               Network performance measurement with periodic streams

Status of this Memo

Copyright Notice

Abstract

   This memo describes a periodic sampling method and relevant metrics
   for assessing the performance of IP networks. First, the memo
   motivates periodic sampling and addresses the question of its value
   as an alternative to the Poisson sampling described in RFC 2330. The
   benefits include applicability to active and passive measurements,
   simulation of constant bit rate (CBR) traffic (typical of multimedia
   communication, or nearly CBR, as found with voice activity
   detection), and several instances in which analysis can be
   simplified. The sampling method avoids predictability by mandating
   random start times and finite length tests. Following descriptions
   of the sampling method and sample metric parameters, measurement
   methods and errors are discussed. Finally, we give additional
   information on periodic measurements, including security
   considerations.

Table of Contents

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 BCP 14, RFC 2119 [2].
   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 that the results of measurements from two different
   implementations are comparable, and to note instances in which an
   implementation could perturb the network.

2. Introduction

   This memo describes a sampling method and performance metrics
   relevant to certain applications of IP networks. The original driver
   for this work was Quality of Service of interactive periodic streams,
   such as multimedia conferencing over IP, but the idea of periodic
   sampling and measurement has wider applicability. Interactive
   multimedia traffic is used as an example below to illustrate the
   concept.

   Transmitting equally sized packets (or mostly same-size packets)
   through a network at regular intervals simulates a constant bit-rate
   (CBR), or a nearly CBR multimedia bit stream. Hereafter, these
   packets are called periodic streams. Cases of "mostly same-size
   packets" may be found in applications that have multiple coding
   methods (e.g. digitally coded comfort noise during silence gaps in
   speech).

   In the following sections, a sampling methodology and metrics are
   presented for periodic streams. The measurement results may be used
   in derivative metrics such as average and maximum delays. The memo
   seeks to formalize periodic stream measurements to achieve comparable
   results between independent implementations.

2.1 Motivation

   As noted in the IPPM framework RFC 2330 [3], 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, some applications also
   sample this limited performance spectrum and their performance may be
   of critical interest.

   Periodic sampling is useful for the following reasons:

   * It is applicable to passive measurement, as well as active

measurement.

   * An active measurement can be configured to match the
     characteristics of media flows, and simplifies the estimation of
     application performance.

   * Measurements of many network impairments (e.g., delay variation,
     consecutive loss, reordering) are sensitive to the sampling
     frequency. When the impairments themselves are time-varying (and
     the variations are somewhat rare, yet important), a constant
     sampling frequency simplifies analysis.

   * Frequency Domain analysis is simplified when the samples are
     equally spaced.

   Simulation of CBR flows with periodic streams encourages dense
   sampling of network performance, since typical multimedia flows have
   10 to 100 packets in each second. Dense sampling permits the
   characterization of network phenomena with short duration.

3. Periodic Sampling Methodology

   The Framework RFC [3] points out the following potential problems
   with Periodic Sampling:

   1. The performance sampled may be synchronized with some other
      periodic behavior, or the samples may be anticipated and the
      results manipulated. Unpredictable sampling is preferred.

   2. Active measurements can cause congestion, and periodic sampling
      might drive congestion-aware senders into a synchronized state,
      producing atypical results.

   Poisson sampling produces an unbiased sample for the various IP
   performance metrics, yet there are situations where alternative
   sampling methods are advantageous (as discussed under Motivation).

   We can prescribe periodic sampling methods that address the problems
   listed above. Predictability and some forms of synchronization can
   be mitigated through the use of random start times and limited stream
   duration over a test interval. The periodic sampling parameters

produce bias, and judicious selection can produce a known bias of
interest. The total traffic generated by this or any sampling method
should be limited to avoid adverse affects on non-test traffic
(packet size, packet rate, and sample duration and frequency should
all be considered).

