stream

4.2. Metric Parameters:

+ Src, the IP address of a host
+ Dst, the IP address of a host
+ Path, the path* from Src to Dst; in cases where there is only one path from Src to Dst, this optional parameter can be omitted
+ T0, a time
+ Tf, a time
+ lambda, a rate in reciprocal seconds

4.3. Metric Units:

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

+ T, a time, and
+ dT, either a non-negative real number or an undefined number of seconds.

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

4.4. 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-Delay at this time. The value of the sample is the sequence made up of the resulting <time, delay> pairs. If there are no such pairs, the sequence is of length zero and the sample is said to be empty.

4.5. Discussion:

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

All the singleton Type-P-One-way-Delay metrics in the sequence will have the same values of Src, Dst, [path,] 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-Delay-Stream sample.

4.6. 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-Delay metric.

Care must, of course, 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].

4.7. 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-time of the sending of the test packets. Problems with this process could be caused by either of several things, including problems with the pseudo-random number techniques used to generate the Poisson arrival process, or with jitter in the value of Hsource (mentioned above as uncertainty in the singleton delay metric). The Framework document shows how to use an Anderson-Darling test for this.

5. Some Statistics Definitions for One-way Delay

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

5.1. Type-P-One-way-Delay-Percentile

Given a Type-P-One-way-Delay-Stream and a percent X between 0% and 100%, the Xth percentile of all the dT values in the Stream. In computing this percentile, undefined values are treated as infinitely large. Note that this means that the percentile could thus be undefined (informally, infinite). In addition, the Type-P-One-way-Delay-Percentile is undefined if the sample is empty.

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

```
     Stream1 = <
     <T1, 100 msec>
     <T2, 110 msec>
     <T3, undefined>
     <T4, 90 msec>
     <T5, 500 msec>
     >
```

Then the 50th percentile would be 110 msec, since 90 msec and 100 msec are smaller and 110 msec and 'undefined' are larger.

5.2. Type-P-One-way-Delay-Median

Given a Type-P-One-way-Delay-Stream, the median of all the dT values in the Stream. In computing the median, undefined values are treated as infinitely large.

As noted in the Framework document, the median differs from the 50th percentile only when the sample contains an even number of values, in which case the mean of the two central values is used.

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

    Stream2 = <
    <T1, 100 msec>
    <T2, 110 msec>
    <T3, undefined>
    <T4, 90 msec>
    >

Then the median would be 105 msec, the mean of 100 msec and 110 msec, the two central values.

5.3. Type-P-One-way-Delay-Minumum

Given a Type-P-One-way-Delay-Stream, the minimum of all the dT values in the Stream. In computing this, undefined values are treated as infinitely large. Note that this means that the minimum could thus be undefined (informally, infinite) if all the dT values are undefined. In addition, the Type-P-One-way-Delay-Minimum is undefined if the sample is empty.

In the above example, the minimum would be 90 msec.

5.4. Type-P-One-way-Delay-Inverse-Percentile

Given a Type-P-One-way-Delay-Stream and a non-negative time duration threshold, the fraction of all the dT values in the Stream less than or equal to the threshold. The result could be as low as 0% (if all the dT values exceed threshold) or as high as 100%.

In the above example, the Inverse-Percentile of 103 msec would be 50%.

6. Security Considerations

This memo raises no security issues.

7. Acknowledgements

Special thanks are due to Vern Paxson of Lawrence Berkeley Labs for his helpful comments on issues of clock uncertainty and statistics. Thanks also to Sean Shapira and to Roland Wittig for several useful suggestions.

8. References

V. Paxson, G. Almes, J. Mahdavi, and M. Mathis, "Framework for IP Performance Metrics", Internet Draft <draft-ietf-ippm-framework-01.txt>, November 1997.

J. Postel, "Internet Protocol", RFC 791, September 1981.

D. Mills, "Network Time Protocol (v3)", RFC 1305, April 1992.

9. Authors' Addresses

Guy Almes <almes@advanced.org>
Advanced Network & Services, Inc.

200 Business Park Drive
Armonk, NY 10504
USA
Phone: +1 914/273-7863

Sunil Kalidindi <kalidindi@advanced.org>
Advanced Network & Services, Inc.
200 Business Park Drive
Armonk, NY 10504
USA
Phone: +1 914/273-1219