Network Working Group
INTERNET-DRAFT
Expiration Date: November 2000

V. Raisanen
Nokia
G. Grotefeld
Motorola
May 2000

Network performance measurement for periodic streams
<draft-ietf-ippm-npmps-01.txt>

1. Status of this Memo

2. Abstract

This document describes some of the issues associated with application-level measurements of network performance for periodic streams. An example application would be the testing of Dst-Src routes for use as bearer for multimedia streams. In this document, the reader is assumed to be familiar with the terminology of the Framework for IP Performance Metrics RFC 2330 [1]. This document is parallel to A One-way Delay Metric for IPPM RFC 2679[2]. A sample metric is described that is suitable for application-level measurement for streaming multimedia over IP. Using such a measurement, transmission service of a network is probed with a traffic stream similar to that of the application of interest, which is likely to be

very dissimilar to the Poisson inter-arrival interval described in [2].

3. Introduction

This document discusses concepts relevant to application-level performance measurements of an IP network. The original driver for this work is Quality of Service of interactive periodic streams such as multimedia conference over IP, but the idea of application-level measurement may have a wider scope. In the following, interactive multimedia traffic is used as an example to illustrate the concept.

A streaming (hereinafter called periodic) multimedia bit stream may be simulated by transmitting uniformly sized packets (or mostly uniformly sized packets) at regular intervals through the network to be evaluated. The "mostly uniformly sized packets" may be found in applications that may use smaller packets during a portion of the stream (e.g. digitally coded voice during silence periods). As noted in the framework document [1], a sample metric using regularly spaced singleton tests has some limitations when considered from a general measurement point of view: only part of the network performance spectrum is sampled. However, from the point of view of application-level performance, this is actually good news as explained below.

IP delivery service measurements have been discussed within the International Telecommunications Union (ITU). A framework for IP service level measurements (with references to the framework for IP performance [1]) that is intended to be suitable for service planning has been approved as I.380 [3]. The emphasis in the ITU recommendation is on passive measurements, though not explicitly forbidding active measurements. The present contribution proposes a method that is usable both for service planning and end-user testing purposes, and is based on active measurements.

3.1 Terminology

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.

3.2 Considerations related to delay

For interactive multimedia sessions, end-to-end delay is an important factor. Too large a delay reduces the quality of the multimedia session as perceived by the participants. End-to-end delay may be best managed in the general case by assigning maximum per-domain quotas (e.g. 50 ms for a particular IP domain). The 50 ms would then be included into a calculation of an end-to-end delay bound.

For example, in estimating the delay bound that can be guaranteed for connections, these measurements can provide a useful tool. This is probably true irrespective of the possible QoS mechanism utilized in the core network. As an example, for a QoS mechanism without hard guarantees, measurements may be used to ascertain that the "best" class gets the service that has been promised for the traffic class in question. Moreover, an operator could study the quality of a cheap, low-guarantee service implemented using possible slack bandwidth in other classes. Such measurements could be made either in studying the feasibility of a new service, or on a regular basis.

3.3 Protocol level issues

In general, the results of measurements may be influenced by individual application requirements/responses related to the following issues:

+ Lost packets: Applications may have varying tolerance to lost packets. Another consideration is the distribution of lost packets (i.e. random or bursty).
+ Long delays: Many applications will consider packets delayed longer than a certain value to be equivalent to lost packets (i.e. real time applications).
+ Duplicate packets: Some applications may be perturbed if duplicate packets are received.
+ Out of sequence: Some applications may be perturbed if packets are received out of sequence. This may be in addition

to the possibility of exceeding the "long" delay threshold as a result of being out of sequence.

+ Corrupt packet header: Most applications will probably treat a packet with a corrupt header as equivalent to a lost packet.
+ Corrupt packet payload: Some applications (e.g. digital voice codecs) may accept corrupt packet payload. In some cases, the packet payload may contain application specific forward error correction (FEC) that can compensate for some level of corruption.
+ Spurious packet: Dst may receive spurious packets (i.e. packets that are not part of the metric). Many applications may be perturbed by spurious packets.

Depending, e.g., on the observed protocol level, some issues listed above may be indistinguishable from others by the application, it may be important to preserve the distinction for the operators of Src, Dst, and/or the intermediate network(s).