The configuration parameters of periodic sampling are:
+ T, the beginning of a time interval where a periodic sample is
   desired.
+ dT, the duration of the interval for allowed sample start times.
+ T0, a time that MUST be selected at random from the interval
   [T, T+dT] to start generating packets and taking measurements.
+ Tf, a time, greater than T0, for stopping generation of packets
   for a sample (Tf may be relative to T0 if desired).
+ incT, the nominal duration of inter-packet interval, first bit to
   first bit.

T0 may be drawn from a uniform distribution, or T0 = T + Unif(0,dT).
Other distributions may also be appropriate. Start times in
successive time intervals MUST use an independent value drawn from
the distribution. In passive measurement, the arrival of user media
flows may have sufficient randomness, or a randomized start time of
the measurement during a flow may be needed to meet this requirement.

When a mix of packet sizes is desired, passive measurements usually
possess the sequence and statistics of sizes in actual use, while
active measurements would need to reproduce the intended distribution
of sizes.

4. Sample metrics for periodic streams

The sample metric presented here is similar to the sample metric
Type-P-One-way-Delay-Poisson-Stream presented in RFC 2679[4].
Singletons defined in [3] and [4] are applicable here.

4.1 Metric name

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

[4.2](#) Metric parameters

[4.2.1](#) Global metric parameters

   These parameters apply in the following sub-sections (4.2.2, 4.2.3,
   and 4.2.4).

   Parameters that each Singleton usually includes:
     + Src, the IP address of a host
     + Dst, the IP address of a host
     + IPV, the IP version (IPv4/IPv6) used in the measurement
     + dTloss, a time interval, the maximum waiting time for a packet
        before declaring it lost.
     + packet size p(j), the desired number of bytes in the Type-P
        packet, where j is the size index.

   Optional parameters:
     + PktType, any additional qualifiers (transport address)
     + Tcons, a time interval for consolidating parameters collected at
        the measurement points.

   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 use packets
   smaller than the maximum or predominant size of packets in the
   periodic stream.

Raisanen, et al.   Standards Track   [Page 5]

   A topology where Src and Dst are separate from the measurement points
   is assumed.

[4.2.2](#) Parameters collected at the measurement point MP(Src)

   Parameters that each Singleton usually includes:
   + Tstamp(Src)[i], for each packet [i], the time of the packet as
      measured at MP(Src)

   Additional parameters:
   + PktID(Src) [i], for each packet [i], a unique identification or
      sequence number.
   + PktSi(Src) [i], for each packet [i], the actual packet size.

   Some applications may use packets of different sizes, either because

of application requirements or in response to IP performance
experienced.

4.2.3 Parameters collected at the measurement point MP(Dst)

   + Tstamp(Dst)[i], for each packet [i], the time of the packet as
      measured at MP(Dst)
   + PktID(Dst) [i], for each packet [i], a unique identification or
      sequence number.
   + PktSi(Dst) [i], for each packet [i], the actual packet size.

   Optional parameters:
   + dTstop, a time interval, used to add to time Tf to determine when
      to stop collecting metrics for a sample
   + PktStatus [i], for each packet [i], the status of the packet
      received. Possible status includes OK, packet header corrupt,
      packet payload corrupt, duplicate, fragment. The criteria to
      determine the status MUST be specified, if used.

4.2.4 Sample Metrics resulting from combining parameters at MP(Src)
      and MP(Dst)

   Using the parameters above, a delay singleton would be calculated as
   follows:

   + Delay [i], for each packet [i], the time interval
                   Delay[i] = Tstamp(Dst)[i] - Tstamp(Src)[i]

Raisanen, et al.   Standards Track   [Page 6]

   For the following conditions, it will not be possible to compute
   delay singletons:

   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.

A sample metric for average delay is as follows

$$AveDelay = (1/N)Sum(from\ i=1\ to\ N,\ Delay[i])$$

assuming all packets i= 1 through N have valid singletons.

A delay variation [5] singleton can also be computed:

+ IPDV[i], for each packet [i] except the first one, delay variation
  between successive packets would be calculated as

$$IPDV[i] = Delay[i] - Delay\ [i-1]$$

IPDV[i] may be negative, zero, or positive. Delay singletons for
packets i and i-1 must be calculable or IPDV[i] is undefined.

An example metric for the IPDV sample is the range:

$$RangeIPDV = max(IPDV[]) - min(IPDV[])$$

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 4.6), etc.
MP(Src) records the parameters above only for packets with timestamps
between and including T0 and Tf having the required Src, Dst, and any
other qualifiers. MP (Dst) also records for packets with time stamps
between T0 and (Tf + dTstop).

Optionally at a time Tf + Tcons (but eventually in all cases), the
data from MP(Src) and MP(Dst) are consolidated to derive the sample
metric results. 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

This sampling methodology is intended to quantify the delays and the
delay variation as experienced by multimedia streams of an
application. Due to the definitions of these metrics, packet loss
status is also recorded. The nominal interval between packets
assesses network performance variations on a specific time scale.

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.

+   The interval T0 to Tf should be specified to cover a long enough
    time interval to represent a reasonable use of the application
    under test, yet not excessively long in the same context (e.g.
    phone calls last longer than 100ms, but less than one week).

+   The nominal interval between packets (incT) and the packet size(s)
    (p(j)) should not define an equivalent bit rate that exceeds the
    capacity of the egress port of Src, the ingress port of Dst, or
    the capacity of the intervening network(s), if known. 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
    errors.

+   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 the NTP agents used, and this delay is what we
    are trying to measure.

+   A given methodology will have to include a way to determine
    whether a packet was lost or whether delay 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: For many
    applications, the treatment of a large delay as infinite/loss will
    be inconsequential. A TCP data packet, for example, that arrives
    only after several multiples of the usual 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 RFC [3] provides general guidance on this point, but we
   note here the following specifics related to periodic streams and
   delay metrics:

   +   Error due to variation of incT. The reasons for this can be
       uneven process scheduling, possibly due to CPU load.

   +   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 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
      measurement or calculation must be repeated 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

   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); we refer to these as "wire times." However, if
   timestamps are applied by software on Src and Dst, 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; 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. The corrected value more accurately estimates the
   desired (host time) metric, and visa-versa.

   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 due to reasons discussed in [4], briefly
   summarized as (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.}

   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.

4.6.4 Errors in incT

The nominal interval between packets, incT, can vary during either
active or passive measurements. In passive measurement, packet
headers may include a timestamp applied prior to most of the protocol
stack, and the actual sending time may vary due to processor
scheduling. For example, H.323 systems are required to have packets
ready for the network stack within 5 ms of their ideal time. There
may be additional variation from the network between the Src and the

MP(Src). Active measurement systems may encounter similar errors,
but to a lesser extent. These errors must be accounted for in some
types of analysis.

4.7 Reporting

The calibration and context in which the method is used MUST be
carefully considered, and SHOULD always be reported along with metric
results. We next 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 [3], the value of a 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 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

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

Raisanen, et al.   Standards Track   [Page 13]

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

5. Additional discussion on periodic sampling

   Fig.1 illustrates measurements on multiple protocol levels that are
   relevant to this memo. The user's focus is on transport quality
   evaluation from the application point of view. However, to properly
   separate the quality contribution of the operating system and codec
   on packet voice, for example, it is beneficial to be able to measure
   quality at the IP level [6]. Link layer monitoring provides a way of
   accounting for link layer characteristics such as bit error rates.

```
        --------------
       | application |
        --------------
       |  transport  | <--
        --------------
       |   network   | <--
        --------------
       |    link     | <--
        --------------
       |   physical  |
        --------------
```
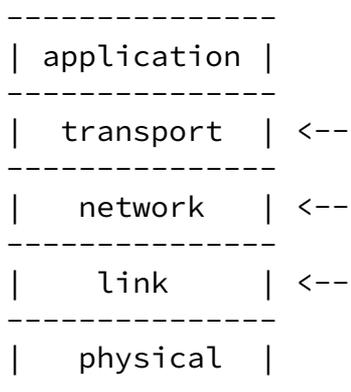
   Fig. 1: Different possibilities for performing measurements: a
   protocol view. Above, "application" refers to all layers above L4
   and is not used in the OSI sense.