Because of the possible errors listed above, in most cases it is recommended to use a packet identifier for each packet generated at Src. Identifiers for the metric sample may be those used by the underlying transport layer (e.g. RTP sequence number) or the same identifiers used by an application if the application to be modeled by the metric uses an identifier. The possibility of identifier roll-over (reuse if intentional) during a metric collected over a "long" (application dependent) time should be observed.

If the application does not use an identifier, it may still be useful to add identifiers to the packets in the metric sample to help identify possible anomalies such as out of sequence packets. This would be most useful in the case where the application expects to receive packets in sequence, but has no capability to identify the sequence of packets received at Dst.

3.4 Application-level measurement

In what follows, a metric is proposed for application-level network performance measurement. In effect, the metric is an emulation of periodic multimedia stream performance. The justification for using realistic application metrics in the measurement:

+ The results of the measurement are automatically relevant to the performance as perceived by the application in question.
+ All the packets in the measurement contribute to accuracy of the estimation of performance variation at timescale that is

important to the multimedia application (packetization interval).

+ Effects of elastic traffic (TCP) on measurement packets are different for a sustained stream than for single packets during overloading situations as discussed in [3].

3.5 Measurement types

Delay measurements can be one-way [2,3], paired one-way, or round-trip [4]. Accordingly, the measurements may be performed either with synchronized or unsynchronized Src/Dst host clocks. Different possibilities are listed below.

The reference measurement setup for all measurement types is shown in Fig. 1.

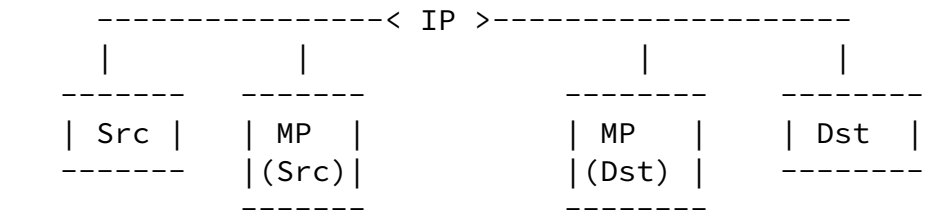

Fig. 1: Example setup for the metric usage.

An example of the use of the metric is a setup with a source host (Src), a destination host (Dst), and corresponding measurement points (MP(Src) and MP(Dst)) as shown in Figure 1. Separate equipment for measurement points may be used if having Src and/or Dst conduct the measurement may significantly affect the delay performance to be measured. MP(Src)should be placed/measured close to the egress point of packets from Src. MP(Dst) should be placed/measure close to the ingress point of packets for Dst. "Close" is defined as a

distance sufficiently small so that application-level performance characteristics measured (such as delay) can be expected to follow the corresponding performance characteristic between Src and Dst to an adequate accuracy.

The test setup just described fulfills two important criteria:
1) Test is made with realistic stream metrics, emulating - for example - a full-duplex VoIP call.
2) Either one-way or round-trip characteristics may be obtained.

It is also possible to have intermediate measurement points between MP(Src) and MP(Dst), but that is beyond the scope of this document.

3.5.1 One way measurement

In the interests of specifying metrics that are as generally usable as possible, application-level measurements based on one-way delays are used in the example metrics. The implication of application-level measurement for bi-directional applications such as interactive multimedia conferencing is discussed below.

Performing a single one-way measurement only yields information on network behavior in one direction. Moreover, the stream at the network transport level does not emulate accurately a full-duplex multimedia connection.

3.5.2 Paired one way measurement

Paired one way delay refers to two multimedia streams: Src to Dst and Dst to Src for the same Src and Dst. By way of example, for some applications, the delay performance of each one way path is more important than the round trip delay. This is the case for delay-limited signals such as voice over IP (VoIP). Possible reasons for the difference between one-way delays is different routing of streams from Src to Dst vs. Dst to Src.

Moreover, paired one way delay measurement emulates a full-duplex VoIP call more accurately than a single one-way measurement only.

3.5.3 Round trip measurement

From the point of view of periodic multimedia streams, round-trip measurements have two advantages: they avoid the need of host clock synchronization and they allow for a simulation of full-duplex connections. The former aspect means that a measurement is easily performed, since no NTP setup is needed. The latter property means that measurement streams are transmitted in both directions. Thus, the measurement provides information on quality of service as experienced by appropriate application.

The downsides of round-trip measurement are the need for more bandwidth than an one-way test and more complex accounting of packet loss. Moreover, the stream that is returning towards the original sender may be more bursty than the one on the first "leg" of the round-trip journey. The last issue, however, means in practice that returning stream experiences worse QoS than the other one, and the performance estimates thus obtained are pessimistic ones. The possibility of asymmetric routing and queuing must be taken into account during analysis of the results.

Please note that with suitable arrangements, round-trip measurements may be performed using paired one way measurements.

4 Sample metric for multimedia stream simulation

The sample metric presented here is similar to the sample metric Type-P-One-way-Delay-Poisson-Stream presented in [2]. "Singletons", as defined in [1] and [2] are not directly used in this document because certain key results (such as duplicate or out of sequence packets) cannot be identified in the context of a singleton, but only as part of a sample.

4.1 Metric name

Type-P-One-way-Delay-Periodic-Stream

4.2 Metric parameters

4.2.1 Global metric parameters

These parameters are applicable to the metrics collected in the following sections (4.2.2, 4.2.3, and 4.2.4).

+ Src, the IP address of a host
+ Dst, the IP address of a host
+ T0, a time, for starting to generate packets and taking

measurements for a sample
+ Tf, a time, greater than T0, for stopping generation of packets for a sample
+ periodic packet interval incT, a time duration
+ packet size p(j), the number of bytes in each packet of Type-P of size j

+ dTloss, a time interval, used for determining if a packet should be considered lost
+ Tcons, a time interval [optional]

While a number of applications will use one packet size (j = 1), other applications may use packets of different sizes (j > 1). Especially in cases of congestion, it may be useful to have packets smaller than the maximum or predominant size of packets in the periodic stream.

4.2.2 Metrics collected at MP(Src)

+ Tstamp(Src)[i], for each packet [i], the time of the packet as measured at MP(Src)
+ PktID [i], for each packet [i], an identification number for the the packet sent from Src to Dst
+ PktSiTy [i], for each packet [i], the packet size and/or type. Some applications may use packets of different size, either because of application requirements or in response to IP performance experienced.

4.2.3 Metrics collected at MP (Dst)

+ dTstop, a time interval, used to add to time Tf to determine when to stop collecting metrics for a sample
+ Tstamp(Dst)[i], for each packet [i], the time of the packet as measured at MP(Dst)
+ PktID [i], for each packet [i], an identification number for the the packet received at Dst from Src. This identification number may be corrupted.
+ PktSiTy [i], for each packet [i], the packet size and/or type. Some applications may use packets of different size, either because of application requirements or in response to IP performance experienced.
+ PktStatus [i], for each packet [i], the status of the packet received. Possible status includes: OK, packet header corrupt, packet payload corrupt, spurious, duplicate

4.2.4 Metrics resulting when metrics collected at MP(Src) and MP(Dst) are merged

These parameters are only available as a complete set when the parameters from the preceding sections (4.2.1, 4.2.2, and 4.2.3 are combined.

+ Tstamp(Src)[i], for each packet [i], the time of the packet as measured at MP(Src). This entry may be blank or noted as N/A for spurious packets received at MP(Dst)
+ Tstamp(Dst)[i], for each packet [i], the time of the packet as measured at MP(Dst). This entry may be blank or noted as N/A for packets not received at MP(Dst), received with corrupt packet headers, or for duplicate packets received at MP(Dst).

+ PktID [i], for each packet [i], an identification number for the the packet received. This identification number may be corrupted for certain packets received at MP (Dst).
+ PktSiTy [i], for each packet [i], the packet size and/or type.
+ PktStatus [i], for each packet [i], the status of the packet received. Possible status includes: OK, packet header corrupt, packet payload corrupt, spurious, duplicate, out of sequence.
+ Delay [i], for each packet [i], the time interval Tstamp(Dst)[i] - Tstamp(Src)[i]. For the following conditions, it will not be possible to be able to compute delay:
  Spurious: There will be no Tstamp(Src)[i] time
  Not received: There will be no Tstamp (Dst) [i]
  Corrupt packet header: There will be no Tstamp (Dst) [i]
  Duplicate: Only the first non-corrupt copy of the packet received at Dst should have Delay [i] computed.
+ SDV[i] [optional] , for each packet [i] except the first one: momentary delay variation between successive packets, i.e., the time interval Delay[i] - Delay [i-1]. SDV[i] may be negative, zero, or positive. Delay for both packets i and i+1 must be calculable according to the definition above or SDV[i] is undefined.