   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.
+ Reordering: Some applications may be perturbed if packets arrive
  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 sent by the Src as 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).

5.1 Measurement applications

   This sampling method provides a way to perform measurements
   irrespective of the possible QoS mechanisms utilized in the IP
   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.

   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 [3]) that is intended to be suitable for service planning
   has been approved as I.380 [7]. ITU-T Recommendation I.380 covers
   abstract definitions of performance metrics. This memo describes a
   method that is useful, both for service planning and end-user testing
   purposes, in both active and passive measurements.

Delay measurements can be one-way [3,4], paired one-way, or round-
trip [8]. Accordingly, the measurements may be performed either with
synchronized or unsynchronized Src/Dst host clocks. Different
possibilities are listed below.

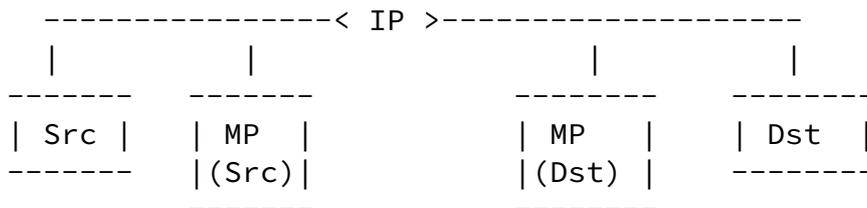The reference measurement setup for all measurement types is shown in
Fig. 2.

```
        ----------------< IP >--------------------
         |          |                 |           |
      -------    -------           --------    --------
     | Src |    | MP  |           | MP   |    | Dst  |
      -------    |(Src)|           |(Dst) |     --------
                -------           --------
```

                Fig. 2: Example measurement setup.

An example of the use of the method 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 2. 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 basic principle here is that measurement
results between MP(Src) and MP(Dst) should be the same as for a
measurement between Src and Dst, within the general error margin
target of the measurement (e.g., < 1 ms; number of lost packets is
the same). If this is not possible, the difference between MP-MP
measurement and Src-Dst measurement should preferably be systematic.

The test setup just described fulfills two important criteria:

1) The test is made with realistic stream metrics, emulating - for
   example - a full-duplex Voice over IP (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.

5.1.1 One way measurement

   In the interests of specifying metrics that are as generally
   applicable 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.

5.1.2 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 VoIP. Possible reasons for the difference
   between one-way delays is different routing of streams from Src to
   Dst vs. Dst to Src.

   For example, a paired one way measurement may show that Src to Dst
   has an average delay of 30ms, while Dst to Src has an average delay
   of 120ms. To a round trip delay measurement, this example would look
   like an average of 150ms delay. Without the knowledge of the
   asymmetry, we might miss a problem that the application at either end
   may have with delays averaging more than 100ms.

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

5.1.3 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
   communication. The former aspect means that a measurement is easily
   performed, since no special equipment or 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 two-way applications.

The downsides of round-trip measurement are the need for more
bandwidth than a 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
the returning stream may experience worse QoS than the out-going one,
and the performance estimates thus obtained are pessimistic ones.
The possibility of asymmetric routing and queuing must be taken into
account during an analysis of the results.

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

5.2 Statistics calculable from one sample

Some statistics may be particularly relevant to applications
simulated by periodic streams, such as the range of delay values
recorded during the sample.