4.3 High level description of the procedure to collect a sample

Beginning on or after time T0, Type-P packets are generated by Src and sent to Dst until time Tf is reached with a nominal interval between the first bit of successive packets of incT as measured at MP(Src). incT may be nominal due to a number of reasons:

variation in packet generation at Src, clock issues (see section 4.7), etc.

MP(Src) records the following information only for packets with timestamps between and including T0 and Tf: timestamp, packet identifier, and packet size/type of each packet sent from Src to Dst that is part of the sample.

MP (Dst) records the following information only for packets with time stamps between T0 and (Tf+ dTstop): timestamp, packet identifier, packet size/type, and received status of each packet received from Src at Dst that is part of the sample. Optionally, at a time Tf + Tcons, the data from MP(Src) and MP(Dst) are consolidated to derive the results of the sample metric.

To prevent stopping data collection too soon, dTcons should be greater than or equal to dTstop. Conversely, to keep data collection reasonably efficient, dTstop should be some reasonable time interval (seconds/minutes/hours), even if dTloss is infinite or extremely long.

4.4 Discussion

The sample metric thus defined is intended to probe the delays and the delay variation as experienced by multimedia streams of an application. Subsequently, the delay is assumed to be measured at transport layer level. Since a range of packet sizes and nominal interval between packets is used, the method probes only a specific time scale of network QoS variations.

There are a number of factors that should be taken into account when collecting a sample metric of Type-P-One-way-Delay-Periodic-Stream.

+ T0 and (Tf + dTloss) should specify a long enough time interval to represent a reasonable use of the application under test (e.g. do not provide only a 100 ms time interval for a phone call)

+ T0 and (Tf + dTloss) should specify a time interval that is not excessively long compared to the usage of the application under test

(e.g. do not provide a one week continuous phone call)

+ The nominal interval between packets (incT) and the packet size(s) (p(j)) should not define an equivalent bit rate that is in excess of the capacity of the egress port of Src, the ingress port of Dst, or the carrying capacity of the intervening network(s). There may be exceptional cases to test the response of the application to overload conditions in the transport networks, but these cases should be strictly controlled.

+ Real delay values will be positive. Therefore, it does not make sense to report a negative value as a real delay. However, an individual zero or negative delay value might be useful as part of a stream when trying to discover a distribution of the delay values of a stream.

+ Depending on measurement topology, delay values may be as low as 100 usec to 10 msec, whereby it may 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). The global metric parameter dTloss defines a time interval such that delays larger than dTloss are interpreted as losses.
{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.}

4.5 Additional Methodology Aspects

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).

4.6 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 [1] provides general guidance on this point, but we note here the following specifics related to delay metrics:

+ Errors or uncertainties due to uncertainties in the clocks of the MP(Src) and MP(Dst) measurement points.

+ Errors or uncertainties due to the difference between 'wire time' and 'host time'.

4.6.1. Errors or uncertainties related to Clocks

The uncertainty in a measurement of one-way delay is related, in part, to uncertainties in the clocks of MP(Src) and MP(Dst). In the following, we refer to the clock used to measure when the packet was measured at MP(Src) as the MP(Src) clock and we refer to the clock used to measure when the packet was received at MP(Dst) as the MP(Dst) clock. Alluding to the notions of synchronization, accuracy, resolution, and skew, we note the following:

+ Any error in the synchronization between the MP(Src) clock and the MP(Dst) clock will contribute to error in the delay measurement. We say that the MP(Src) clock and the MP(Dst) clock have a synchronization error of Tsynch if the MP(Src) clock is Tsynch ahead of the MP(Dst) clock. Thus, if we know the value of Tsynch exactly, we could correct for clock synchronization by adding Tsynch to the uncorrected value of Tstamp(Dst)[i] - Tstamp(Src) [i].

+ 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. When computing delays, we are interested only in the differences between clock values, not the values themselves.

+ The resolution of a clock adds to uncertainty about any time measured with it. Thus, if the MP(Src) clock has a resolution of 10 msec, then this adds 10 msec of uncertainty to any time value measured with it. We will denote the resolution of the source clock and the MP(Dst) clock as ResMP(Src) and ResMP(Dst), 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 Tstamp(Dst)[i] - Tstamp(Src) [i] will be off by Tsynch(t) +/- (ResMP(SRc) + ResMP(Dst)). 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.

4.6.2. Errors or uncertainties related to Wire-time vs Host-time

As we have defined one-way periodic delay, we would like to measure the time between when a packet is measured and time-stamped at MP(Src) and when it arrives and is time-stamped at MP(Dst) and we refer to these as "wire times." 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 generates the packet just prior to sending the test packet and when Dst has started to process the packet after having received the test packet, and we refer to these two points as "host times".

To the extent that the difference between wire time and host time is accurately known, this knowledge can be used to correct for wire time measurements and the corrected value more accurately estimates the desired (host 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 of MP(Src) and host time on the Src host, and similarly define Hdest for the difference between the host time on the Dst host and the wire time of MP(Dst). 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.

4.6.3. Calibration

Generally, the measured values can be decomposed as follows:

measured value = true value + systematic error + random error

If the systematic error (the constant bias in measured values) can be determined, it can be compensated for in the reported results.

reported value = measured value - systematic error

therefore

reported value = true value + random error

The goal of calibration is to determine the systematic and random error generated by the instruments themselves in as much detail as possible. At a minimum, a bound ("e") should be found such that the reported value is in the range (true value - e) to (true value + e) at least 95 percent of the time. We call "e" the calibration error for the measurements. It represents the degree to which the values produced by the measurement instrument are repeatable; that is, how closely an actual delay of 30 ms is reported as 30 ms. {Comment: 95 percent was chosen because (1) some confidence level is desirable to be able to remove outliers, which will be found in measuring any physical property; (2) a particular confidence level should be specified so that the results of independent implementations can be compared; and (3) even with a prototype user-level implementation, 95% was loose enough to exclude outliers.}

From the discussion in the previous two sections, the error in measurements could be bounded by determining all the individual uncertainties, and adding them together to form

Esynch(t) + ResMP(Src) + ResMP(Dst) + Hsource + Hdest.

However, reasonable bounds on both the clock-related uncertainty captured by the first three terms and the host-related uncertainty captured by the last two terms should be possible by careful design techniques and calibrating the instruments using a known, isolated, network in a lab.

For example, the clock-related uncertainties are greatly reduced through the use of a GPS time source. The sum of Esynch(t) +

ResMP(Src) + ResMP(Dst) is small, and is also bounded for the duration of the measurement because of the global time source.

The host-related uncertainties, Hsource + Hdest, could be bounded by connecting two instruments back-to-back with a high-speed serial link or isolated LAN segment. In this case, repeated measurements are measuring the same one-way delay.

If the test packets are small, such a network connection has a minimal delay that may be approximated by zero. The measured delay therefore contains only systematic and random error in the instrumentation. The "average value" of repeated measurements is the systematic error, and the variation is the random error.

One way to compute the systematic error, and the random error to a 95% confidence is to repeat the experiment many times - at least hundreds of tests. The systematic error would then be the median. The random error could then be found by removing the systematic error from the measured values. The 95% confidence interval would be the range from the 2.5th percentile to the 97.5th percentile of these deviations from the true value. The calibration error "e" could then be taken to be the largest absolute value of these two numbers, plus the clock-related uncertainty. {Comment: as described, this bound is relatively loose since the uncertainties are added, and the absolute value of the largest deviation is used. As long as the resulting value is not a significant fraction of the measured values, it is a reasonable bound. If the resulting value is a significant fraction of the measured values, then more exact methods will be needed to compute the calibration error.}

Note that random error is a function of measurement load. For example, if many paths will be measured by one instrument, this might increase interrupts, process scheduling, and disk I/O (for example, recording the measurements), all of which may increase the random error in measured singletons. Therefore, in addition to minimal load measurements to find the systematic error, calibration measurements should be performed with the same measurement load that the instruments will see in the field.

We wish to reiterate that this statistical treatment refers to the calibration of the instrument; it is used to "calibrate the meter stick" and say how well the meter stick reflects reality.

4.7 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 five items to consider: the Type-P of test packets, the threshold of delay equivalent to loss, error calibration, the path traversed by the test packets, and background conditions at Src, Dst, and the intervening networks during a sample. This list is not exhaustive; any additional information that could be useful in interpreting applications of the metrics should also be reported.

4.7.1. 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-Periodic-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.

4.7.2. Threshold for delay equivalent to loss

In addition, the threshold for delay equivalent to loss (or methodology to determine this threshold) MUST be reported.

4.7.3. Calibration results

+ If the systematic error can be determined, it SHOULD be removed from the measured values.

+ You SHOULD also report the calibration error, e, such that the true value is the reported value plus or minus e, with 95% confidence (see the last section.)