For example, a sample metric generates 100 packets 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 delay
   [i] <= 20 ms
+   2 packets that duplicate one of the 80 packets received correctly
   as indicated in the first item

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

5.3 Statistics calculable from multiple samples

There may be value in running multiple tests using this method to
collect a "sample of samples". For example, it may be more
appropriate to simulate 1,000 two-minute VoIP calls rather than a
single 2,000 minute call. When considering a collection of multiple
samples, issues like the interval between samples (e.g. minutes,
hours), 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 usage
pattern for the application of interest should be considered.

When computing statistics for multiple samples, more general
statistics (e.g. median, percentile, etc.) may have relevance with a
larger number of packets.

5.4 Background conditions

In many cases, the results may be influenced by conditions at Src,
Dst, and/or any intervening networks. Factors that may affect the
results 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 external means (e.g. phone calls,
television) and may only become available days after samples are
taken.

5.5 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. One approach for managing
end-to-end delays on an Internet path involving heterogeneous link
layer technologies is to use per-domain delay quotas (e.g. 50 ms for

a particular IP domain). However, this scheme has clear
inefficiencies, and can over-constrain the problem of achieving some
end-to-end delay objective. A more flexible implementation ought to
address issues like the possibility of asymmetric delays on paths,
and sensitivity of an application to delay variations in a given
domain. There are several alternatives as to the delay statistic one
ought to use in managing end-to-end QoS. This question, although
very interesting, is not within the scope of this memo and is not
discussed further here.

6. Security Considerations

6.1 Denial of Service Attacks

This method generates a periodic stream of packets from one host
(Src) to another host (Dst) through intervening networks. This
method 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

Raisanen, et al.   Standards Track   [Page 19]

method in excess of the terms agreed between the participants may be
cause for immediate rejection, discard of packets, or other
escalation procedures defined between the affected parties.

6.2 User data confidentiality

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

6.3 Interference with the metric

It may be possible to identify that a certain packet or stream of
packets is part of a sample. 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.

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

7. IANA Considerations

Since this method and metric do not define a protocol or well-known
values, there are no IANA considerations in this memo.

8. Normative References

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

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

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

   [4]   Almes, G., Kalidindi, S. and M. Zekauskas, "A one-way delay
         metric for IPPM", RFC 2679, September 1999.

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

9. Informative References

   [6]   "End-to-end Quality of Service in TIPHON systems; Part 5: Quality
         of Service (QoS) measurement methodologies", ETSI TS 101 329-5
         V1.1.2, January 2002.

   [7]   International Telecommunications Union, "Internet protocol data
         communication service _ IP packet transfer and availability

performance parameters", Telecommunications Sector
Recommendation I.380 (re-numbered Y.1540), February 1999.

[8] Almes, G., Kalidindi, S. and M. Zekauskas, "A round-trip delay
metric for IPPM", RFC 2681, September 1999.

10. Acknowledgments

The authors wish to thank the chairs of the IPPM WG (Matt Zekauskas
and Merike Kaeo) for comments that have made the present document
more clear and focused. Howard Stanislevic and Will Leland have also
presented useful comments and questions. We also gratefully
acknowledge Henk Uijterwaal's continued challenge to develop the
motivation for this method. The authors have built on the
substantial foundation laid by the authors of the framework for IP
performance [3].

11. Author's Addresses

Vilho Raisanen
Nokia Networks
P.O. Box 300

   FIN-00045 Nokia Group
   Finland

   Phone: +358 7180 8000
   Fax: +358 9 4376 6852
   EMail: Vilho.Raisanen@nokia.com

   Glenn Grotefeld
   Motorola, Inc.
   1501 W. Shure Drive, MS 2F1
   Arlington Heights, IL 60004 USA

   Phone: +1 847 435-0730
   Fax: +1 847 632-6800
   EMail: g.grotefeld@motorola.com

   Al Morton
   AT&T Labs
   Room D3 - 3C06
   200 Laurel Ave. South
   Middletown, NJ 07748 USA

   Phone: +1 732 420 1571
   Fax: +1 732 368 1192
   EMail: acmorton@att.com

12. Full Copyright Statement

Acknowledgement