+ If possible, the conditions under which a test packet with finite delay is reported as lost due to resource exhaustion on the measurement instrument SHOULD be reported.

4.7.4. Path

Finally, the path traversed by the packets 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 packets 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 may be 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.}

4.7.5 Background conditions

In many cases, the results of a sample may be influenced by conditions at Src, Dst, and/or any intervening networks. Some things that may affect the results of a sample include: traffic levels and/or bursts during the sample, link and/or host failures, etc. Information about the background conditions may only be available by non-Internet means (e.g. phone calls, television) and may only become available days after samples are taken.

4.8 Single sample vs. a "sample of samples"

Because this metric represents a periodic stream as one sample, there may be value in running multiple tests using this metric to collect a "sample of samples". For example, it may be more appropriate to test 1,000 two-minute VoIP calls rather than a single 2,000 minute VoIP call. When considering collection of a sample of samples, issues like the interval between samples (e.g. Poisson vs. periodic, time of

day/day of week), composition of samples (e.g. equal (Tf-T0 duration, different packet sizes), and network considerations (e.g. run different samples over different intervening link-host combinations) should be taken into account. For items like the interval between samples, the pattern of use of the application being measured should be considered.

4.9 Statistics based on Type-P-One-way-Delay-Periodic-Stream

4.9.1 Statistics calculable from one sample

As a metric based on a sample representative of certain applications, some general purpose statistics (e.g. median and percentile) may be less applicable than ways to characterize the range of delay values recorded during the sample metrics.

Example, a sample metric generates 100 packets as measured at MP(Src) with the following measurements at MP(Dst)

+ 80 packets received with delay [i] <= 20 ms
+ 8 packets received with delay [i] > 20 ms
+ 5 packets received with corrupt packet headers
+ 4 packets from MP(Src) with no matching packet recorded at MP(Dst) (effectively lost)
+ 3 packets received with corrupt packet payload and and delay [i] <= 20 ms
+ 2 packets that duplicate one of the 80 packets received correctly in the first line

For this example, packets are considered acceptable if they are received with less than or equal to 20ms delays and without corrupt packet headers or packet payload. In this case, the percentage of acceptable packets is 80/100 = 80%.

For a different application which will accept packets with corrupt packet payload and no delay bound (so long as the packet is received), the percentage of acceptable packets is (80+8+3)/100 = 91%.

4.9.2 Statistics calculable from multiple samples

For computing statistics, a "sample of samples" series of measurements may be performed. As discussed in section 4.8, under these conditions, general purpose statistics (e.g. median, percentile, etc.) may be more relevant as a more statistically significant number of packets are used.

5. Security Considerations

5.1 Denial of Service Attacks

This metric generates a periodic stream of packets from one host (Src) to another host (Dst) through intervening networks. This metric could be abused for denial of service attacks directed at Dst and/or the intervening network(s).

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

5.2 User data confidentiality

This metric generates packets for a sample metric, rather than taking samples based on user data. Thus, this metric does not threaten user data confidentiality.

5.3 Interference with the metric

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

6. Acknowledgements

The authors wish to thank the chairs of the IPPM WG, Howard Stanislevic and Al Morton for comments that have made the present draft clearer and more focused. The authors have also built on the substantial foundations laid by the authors of the framework for IP performance [1].

7. References

[1] V.Paxson, G.Almes, J.Mahdavi, and M.Mathis: Framework for IP Performance Metrics, IETF RFC 2330, May 1998.
[2] G.Almes, S.Kalidindi, and M.Zekauskas: A one-way delay metric for IPPM, IETF RFC 2679, September 1999.
[3] International Telecommunications Union recommendation I.380, February 1999.
[4] G.Almes, S.Kalidindi, and M.Zekauskas: A round-trip delay metric for IPPM, IETF RFC 2681.
[5] S. Bradner: Key words for use in RFCs to Indicate Requirement Levels, RFC 2119, March 1997.

8. Authors' contact information

Vilho Raisanen <Vilho.Raisanen@nokia.com>
P.O. Box 407
Communication Systems Laboratory
Nokia Research Center
FIN-00045 Nokia Group
Finland
Phone +358 9 4376 1
Fax. +358 9 4376 6852

Glenn Grotefeld <g.grotefeld@motorola.com>
Motorola, Inc.
1303 E. Algonquin Road
4th Floor
Schaumburg, IL 60196
USA
Phone +1 847 576-5992
Fax +1 847 538-7455

EXPIRES NOVEMBER 2000